

SPATIALANALYZER USER'S MANUAL



New River Kinematics, Inc.

1.21.2008

United States copyright law protects this manual, associated software disks, algorithms, and visual content. All Colors are and shall remain the sole property of New River Kinematics, Inc.

Copyright 1996-2008 New River Kinematics

SOFTWARE LICENSE AGREEMENT OF NEW RIVER KINEMATICS, INC.

("MANUFACTURER")

NOTICE -- READ BEFORE INSTALLING THIS PACKAGE

CAREFULLY READ THE TERMS AND CONDITIONS OF THIS AGREEMENT BEFORE INSTALLING THIS PACKAGE. INSTALLING THIS PACKAGE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS AGREEMENT, PROMPTLY RETURN THIS PACKAGE TO THE PLACE OF PURCHASE FOR REFUND IN THE AMOUNT YOU PAID.

1. Definitions

The Software Package is licensed (not sold) to you, and Manufacturer owns all copyright, trade secret, patent and other proprietary rights in the Software Package. The term "Software Package" includes all copies of the Spatial Analyzer computer program and its documentations.

2. License

a. Authorized Use. Manufacturer grants you a nonexclusive license to use the Software Package on a single computer. You may make one copy of the Software Package's computer program for archive and back-up purposes only.

b. Restrictions: You may not: (1) copy (other than once for archive and back-up purposes), distribute, rent, lease or sublicense all or any portion of the Software Package; (2) modify or prepare derivative works of the Software Package; (3) remove or destroy and proprietary marking of Manufacturer; or (4) reverse engineer, decompile or disassemble the Software Package. You agree to keep confidential and use your best efforts to prevent and protect the contents of the Software Package from unauthorized disclosure or use.

c. Transfer: You may transfer the Software Package, but only if the recipient agrees to accept the terms and conditions of this Agreement. If you transfer the Software Package, you must transfer all computer programs and documentation and erase any copies residing on computer equipment. Your license is automatically terminated if you transfer the Software Package.

3. Disclaimer of Warranty and Limitation of Remedies

YOU UNDERSTAND AND AGREE AS FOLLOWS:

a. MANUFACTURER DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL OUR LIABILITY OF ANY KIND INCLUDE ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES,

INCLUDING LOST PROFITS, EVEN IF WE HAVE KNOWLEDGE OF THE POTENTIAL LOSS OR DAMAGE.

b. We will not be liable for any loss or damage caused by delay in furnishing a Software Package or any other performance under this Agreement.

c. Our entire liability and your exclusive remedies for our liability of any kind (including liability for negligence except liability for personal injury caused solely by our negligence) for the Software Package covered by this Agreement and all other performance or nonperformance by us under or related to this Agreement are limited to the remedies specified by this Agreement.

d. Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights, which vary from state to state.

4. Termination

This Agreement is effective until terminated. You may terminate it at any time by destroying the Software Package, including all computer programs and documentation, and erasing any copies residing on computer equipment. This Agreement also will terminate if you do not comply with any terms or conditions of this Agreement. Upon such termination you agree to destroy the Software Package and erase all copies residing on computer equipment.

SPATIALANALYZER USER'S MANUAL	1
SOFTWARE LICENSE AGREEMENT OF NEW RIVER KINEMATICS, INC.	2
INTRODUCTION	2
SA Features	2
Modern 32-Bit Windows Application	2
User Friendly Graphical Environment	3
Customizable Reporting.....	3
Multiple Simultaneous Active Instruments	4
Import and Export of Popular File Formats	5
Direct CATIA, UG, and ProE	5
Real-time Build-Measure with CAD Geometry	6
Efficient, Tightly Integrated System Architecture	6
Custom Scripting and Macros	6
Advanced Analysis Routines	6
Versatile Coordinate System Management	7
About this Manual.....	7
Before Proceeding.....	7
System Requirements.....	7
GETTING STARTED	8
Installation	8
Required Access Privileges	8
General Installation	8
Readme.txt File	13
Hardware Lock Installation.....	14
Software License Control and Distribution	18
Installation of the Dongle (Hardware Lock)	19
Installation of SA Meta-Key file.....	19
Installation of a Software Lock	19
TCP/IP Protocol and Network Configuration	22
Windows Network Configuration	23
Windows2000® and XP® Network Configuration	28
Checking the Final Configuration	29
System Configuration.....	30
Virtual Memory.....	30
SYSTEM ARCHITECTURE	33

SpatialAnalyzer System Overview	33
SpatialAnalyzer Central Application.....	34
User Interface Features	34
Special User Interface Features.....	39
Cursor Prompting	39
Mouse Functions	40
Hover Point view	40
Repeat Command(s).....	41
Toolbar	42
Geometric Objects within SpatialAnalyzer	47
The Special Case of Groups, Points, and Targets	47
Instrument Interfaces.....	48
Starting an Instrument Interface.....	48
Gathering Data from multiple instruments.....	51
Multiple instruments working independently.....	51
Cooperative instrumentation example (locating one instrument with another).....	52
Cooperative instrumentation example (real-time monitoring position and orientation of a part)	53
Cooperative instrumentation example (multi-lateration for extreme precision)	54
Remotely Monitoring a Measurement Session with another copy of SpatialAnalyzer	54
Survey templates.....	56
Import/Export Modules	57
Importing Data	57
Exporting Data	63
USER INTERFACE CUSTOMIZATION AND PROFILES	71
Customizing the Interface.....	71
Recalling a Saved Profile	73
Removing Profiles.....	73
GROUPS, POINTS AND TARGETS	74
Notational Conventions.....	74
Graphical Distinction between Points and Targets	75
Creating a New Group	76
Inspecting the Properties of a Group.....	76
Moving, Copying, and Deleting Groups of Data.....	77
View Point List.....	77
SpatialAnalyzer User Manual	5

Editing Point List.....	78
Using the Group Manger	79
Renaming Points and Targets.....	80
Deleting Groups or Individual Points.....	82
Moving Whole Groups of Data.....	83
Copying Whole Groups of Data.....	84
Moving Individual Points and Targets	85
Copying Individual Points and Targets	86
Surface Normals (Direction).....	87
Reversing Surface Normals.....	88
Show Object Lables.....	89
Querying Point/Target Data	90
Querying Target Information	90
Querying a Single Target	90
Querying Relationships between Targets.....	93
COLLECTIONS	95
Adding Collections	95
Activating a Collection	96
Deleting Collections	97
Renaming Collections.....	97
Importing Jobs into Collections	98
Copy and Move data between collections.....	98
CONSTRUCT	100
Properties Common to all Geometric Entities	100
Color Property.....	100
Facet Count	100
Construct Geometries	101
Points and Targets.....	101
Constructing Points and Targets	103
Entering Coordinates.....	104
Center Point Methods.....	105
Point Projections Methods	105
Projection of Points onto Objects.....	110
Intersection Point Construction Methods	110
End of a Line.....	112
Shift Point in a Direction.....	112
Fit to Points	113

Hidden Point Bar.....	113
Patch Normal Shift Construction Methods	113
Plane Section Crossing.....	115
From a Surface Mesh	115
From Cloud Points	116
From Existing Points.....	118
Filter Clouds to Points.....	119
Filter Clouds to Plane.....	121
Auto-Correspond Two Groups.....	122
Scale Points	125
Layout	128
Lines.....	138
Fit to Points	139
Enter.....	140
2 Points Ctrl+L.....	141
Mutual Perpendicular.....	141
Projection >> Line to >> Plane // Circle	141
Between a Point and a Line.....	141
2 Plane Intersection.....	142
Center of Cylinder.....	142
From Instrument Shot	142
Normal to Object(s)	143
Normal to Objects, through Point	143
Parallel to Line, Through Point	144
Circles.....	144
Fit to points	145
Enter.....	146
Maximum Material Condition (from scan patch).....	147
Planes.....	147
Fit to Points	149
Enter.....	150
Normal to Object, through Point.....	150
Normal to Line, through end.....	150
Mirror from 2 measured points	150
Cylinders	152
Enter.....	152
Fit to Points	153
Sphere.....	154
Enter.....	154
Fit to Points	155
Paraboloid	156
Enter.....	156
Fit to Points	157
Cone.....	157
Enter.....	157
Fit to Points	158

Ellipse	158
Enter.....	158
Re-Fit Geometry, Use Existing Geometry as Starting Point.....	159
Curves.....	160
From Surfaces	161
Surfaces	161
From a Collection of Surfaces.....	161
By Dissecting Surfaces	162
From Plane	162
From Point Groups and From B-Splines.....	162
From B-Splines	162
GD&T MODULE	164
Overview	164
Getting Started	164
Creating GD&T Datum Annotations	164
Creating a GD&T Feature Check Annotation.....	171
Using GD&T Annotations	175
GD&T Module.....	175
GD&T Toolbar	185
Overview	185
Getting Started	185
Creating Datum Annotations.....	186
Creating Feature Check Annotations	187
Editing/Deleting Annotations.....	188
GD&T Manual Creation.....	189
Overview	189
Getting Started	189
Creating GD&T Datum Annotations	190
Creating a GD&T Feature Check Annotation.....	197
Using the GD&T Annotations	201
COORDINATE FRAMES AND SPATIAL TRANSFORMATIONS	202
Creating New Coordinate Frames	202
Manual Entry of Coordinate Frame Transformation (Enter).....	202
Plane, Line, Point (3-2-1) Method	204
Using Points and Vectors Method.....	207
Three Point Methods	208
Z Axis Normal to Plane, Normal to Circle, or Along a Line	211
On Instrument	211
Pick Origin Point.....	212
3-Plane Frame Method.....	213
Mirror Cube Frame Assignment	214
On an Object Method.....	216
Fit to Coordinate Components	216
Toggle Left-Handed Frame Construction Method.....	217

Changing the Working Coordinate Frame	218
Basics of Spatial Transformations	218
The old viewpoint debate: Are we describing or transforming?	219
Unit Rotation Matrices	219
Compound Rotations.....	219
Fixed Angle XYZ.....	220
Fixed Angle ZYX.....	220
Euler Angle XYZ.....	220
Euler Angle ZYX.....	220
Euler Angle ZYZ	220
Equivalent Angle Axis	221
Quaternion (also know as Euler parameters)	221
MEASUREMENT ERROR AND UNCERTAINTY.....	222
Measurement Error.....	222
What is Systematic Error?.....	222
What is Random Error?.....	222
Reducing Measurement Error	223
Sources of Measurement Error	223
Operator Error	224
Instrument Error.....	225
Instrument Resolution	225
Instrument Calibration	225
Uncertainty contribution from the Survey or Job Setup.....	226
Coordinate Uncertainty	226
Measurement Perturbations.....	229
Uncertainty Analysis and Visualization.....	229
BEST-FIT COORDINATE TRANSFORMATIONS (OR 7-PARAMETER TRANSFORMATIONS)	233
Least Squares Adjustment.....	233
Best Fit Spatial Transformations	235
User Interface	237
Controlling Best-Fit Transformations	239
Best-Fit Scale	242
Best-Fit Set Scale	244
Best-Fit Results Reporting	245
Applying Best-Fit Transformations	246
Best-Fit Dialog Functionality.....	246
Applying the Transformation.....	249
Locating an Instrument using a Best-Fit Transformation.....	249

Advanced Spatial Transformations	252
Surface transformations	252
 BUNDLE ADJUSTMENT	 259
Initial Guess	259
Scale-Bar Database.....	259
Running the Bundle Algorithm	259
Selecting Targets to Bundle	261
Scale-Bar Weight Factors	261
Viewing System Errors	261
Beginning Computation	262
 Local Minima	 262
Weighting	262
 UNIFIED SPATIAL METROLOGY NETWORK.....	 263
Basic USMN Process	264
Observations and Point Group Requirements for USMN	264
Typical Process Flow with USMN.....	267
USMN with Point Groups	268
 USMN User Interface.....	 269
Initiating the USMN.....	269
Overview USMN User Interface.....	269
The Uncertainty Field Analysis controls are used to compute point uncertainty estimates. The estimates are based on individual measurement component uncertainties for each station and the geometric configuration of the network	270
Solve	270
Best-Fit and Solve	271
Auto Solve	271
Solution Quality Indicators	271
Ranking and Max Error.....	272
USMN Composite Point Group	273
Exclude Measurements	273
Trimming Outliers.....	273
Instrument List	274
Fixed vs. Un-Fixed Instruments	274
Instrument Properties	274
Scaling Instruments in USMN Solution.....	276
Point List.....	277
Point Properties	278
Uncertainty Field Analysis.....	279
Reporting.....	280
Instrument Uncertainty Analysis.....	280
Changing Instrument Uncertainty Estimates.....	281
Summary Statistics.....	281
Apply Results.....	281
Measurement configuration differences between Bundle and USMN	282

Why USMN?	283
Why is it Necessary to Combine Coordinate Acquisition Systems?	283
Uncertainty Issues when Combining Measurement Systems	285
Representing Coordinate Uncertainty Fields	286
Requirements for Uncertainty Field Representation	286
Adequate Representation of the Spatial Uncertainty	286
Transformable Between Coordinate Systems	287
Combinable to Produce Composite Uncertainties	288
Intuitive and Easily Applied to Measurement Process Design	289
Computationally Reasonable	289
Applicable to Measurement Analysis Operations	289
Discrete Point Cloud Uncertainty Field Representation	290
Advantages of the Discrete Point Cloud Uncertainty Field Representation	294
Quantifying Coordinate Acquisition System Uncertainty Fields	295
Instrument Uncertainty Characterization	295
Measurement Process	296
Data Analysis and Optimization	298
Spatial Analyzer User Options	298
Reporting for USMN	300
Callout View Reporting of Uncertainty	300
Quick Report with Uncertainties	302
Fit and Query Reports with Uncertainty Analysis Results	304
Uncertainties of Individual Network Components	304
Design Model Uncertainty	305
Solving the Unified Spatial Metrology Network	305
Example: Weighting by Relative Uncertainty	307
Forming Combined Coordinate Acquisition System Uncertainty Fields	310
Unified Spatial Metrology Network Results	317
Uncertainty Field Density	318
Measurement Analysis within the Unified Spatial Metrology Network	321
Distance Uncertainty between Points	321
Geometric Object Fitting	322
Sphere Fit Uncertainty Example	323
Cylinder Fit Uncertainty	324
Case Study: Laser Tracker Performance Survey	325
Application of Uncertainty Characterization to Instrument Performance Evaluation	330
Instrument Performance Evaluation Example: Before and After Instrument Compensation	331
Aircraft Carrier Catapult Alignment	335
Disney Concert Hall	341
Submarine Section Assembly	344
Nuclear Power Plant Steam Generator Replacement	355
USMN Review	360
Coordinate Uncertainty Field Representation	360
Instrument Uncertainty Characterization	360
Measurement System Combination Methods	360
Task Specific Measurement Uncertainty	361
Summary	361

RELATIONSHIPS.....	362
Introduction to Relationships	362
Relationship Optimization Example.....	363
Add Simple Design Geometry	363
Add an Instrument + Fabricate Measurements.....	368
Making Relationships.....	371
Move the Instrument and the measured points follow	374
Optimize with Relationships	376
Weighted Relationship Fitting	379
Add Additional Constraints with Relationships	381
Minimize all the Relationships Simultaneously	384
 WATCH WINDOWS	 389
Complete Coordinate Instruments.....	389
Add Point to	389
Add Instrument's Point	390
Add Closest Point.....	390
Theodolite Data Observer (TDO)	391
 SIMULATION AND SURFACE ANALYSIS	 392
Measurement Survey Simulation	392
Surface Analysis	399
 REPORTS	 405
Configure Default SA Report Options.....	405
Report Templates	406
Report Templates Walkthrough	407
Quick Reports	412
Composite Quick Reports	413
Extending Quick Reporting.....	414
Template Reporting Example	415
Opening a Report View.....	417
HTML Web Reporting.....	417
Office XP Reporting	419
 CHARTING, TREND ANALYSIS AND SPC	 425
Chart Types.....	425
Individual X – Moving Range SPC Chart.....	426
Bullseye Chart.....	426
SpatialAnalyzer User Manual	12

Run Chart	427
Making a Chart from a Vector Group	427
Chart Options	429
Outputting Charts	430
SA DOCUMENTS.....	431
Inserting Objects into SADocs.....	431
Saving SADocs	433
USER OPTIONS.....	435
Display	435
Graphics Mode	436
Autoscale Settings.....	436
Big Window Cache Size	437
View Animation Steps	437
Line Arrow Size Factor.....	437
Shots Settings.....	437
Coordinate Frame Settings.....	437
Target (or Point) Settings	437
Analysis	438
Uncertainty Options	438
Target Computation	439
Reporting Tolerances	439
Automatic File Backup	439
Scale-Bar Database.....	440
How Scale-Bars are Used.....	440
Adding a Scale-Bar	441
Modifying a Scale-Bar	441
Hidden-Point Bar Database.....	442
Adding a Hidden-Point Bar.....	442
Modifying a Hidden-Point Bar.....	443
Automatic Hidden-Point Computation.....	443
Security.....	444
Login Settings	444
Configuring Security	445
Machine Configuration.....	445
Units.....	446
Length Area	446
Increments.....	447
Idle Counts Before Update.....	447
Edit Angle-Distance Weights.....	447
SpatialAnalyzer User Manual	13

Convergence Tolerance.....	447
Length and Angle Perturbations.....	448
Damping.....	448
Light Conditions	449
Ambient, Diffuse, and Specular Light Settings.....	449
Automation.....	451
MEASUREMENT PLAN	452
Measurement Plan.....	452
MP Benefits	452
Overview of MP Architecture	453
Measurement Plan Components: Template, Script and Interface	453
MP Editor.....	454
Detailed View	456
An MP Step.....	457
Data Types	459
Selection Methods.....	463
Result Arguments.....	465
Running a MP	466
Applications	466
Viewing and Running a Sample MP.....	467
Constructing a basic MP.....	467
Executing the basic MP.....	485
Advanced MP concepts	487
Executing from a command Line	487
MP Process Flow Checks.....	488
References to other arguments	488
Sharing data between plans and with other applications.....	489
Database (ODBC) Functions with MP.....	492
MP Creation.....	493
Inserting New Records	498
Verify Record Insertion.....	501
Updating Existing Records	501
Verify Record Update	503
Retrieving Data from the Database	503
Verify Record Retrieval	505
Database Install and Setup	506
Download and Installation	506
Test the Connection and Create the Sample DB	506
Quick Measurement Plans (QuickMPs)	507
Accessing Quick MPs	507
Configuring Quick MP Assignments	508
Utilizing User Interface Customization.....	509

MP Function Listing.....	510
File Operations.....	510
MP Subroutines.....	529
View Control.....	531
Construction Operations.....	535
Analysis Operations.....	570
Reporting Operations.....	599
MS Office Reporting Operations.....	604
Process Flow Operations.....	609
Scalar Math Operations.....	615
Vector Math Operations.....	618
Instrument Operations.....	620
Utility Operations.....	648

REVERSE DESIGN	653
-----------------------------	------------

MENU REFERENCE	655
-----------------------------	------------

File	655
New Open... Ctrl+O Save Ctrl+S Save As.....	656
Backup Now Alt+SPACE.....	656
Open Template File.....	656
Print Setup.....	656
Print Screen	656
Import.....	657
Export.....	663
Remote Monitor another SpatialAnalyzer.....	670
Remote Window Viewer.....	671
Recent Files.....	673
Exit.....	673

Edit.....	673
Copy graphics to clipboard	674
Pick Working Frame Alt+W	675
User Options... Ctrl+U	676
Set Object's Color to Working Color.....	676
Object Properties Alt+Q.....	677
Move Objects	677
Copy Objects Ctrl+C.....	686
Mirror Objects.....	686
Delete Objects Del	686
Delete CAD Objects by Source.....	686
Group Manager...	687
Delete Points	688
Delete Cloud Points	688
Rename	688
Surface Normal Conditioning	688
Calculator.....	689

View	690
Show Labels.....	690
Show Instrument Shots	691
Big Data Window.....	691

Watch Windows	692
Point Selection Tree-View F2	692
Point Lists for Groups	692
Object Visibility	693
Set View Point	695
Zoom	696
Pan	697
Viewport Control	697
Refresh	697
Create AVI Movie.....	697
Construct.....	700
Point(s)	700
Entering Coordinates.....	701
Center Point Methods.....	702
Point Projections Methods	702
Projection of Points onto Objects.....	707
Intersection Point Construction Methods	707
End of a Line.....	709
Shift Point in a Direction.....	709
Fit to Points	710
Hidden Point Bar.....	710
Patch Normal Shift Construction Methods	710
Plane Section Crossing.....	712
From a Surface Mesh	712
From Cloud Points	713
From Existing Points.....	715
Filter Clouds to Points.....	716
Filter Clouds to Plane.....	718
Auto-Correspond Two Groups.....	719
Scale Points	722
Layout	725
Frames.....	736
Line	736
Fit to Points	737
Enter.....	737
2 Points Ctrl+L.....	738
Mutual Perpendicular	738
Projection >> Line to >> Plane // Circle	738
Between a Point and a Line.....	738
2 Plane Intersection	739
Center of Cylinder.....	739
From Instrument Shot	739
Normal to Object(s)	740
Normal to Objects, through Point	740
Parallel to Line, Through Point	741
Circles.....	741
Fit to points	742
Enter.....	743
Maximum Material Condition (from scan patch).....	744

Planes.....	744
Fit to Points	745
Enter.....	747
Normal to Object, through Point.....	747
Normal to Line, through end	747
Mirror from 2 measured points	747
Cylinders	748
Enter.....	748
Fit to Points	749
Sphere.....	750
Enter.....	750
Fit to Points	751
Paraboloid	752
Enter.....	752
Fit to Points	753
Cone	753
Enter.....	753
Fit to Points	754
Ellipse	754
Enter.....	754
Re-Fit Geometry, Use Existing Geometry as Starting Point.....	755
Curves.....	755
From Surfaces	755
Surfaces	756
From a Collection of Surfaces.....	756
By Dissecting Surfaces	756
From Plane.....	757
From Point Groups and From B-Splines.....	757
From B-Splines	757
Boxes.....	758
Vector Group.....	758
User Defined	758
Pyramid... ..	759
Scale-Bar.....	759
Query.....	760
GD&T	760
Point to	760
Points to Object.....	762
Line	765
Plane to Plane Angle	766
Frame to Frame (Relationship)	766
Group to Point.....	767
Group to Group	767
Make Vector Group (whiskers).....	768
Add a Group to Group Relationship	768

Object to Object Direction (Relationship).....	768
Sensitivity Cloud to Sensitivity Cloud	769
Relationships.....	769
Instrument.....	771
Add... Alt+I	771
Run Interface Module	772
Run Interface Module and Connect	773
Connection Status	773
Properties	774
Drag Position.....	775
Remove	776
Locate (Transform to Part).....	776
Bundle Adjust...	776
Single Theodolite Setup	776
Synchronized Measurement	777
Theodolite Data Observer	777
Auto Measure Mode.....	778
Auto Measure Mode (streamlined)	778
Auto Measure a Vector Group (multi-measure).....	778
Measure Batch of Points	778
AutoCorrespond with Proximity Trigger >> Points ... Vectors.....	778
Watch Windows with Auto-Coorespond with Proximity Trigeers	779
Create Vectors while Measuring with Auto-Coorespond with Proximity Trigger modes	780
Point Instrument at a Point	781
Analysis	781
Re-Compute Targets From Shots	782
Re-Compute Hidden Points.....	782
Activate All Measurements	782
Best-Fit Transformation	782
Undo Last Best-Fit Transform	782
Temperature Compensation	783
Measurement Simulation	784
Coordinate Uncertainty	784
Pipe Relationships Module	785
Benefits of Pipe Fitting Relationships.....	786
SA Robot Calibration Module	786
Survey Data Collection	787
Optimization.....	788
Robot Calibration (background).....	Error! Bookmark not defined.
Scripts.....	792
Run Measurement Plan	793
Quick MP...	793
Accessing Quick MPs	793
Configuring Quick MP Assignments	794
Create/Edit Measurement Plan...	795
Embedded Measurement Plan	795
Repeat Command ... Ctrl+Tab	796
Command History ... Ctrl+Shift+Tab.....	796
Reports	796

Configure Default SA Report Options	797
Report Templates	798
Report Templates Walkthrough	799
Quick Reports	800
Composite Quick Reports	802
Extending Quick Reporting.....	803
MS Office Reporting.....	803
INSTRUMENT INTERFACES.....	805
Adding an Instrument to an SA Job	806
Starting an Instrument Interface	807
LASER TRACKER INTERFACES: LEICA, FARO, SMX, API.....	812
Laser Tracker Overview.....	812
IFM and ADM Measurement.....	813
System Architecture.....	813
Retro-Reflector Targets.....	813
Common Tracker Interface	814
Starting the Interface	814
Initialize API and Simulation Trackers	815
Start Leica Trackers	816
Start Faro or SMX Trackers.....	816
Tracker Simulator	817
Close Instrument Interface	818
Taking Measurement Data	818
Auto-Measure commanded from SA	819
Overview of the Tracker Interface.....	820
Instrument Index and Target Naming	820
Measurement, Home, and ADM Controls.....	822
Measurement Profiles and Target/Retro Interface	826
Statistical Filtering for Discrete Point Measurements.....	826
Modify Sphere Profile to configure Tooling Ball Profile.....	839
Setting a Measurement Delay	845
Managing Measurement Profiles	846
Tracker Interface Configuration Menu.....	847
Adding Reflectors and Targets to the Interface.....	847
Ops Checks / Field Checks	852
Setting and Monitoring the IFM Beam	855
Adding Reflectors and Targets to the Interface.....	856
Advanced Features	861
General Tracker Settings.....	861
SA Point Request	861
Environmental Monitoring	862
Gravity Measurement.....	863
SpatialAnalyzer User Manual	19

Tracker Interface Units	867
Importing Tracker Data.....	867
Viewing Current Position.....	868
Tracker Status	869
Importing Polar Data into SA	870
MP OpCheck string commands for trackers... ..	876
Leica NETBEUI (TP-Link) Setup Process	876
TP-Link → Setting-Up the PC	877
Preparing an XP PC to run Leica TP-Link Tracker	877
Connecting to the Tracker.....	877
Initializing the tracker from SA	878
Initializing in Axyz before starting SA	880
Connection of The Weather Station.....	882
Tracker Calibration	883
 PORTABLE CMM INTERFACE: FARO, ROMER, SANDIA NL	 885
Starting the Interface.....	885
Customizing the Interface.....	889
Arm Calibration	894
Measure Modes.....	895
Single Points	897
Stream Points:	899
Patch Measurement:.....	901
Pin Measurement:	904
Hole Measurement:	906
Slot Measurement:	907
Line Measurement:	909
Circle Measurement:	911
Plane Measurement:.....	912
Sphere Measurement:.....	913
Cross Section Measurement:.....	914
Frame Measurement:.....	916
Guided Measurement:	917
Inspections:	919
Running Perceptron V5 Scanner with Cimcore Arm in Spatial Analyzer	920
Scanner Settings:.....	924
Scanner Calibration:.....	927
 LASER SCANNER: METRIS CLR	 928
Getting Started	928
Taking Your First Measurement	931
Interface Description.....	933
Scanning	944

Special Scan modes.....	947
SpatialAnalyzer Measurement Plan Functionality.....	947
Instrument Operations.....	948
Scan Within Perimeter	949
Auto-Measure Vectors (multi-measure).....	950
Cloud Viewer Operations	950
Utility Operations	951
Composite Core Measurement	951
Auto-Measure Composite Core Mode in SA	953
 LASER SCANNER: MINOLTA VIVID	 956
Taking Your First Scan.....	958
Interface Features.....	960
Measurement Plan Functions	974
 THEODOLITE & TOTAL STATION INTERFACES.....	 977
Cabling and Power Options.....	977
Direct to the Instrument	977
Using a Serial Distribution Interface (SDI).....	978
Adding a Theodolite to an SA Job	978
Measurement Modes	982
Freeform Mode	983
Single Target Mode.....	985
Prompted List Mode.....	986
Theodolite Data Observer.....	990
 ARCSECOND NETWORK AND TRANSMITTER SIMULATOR INTERFACES	 991
Adding a Arcsecond Network.....	992
Connecting to 3DiWorkbench	993
Sending Updates to SA	995
Watch Stream	996
Sending Measurements to SA	996
 SpatialAnalyzer User Manual	 21

Measuring Discrete Points from the Interface	997
Interface Settings	997
Language Settings	998
KRYPTON – SPACE PROBE INTERFACE	1000
Metris Contact Info:	1000
KCMM Software Script	1000
KCMM Settings Configuration	1001
LASER PROJECTOR INTERFACE	1004
SCALERT TEMPERATURE MONITORING INTERFACE	1012
Basic Setup and Monitoring	1012
Basic Configuration	1015
Sending Data to SA	1019
Viewing Data in SA	1022
Watch List	1026
API LASER RAIL INTERFACE	1032
OPEN INSTRUMENT INTERFACE	1038
Creating the Interface (Visual C++)	1038
List of Open Instrument Interface Control Methods	1044
Order of Operation	1046
Example Code	1047
SABUTTON CONTROL INTERFACE	1050

INTRODUCTION

SpatialAnalyzer is a powerful, general-purpose metrology package designed to allow users to quickly acquire measurement data, check its validity, and perform complex geometric analysis. The system design is based on a central graphical environment, which provides, among other features, the computational power necessary to orient instrument networks, merge measurement systems together based on any number of common points or common geometrical features, and compute the error bounds for each measured target.

SA Features

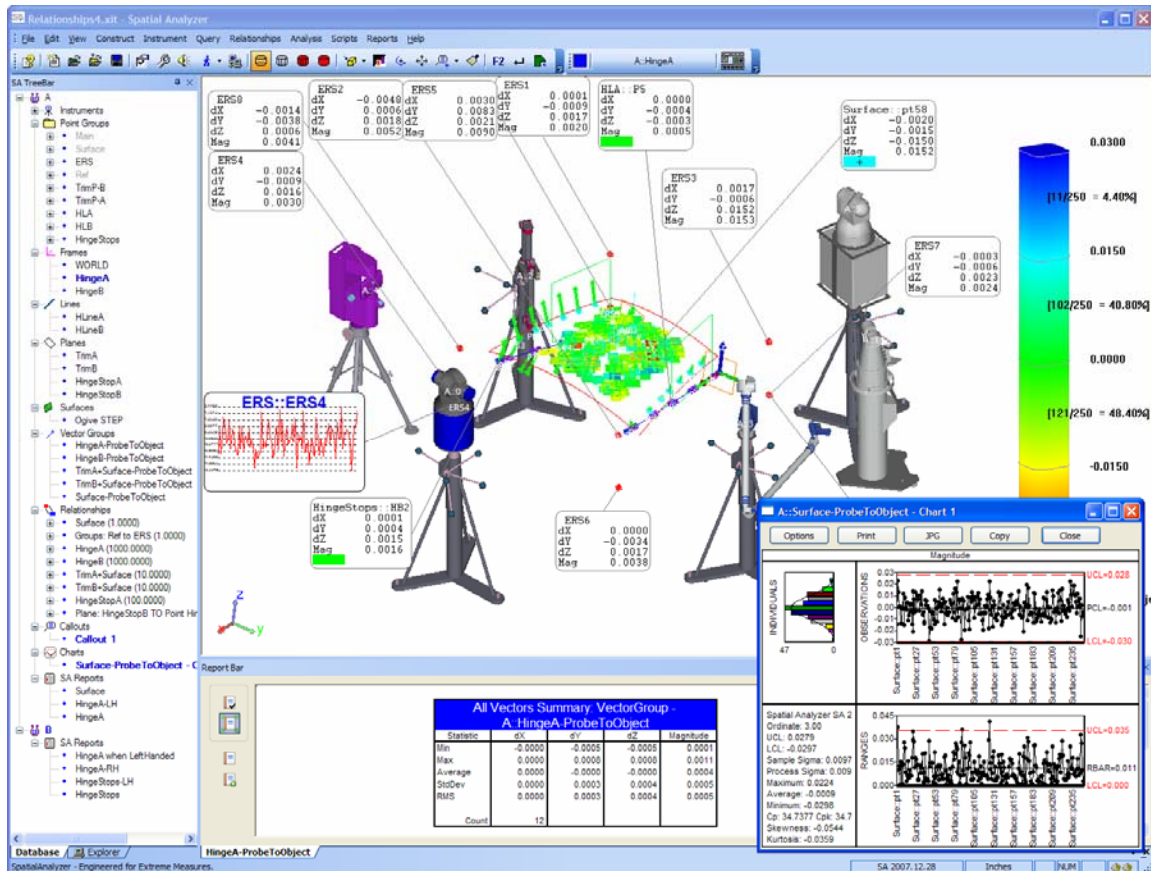
This section presents some of the basic features present in **SpatialAnalyzer**.

Modern 32-Bit Windows Application

SpatialAnalyzer was designed from the ground up with the intent of being always compatible with the Microsoft 32-Bit Windows environments (Windows NT 4.0, Windows 2000, WindowsXP, etc). These are the most widely used operating systems in the world, and serve as a standard platform for many other metrology, scientific, and engineering applications.

By using the Windows environment **SpatialAnalyzer** benefits from the acceptance and familiarity already achieved by the operating system. A large reason for the wide spread acceptance Windows is that Graphical User Interface (GUI) standards were adopted and enforced early on. Many of the functions that an operator has learned for word processing, or Web browsing, can be directly applied to **SpatialAnalyzer**. In short, an operator doesn't need to be trained to open a new file, save a job, or print a report. He or she simply uses the same file open, file save, print, or print-preview sequences one does in Microsoft Word. Adhering to the interface standards established by Microsoft also allows seamless integration of images and reports into other applications via the Windows Clipboard interface.

User Friendly Graphical Environment



Graphical User Interface of SpatialAnalyzer

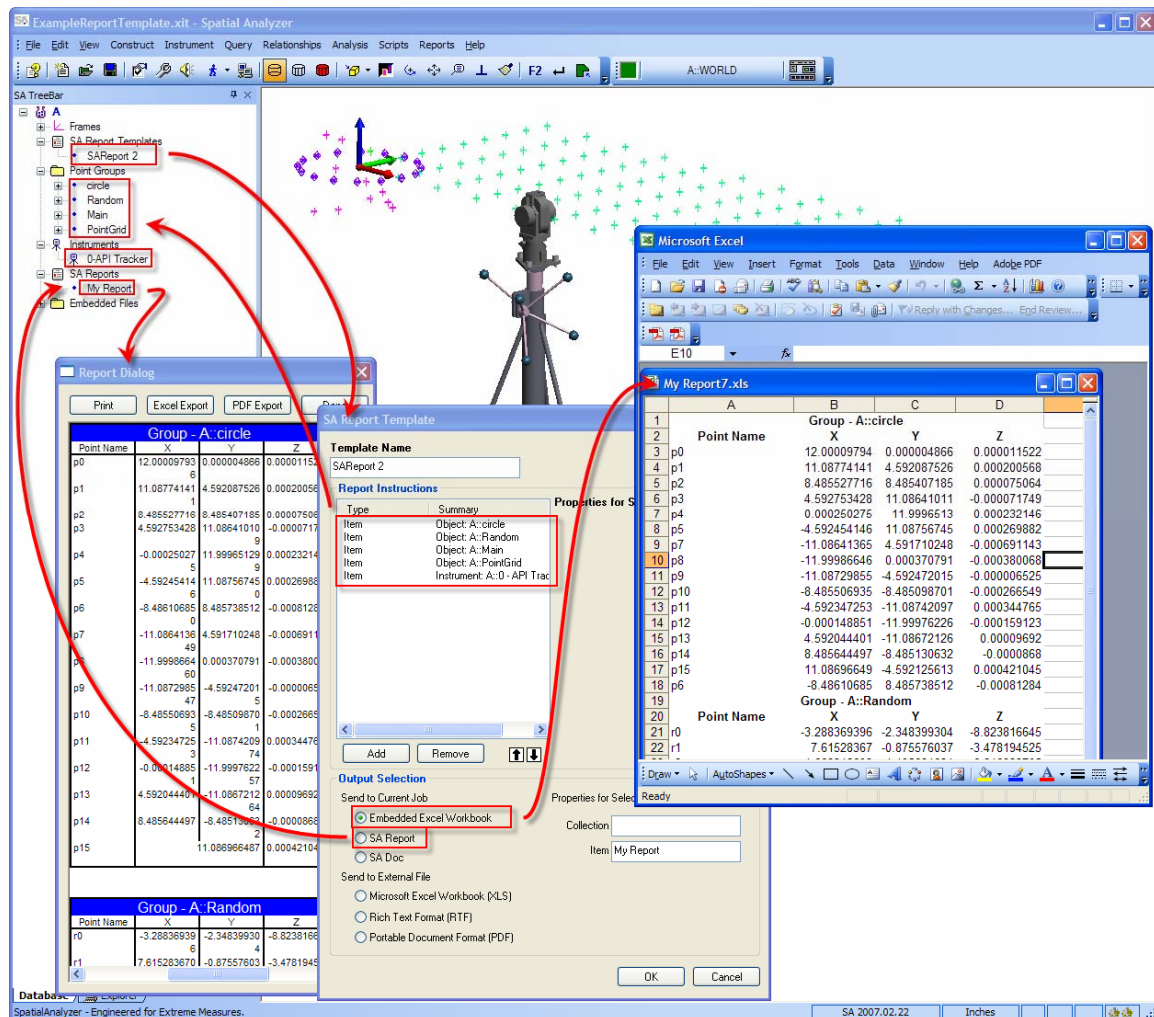
Graphics are a natural way to present 3-D information. In fact, when confronted with a list of three 3-D coordinates to verify, most experienced users simply form a 3-D picture in their head, mentally adding one point at a time to see if it relates to the coordinates of other points in the job as expected. Of course, this approach has its limitations and relies on an operator that has vast experience and good 3-D analytical abilities. The default method of viewing data in **SpatialAnalyzer** is the graphical view. Using the graphics window a user may simultaneously view all the objects in a job from a perspective that they control. The graphical view provides an easy intuitive method for orienting the user with respect to the data being presented. This should be the primary goal of all metrology software.

Customizable Reporting

An important feature of **SpatialAnalyzer** is its ability to generate customizable reports. They can include graphics, data, instruments, measurements, and charts. The report templating mechanism supports outputting a variety of different file formats. Output formats include SA Docs, Excel, pdf, RTF, and SA Reports. The reporting framework in **SpatialAnalyzer** is designed to provide flexibility to satisfy users, engineers, and managers. Various levels of technical detail are supported with choices on graphical

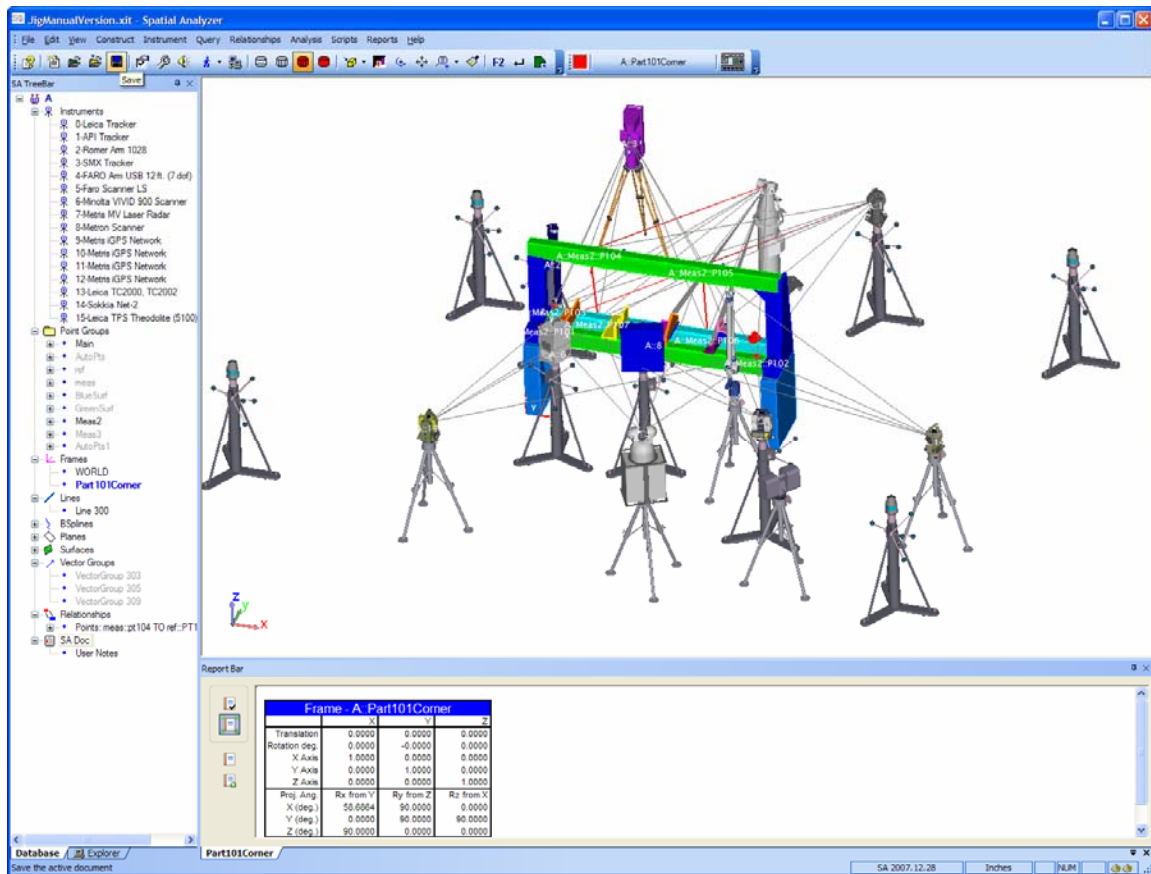
views and measurement traceability. It allows reports to be tailored for specific audiences.

Configuring a custom report is straight-forward with the SA Report Templates. With the current release you can create a Report Template and include each point group. Use the Reports >> Add new SA Report Template. The figure below shows the Report template properties dialog. Add items for each point group and instrument in your job. Select the output mode. Select the Report Template on the treeview and then right-click it. Select the Generate option to create the report.



Multiple Simultaneous Active Instruments

Specific instrument interface modules provide communication from **SpatialAnalyzer** to the measurement instruments. These modules contain the specific code necessary to communicate, monitor, and configure individual instruments. In addition, these modules contain a general communications protocol designed to communicate the measurement data directly to **SpatialAnalyzer**.



Example of Job with Multiple Instrument Types

The communication between the instrument interface and **SpatialAnalyzer** is accomplished using standard TCP/IP Internet Sockets¹. TCP/IP provides the flexibility to distribute the instrument interface modules and the main **SpatialAnalyzer** module across multiple computers. These computers may be distributed in the same area using Ethernet or across the country using the Internet. In single computer configurations, the instrument interface modules are run on the same platform as **SpatialAnalyzer** using the same general socket protocol.

Import and Export of Popular File Formats

SpatialAnalyzer supports many popular file formats including STEP, IGES, ASCII, VDA, DMIS, Geodetic Services Inc. VSTARS, and DXF files.

Direct CATIA, UG, and ProE

SA uses an interface to the Adobe 3D translation provider for Direct CATIA, UG, and ProE data exchange. The interface is able to import product manufacturing information

¹ TCP/IP is a set of protocols developed to allow cooperating computers to share resources across a network.

(PMI) such as dimensions, tolerances, and annotations directly from CATIA V5 files. The annotations are shown on the geometry and in the treeview. The metrology related annotations are easily turned into Datums and Feature Checks.

The native CAD file conversion is a feature based translation. Geometric entities (e.g., circle, lines, edges, curves, surfaces, solids etc.) along with their parameters (e.g., names, colors, and levels) from the originating CAD system are imported into the active SA collection. SA uses the translation function to read the inbound model's geometry, features, parametric properties. Save entities in an SA xit file. Use it as a template for measuring multiple parts. Measure each unit in a new collection comparing each back to the original CAD.

Real-time Build-Measure with CAD Geometry

SpatialAnalyzer shows users on the floor the actual location of the real components and sub-assemblies against the nominal CAD model of the assembly and each of the components and sub-assemblies. The real-time visualization of the part against the nominal location allows operators to easily and competently make minor adjustments to component positions and therefore produce assemblies that consistently meet precise functional specifications. At the user's request, **SpatialAnalyzer** continuously collects measurements from the connected measurement systems and uses the data to show the graphical representation of the component in moving position and orientation. A ghosted representation of the component's nominal location and orientation is always shown to provide an easy graphical image of the nominal and actual locations. As the user moves the real component to its nominal position and orientation **SpatialAnalyzer** will show the component moving against the backdrop of the larger structure of the total assembly. When the component is physically within the tolerances defined by either the CAD model or user; the system will change the components color in a distinctive pattern (i.e., blinking or changing colors faster and faster) as it approaches the nominal location. The user knows the real component is located within tolerance when its graphical representation is not blinking or changing colors.

Efficient, Tightly Integrated System Architecture

SpatialAnalyzer manages its own internal database, renders all graphics itself and it performs fits and optimizations. In order to produce a package that is specifically tailored toward real-time spatial metrology, these operations had to be designed specifically for the task. This provides performance that is not possible with code dependencies on other general packages.

Custom Scripting and Macros

Measurement Planning (MP) is a powerful process control and programming capability for users **SpatialAnalyzer**. By using **SpatialAnalyzer's** Measurement Plans, operators can script repetitive or automated metrology tasks. These plans can vary from simple utilities constructed in just a few minutes to comprehensive inspection plans that provide go/nogo type automation and drive other machinery such as robots or CNC machines.

Advanced Analysis Routines

At the core of **SpatialAnalyzer** is a robust analysis engine. This engine along with the efficient database and data storage methods allows **SpatialAnalyzer** to handle large

quantities of data in real-time. Geometric fits containing thousands of data points are achieved readily and accurately.

Versatile Coordinate System Management

SpatialAnalyzer allows an operator to effortlessly manage an unlimited number of coordinate systems. Coordinates may be presented and spatial relationships queried with respect to any of the established coordinate frames.

About this Manual

You are about to embark on learning how to adeptly use the most sophisticated metrology software package to date. We think this learning period will prove to be remarkably short. Most users are able to confidently acquire and analyze data after only one day. In fact, it's quite possible that you can start using **SpatialAnalyzer** simply by performing a few trial and error test drives. The software is robust, and in general, it will politely inform you if you try something impossible.

We suggest that you may ultimately save time by taking a few moments to read through the manual. Don't be too particular about memorizing anything. After all, the complete manual is available from within **SpatialAnalyzer**. We hope you will obtain an understanding of the philosophy behind **SpatialAnalyzer** by reading this manual.

Above all else, the goal of our work has been to develop a software package where users of all levels of expertise may actually enjoy taking and analyzing measurements and may have confidence in the numbers they report.

Before Proceeding

Warning!

It is the responsibility of the user to provide appropriate training for operators of this software and associated equipment and to design fail-safe protection for all individuals and equipment near any machinery. New River Kinematics shall not be liable or responsible for any incidental or consequential damages resulting from the use or misuse of this software.

System Requirements

SpatialAnalyzer is designed to run on relatively modest computing platforms. Most currently produced 32-Bit personal computers are capable of running this software.

Recommendations for best performance

The system outlined below will provide a good base platform for using the most advanced features of **SpatialAnalyzer**. With this equipment measurement data may be gathered with high data rates (1 kHz sampling rates) into complex models containing numerous CAD surfaces.

- Greater than 1GHz Pentium IV PC

- Windows 2000® or WindowsXP® operating system
- +512 Megabytes of system RAM
- 5 Gigabyte Free Hard Drive Storage Space (Both for models and Program)
- 1024 x 768 Video Driver with 64K Color Depth
- 64 Meg VRAM (or better) preferred.

GETTING STARTED

In order to familiarize you with **SpatialAnalyzer** we will step you through configuring a simple measurement system and taking measurements. This “tutorial” will provide an overview of the powerful functionality that will be presented in greater detail in subsequent chapters of this manual.

Installation

The installation of **SpatialAnalyzer** is similar to most Windows packages. **SpatialAnalyzer** is supplied on either a CD-ROM or available on the Internet at the New River Kinematics website <http://www.kinemantics.com/>. In all cases, the goal of the installation procedure outlined below is to run the **SpatialAnalyzer 200X.XX.XX-Installer.exe** program contained on the installation CD or on your hard drive after downloading it. The date is used in the install application name to identify the specific version of Spatial Analyzer. Users familiar with how to accomplish this task in Windows® may proceed with any suitable method. For those unfamiliar with these operations, we have outlined below the specific steps for installing on Windows Me®, Windows NT®, Windows 2000®, or WindowsXP® operating systems.

Required Access Privileges

Administration privileges are typically required to install the SA software and hardware lock driver (from Rainbow). Once setup the typical user profile is sufficient for running the software.

General Installation

Insert the first installation disk in a CD-ROM drive.

Hit the button on the taskbar.

Select the “...” option.

In the space provided type “<Drive Letter>: \ **SpatialAnalyzer 200X.XX.XX-Installer.exe**”, (e.g. d:\sainstall.exe) and hit OK.

You will be prompted for necessary information during the installation. The following is a summary of a typical installation procedure.

The following procedure is typical of many program installations. If you are comfortable with standard installations, just proceed as you normally would. If you would like to, you can also follow the descriptions presented below to walk you through the installation process.

Software License Agreement Page

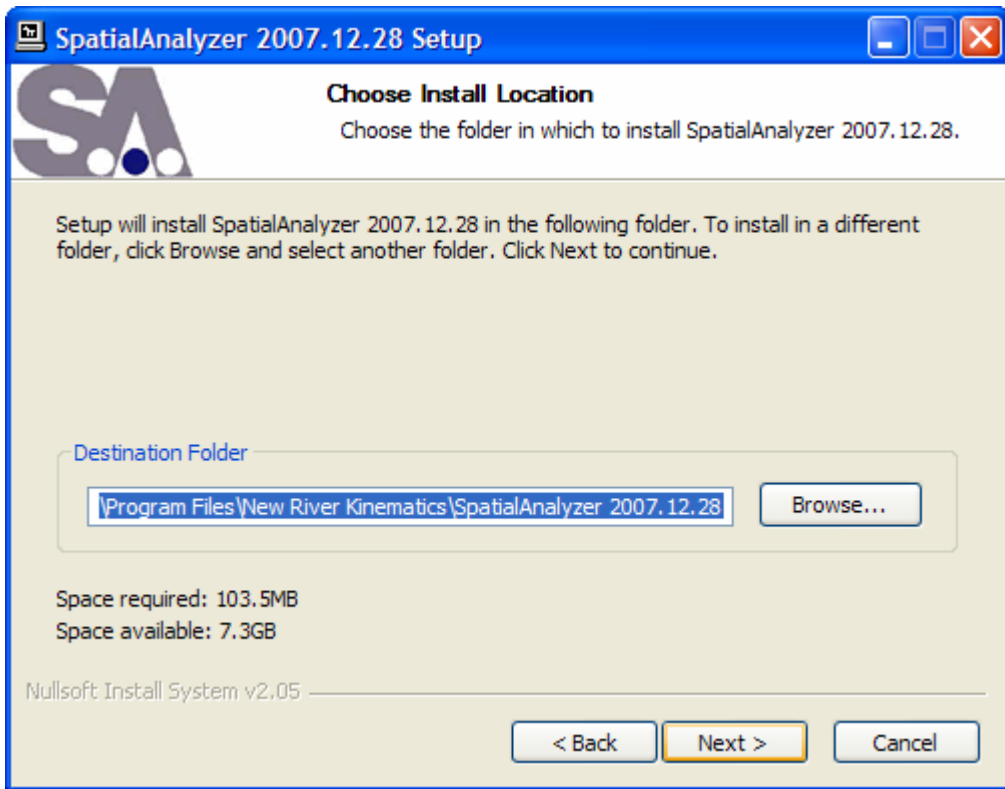
Read this section carefully. You must agree to the terms and conditions of the License Agreement before proceeding with the installation.



New River Kinematics Software License Agreement

Select Destination Page

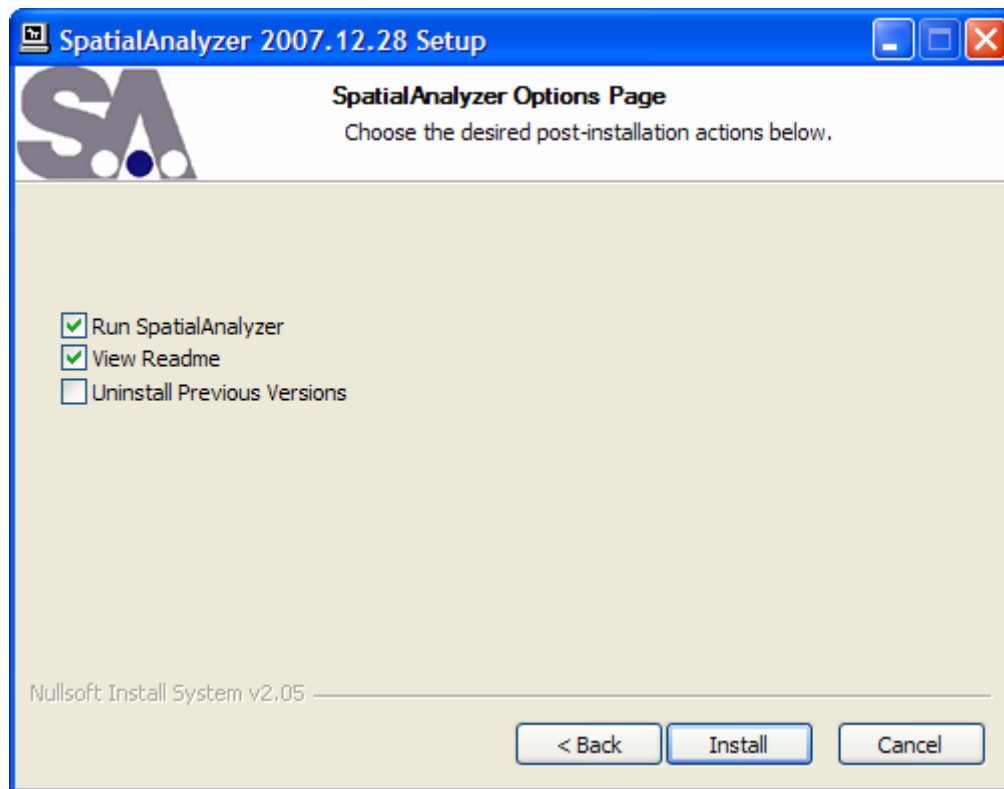
Use this page to indicate where on your hard drive you wish to place the files. You may accept the default, edit the text selection, or browse for a new location in which to store the files.



Installation Destination Selection Page

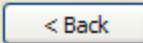
Select the options after the installation is complete

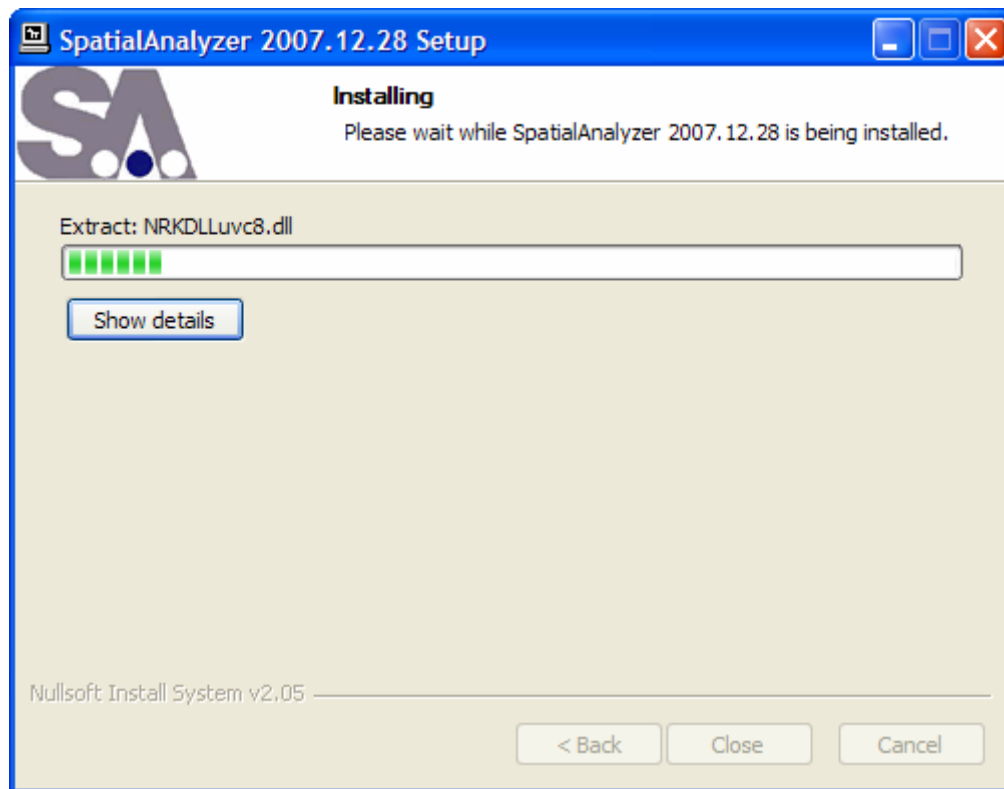
Configure the options after the installation is complete with the Options Page. Control over whether SA automatically starts, the readme file opens and whether all prior copies of SA are uninstalled. Config each option and select the install button.



Installation options page

Installation Page

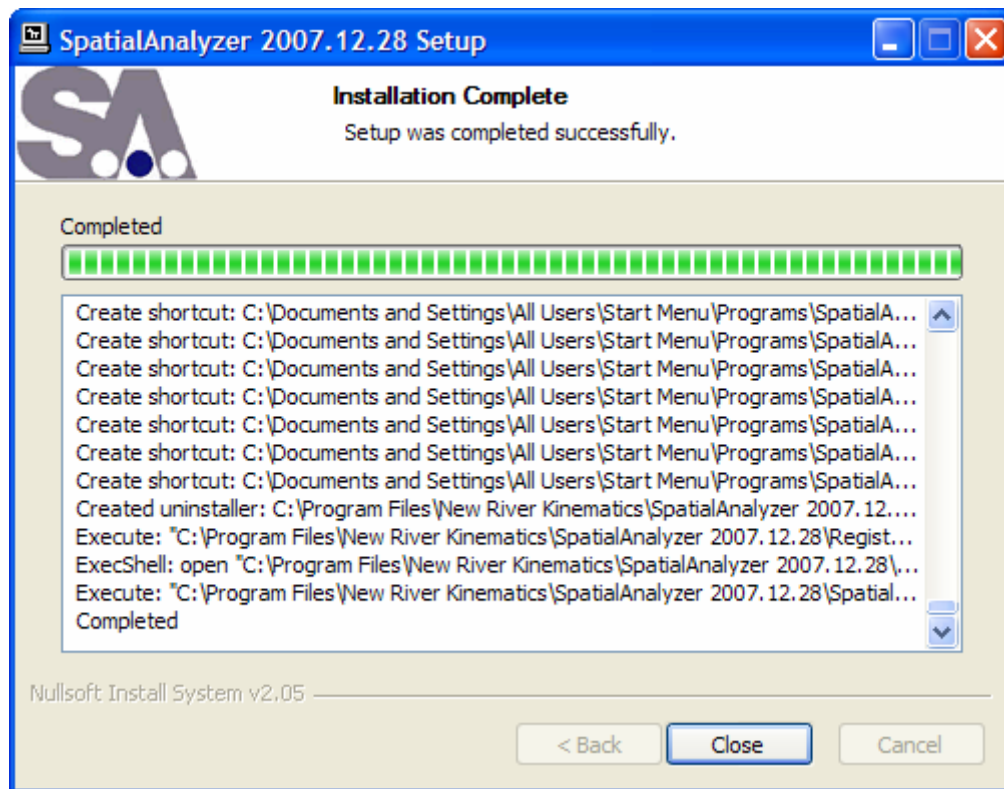
Hit the  button to begin copying **SpatialAnalyzer** files. The progress of the installation is shown in a dialog like shown in the figure below.



Installation progress dialog

Final Installation Page

After the installation is complete a dialog showing the results is shown (see the figure below).

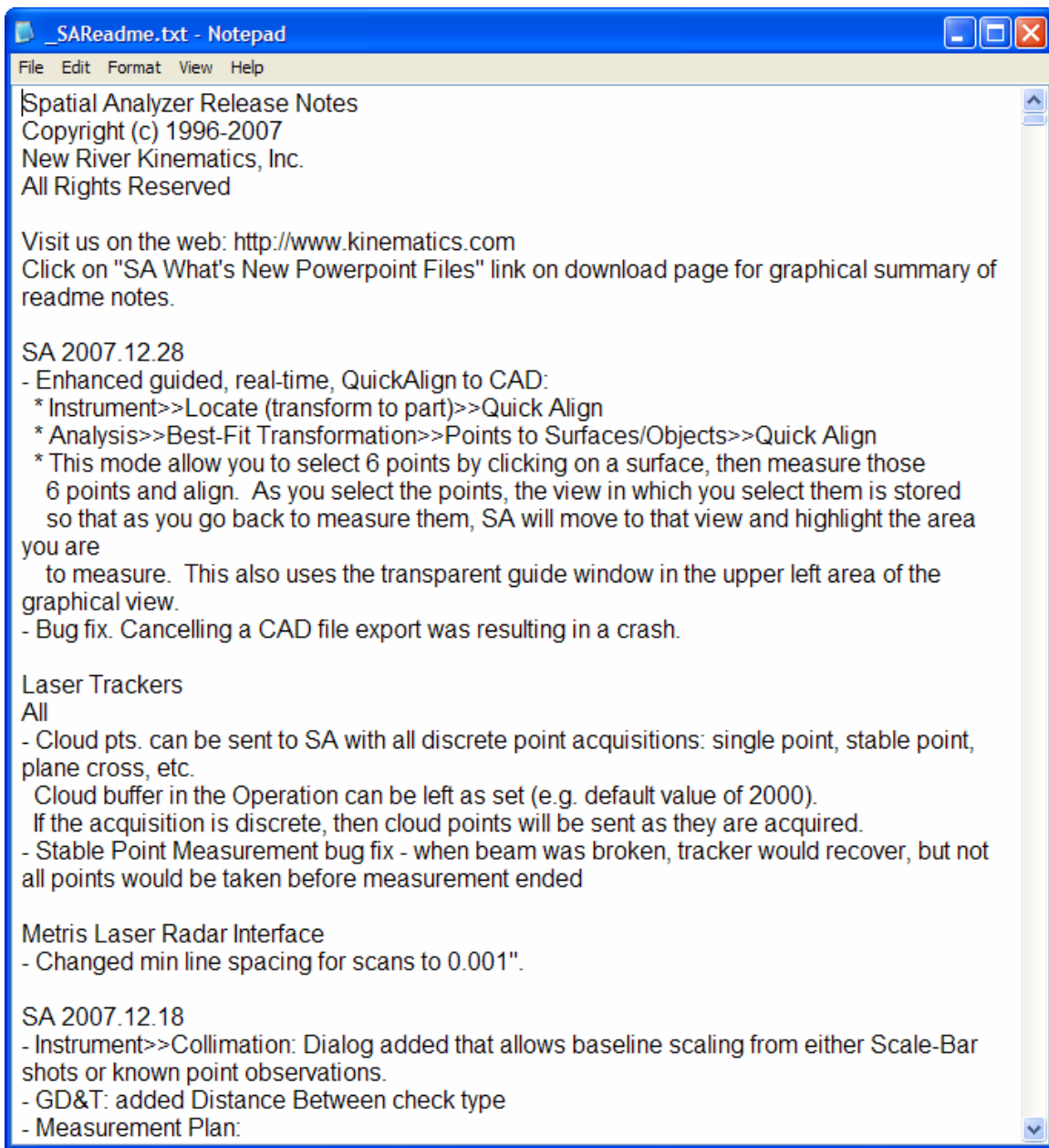


Installation Complete dialog

After the first installation of SpatialAnalyzer on a computer is complete, you will need to install the hardware lock driver. Hardware Lock Driver Installation is covered in a following section.

Readme.txt File

When SpatialAnalyzer is installed a Readme.txt file is placed in the installation directory. Refer to this file to determine what changes and enhancement have been made to SpatialAnalyzer. The file is organized by date and contains descriptions of the important changes since SA was initially released. The figure below shows an example for the 2006.01.11 version of SpatialAnalyzer.



SpatialAnalyzer's ReadMe.txt File

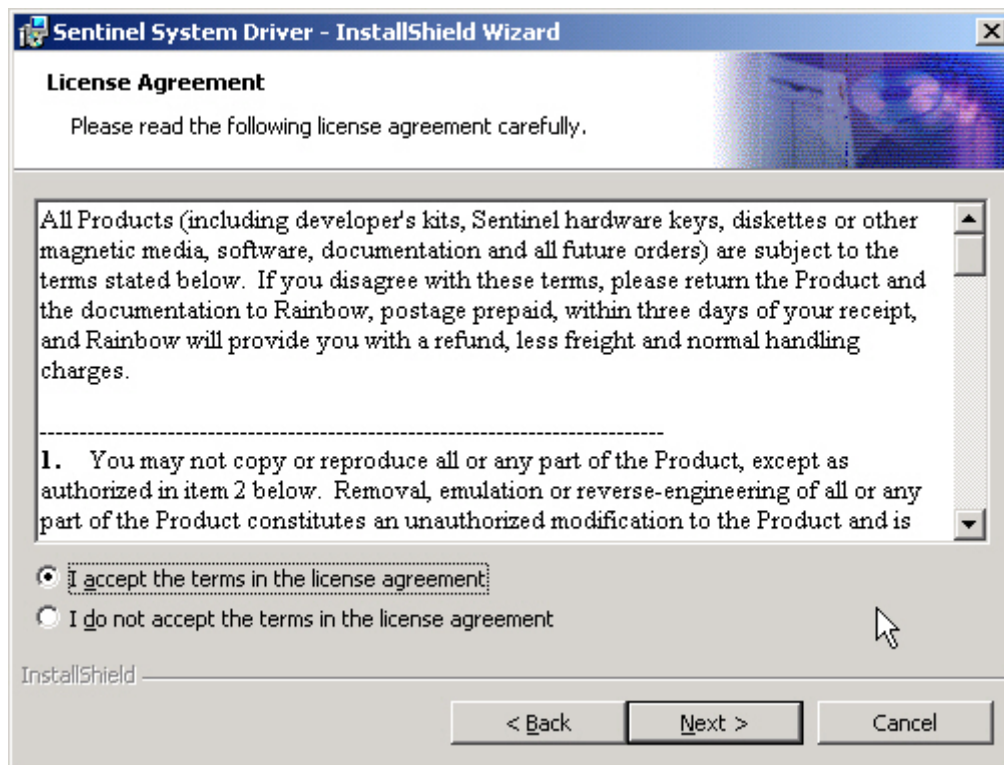
Hardware Lock Installation

Your dongle (hardware lock) will not be recognized and the program will run in the SA Viewer Mode. On the Installation CD (and on the New River Kinematics website <http://www.kinematics.com/>), you will find the SA Hardware Lock Driver Installation program. Copy this file to your hard drive. To start the installation, navigate to the program location double click the file named "HWLock Driver - SSDXXXX-32bit.exe." This will install the hardware lock drivers on your system.



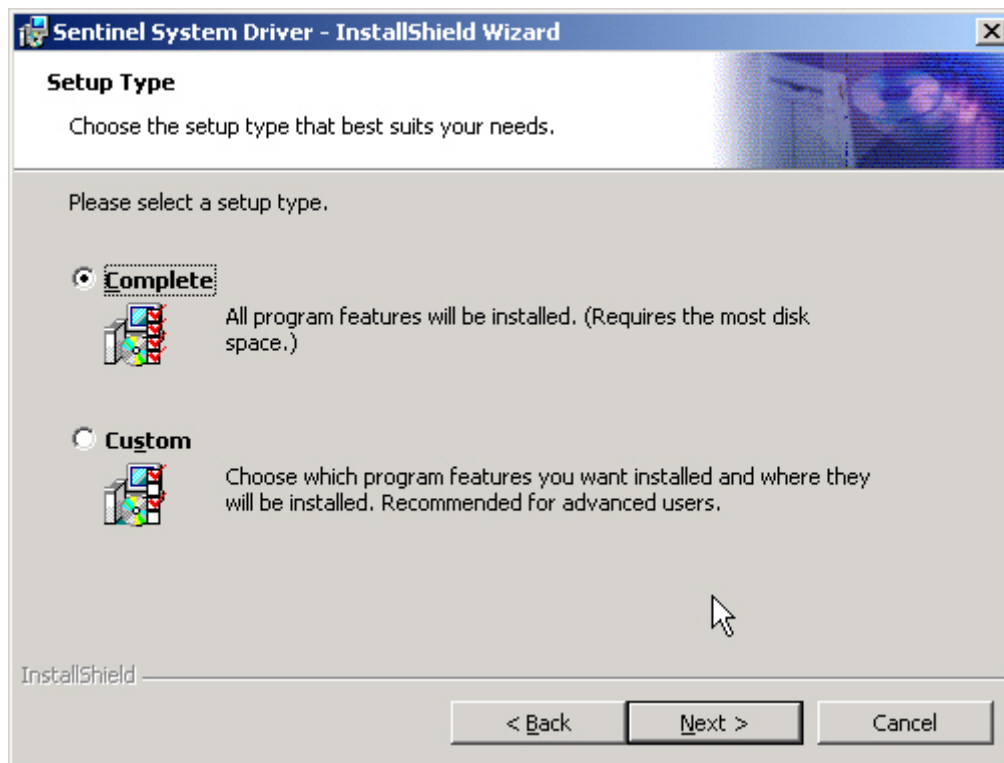
Sentinel System Driver Installation

The first dialog that appears in the Sentinel System Driver Installation is shown in To continue hit the Next > button. The next step covers Sentinel's License agreement.



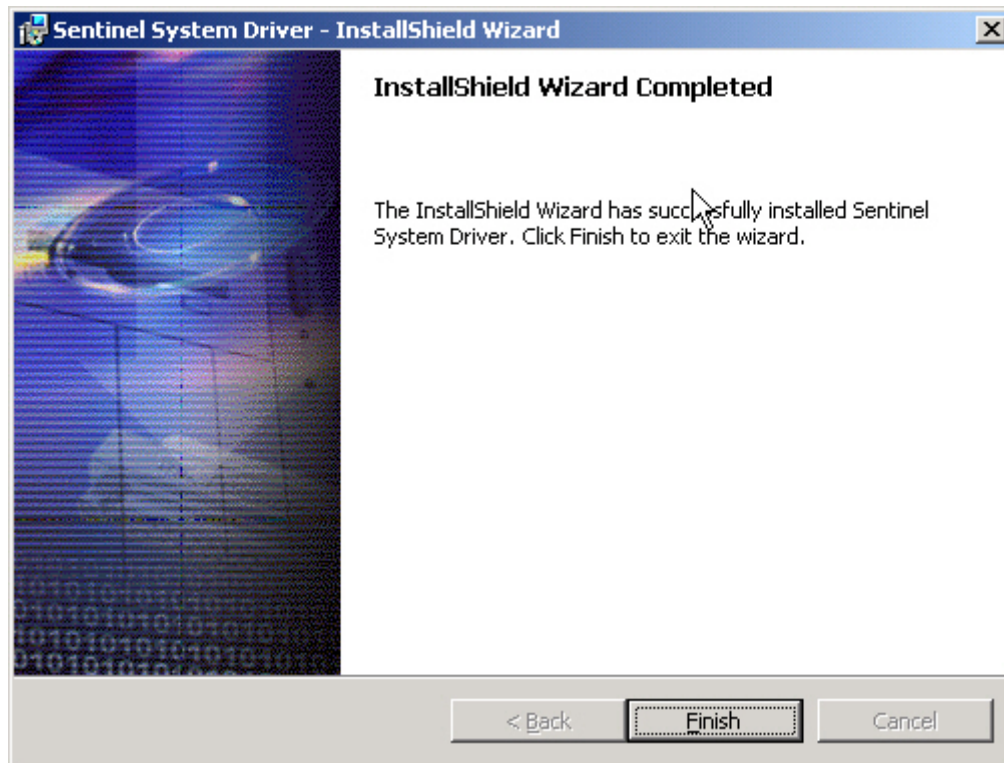
Sentinel System Driver License Agreement Dialog

The next dialog that appears in the Sentinel System Driver Installation is shown in .Please review the Sentinel license terms presented in this dialog and if you find them acceptable, select the "I accept the terms in the license agreement" radio button and then hit the Next button.



Sentinel System Driver Setup Type Selection Dialog

The Setup Type is the next dialog that appears in the Sentinel System Driver Installation. It is shown in . Ensure the Complete Installation radio button option is selected and then hit the Next button. After selecting this option the installation procedure will copy and configure the necessary files into their proper locations on your computer.



Sentinel System Driver Completed Installation Dialog

When the installation complete you will be shown a dialog like that in .Select the Finished button.

Install the Sentinel Hardware Lock on the computer's parallel port.

Reboot the computer.

Run **SpatialAnalyzer** to confirm that the installation was successful.

After the installation is complete you should be able to launch any of the software components by selecting from the Windows start-bar, then , then **SpatialAnalyzer** for the version just installed. As subsequent updates are installed they will be added to this list. Installing a new version of **SpatialAnalyzer** will never remove or otherwise deactivate an existing installation of **SpatialAnalyzer**. With this installation methodology, you never risk losing an operable system due to corrupt installation. After gaining confidence in a new installation, you may remove an older installation using the Windows uninstall feature in the Control Panel >> Add Remove Programs - applet.

Software License Control and Distribution

SpatialAnalyzer offers different management mechanisms for you to control the distribution of your **SpatialAnalyzer** license(s). They are the traditional hardware lock (i.e., dongle) and a software lock key file. Each mechanism has its advantages and limitations. To support your special requirements, **SpatialAnalyzer** can be adapted to

meet your license distribution needs, as these software license control mechanisms can be used together or independently.

A system to distribution licenses from a network server is also available. This mechanism allows users to check a license out when they need one. Please contact New River Kinematics staff for technical support concerning this option.

Installation of the Dongle (Hardware Lock)

SpatialAnalyzer uses the most sophisticated dongle hardware lock available. We have researched alternatives thoroughly and taken all precautions necessary to ensure that these devices will in no way hinder the operation of your computer system.

To install the dongle, remove the cables of any current parallel devices, insert the dongle into your parallel printer port, and reconnect the parallel cables into the back of the dongle. Be certain not to install the dongle backward into any 25-pin serial ports that you may have. If allowed by your computer manufacturer, the dongle may be removed or installed at any time and is not necessary at start up.

Installation of SA Meta-Key file

Meta-Keys are used to upgrade your hardware lock. They are small encrypted files that SA opens and uses to enable capability in your license. By placing a Meta-Key file in your [Analyzer Data\License] directory, you can upgrade your hardware lock. When the file is present, it supersedes the information on the hardware lock. This requires that the .met file always follow the key if it is moved to another computer.

Installing Meta-Key is a simple

- Download SpatialAnalyzer (v2003.06.06 or greater) and the hardware lock driver from <http://www.kinematics.com/>
- Install the hardware-lock driver.
- Install SpatialAnalyzer
- Double-click the ".mez" file and SA will automatically install it. This step requires SA already be installed on the machine).
- Ensure the hardware lock with the matching serial number is on the computer's parallel port. If you're using a USB key then ensure that it is connected to the port.
- Run SpatialAnalyzer
- Go to SA's Help>>Module Licenses to verify your license.

The Meta-Key file must follow the hardware lock if you move it to another computer.

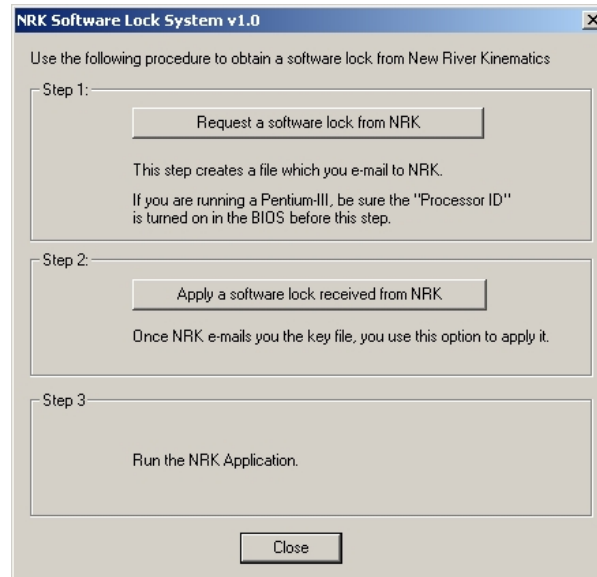
Installation of a Software Lock

SpatialAnalyzer offers Software Lock key files as an alternate software control mechanism. The Software Lock Key file does not depend on a dongle, however it does limit the use of **SpatialAnalyzer** to the particular computer that it is created for.

The process starts by running the NRKSoftKeyRequest.exe program on the computer that will be running **SpatialAnalyzer**. This program is available by request from New

River Kinematics. Please send email to <mailto:info@kinematics.com> requesting the NRKSoftKeyRequest.exe program.

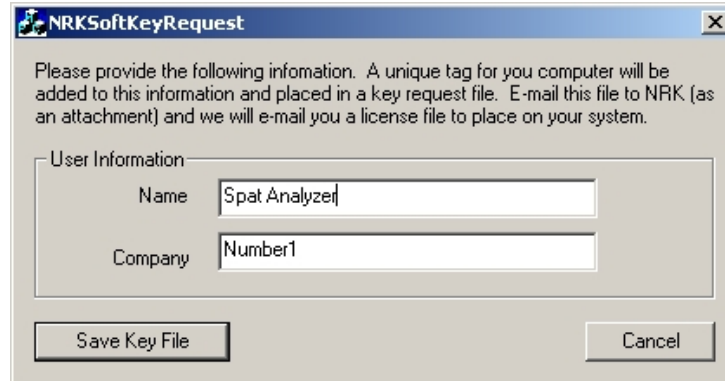
When you start the NRKSoftKeyRequest.exe program, it presents a dialog as shown in the figure below. The process is divided into a series of steps. The first creates a special software key request file (with an nsr file extension) for the computer running the program, and then you have to send that file to New River Kinematics. A software key release file (with an nsk file extension) is then created and sent back to you. With the software key release file the NRKSoftKeyRequest program will then help you apply it. The specific process steps are described below.



NRK Software Lock Dialog

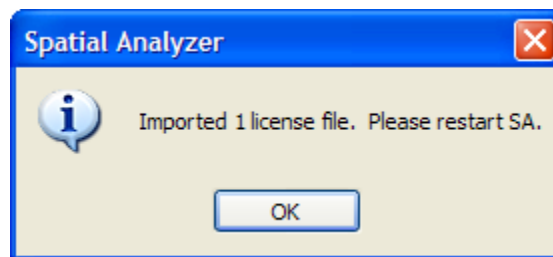
The software lock release process is divided into three steps. In the first step, the program will create a unique key file for the computer running the program. The figure below show the NRK Soft Lock Request dialog that it used to create the file. Two inputs are required on this dialog, user and company information. The software key request file is created when you hit the Save Key file button on the dialog. You are then prompted to name the file. Choose a name that identifies the computer that is running the NRKSoftKeyRequest program. This file will have an nsr file extension. Send this software request key file to New River Kinematics (i.e., info@kinematics.com) and request a software key release file.

The creation of the software key lock file by the staff at New River Kinematics is the second step in the process. Note; the file will release only those modules identified on your software license agreement. The software key release file will be sent back to you and it will have an nsk file extension.

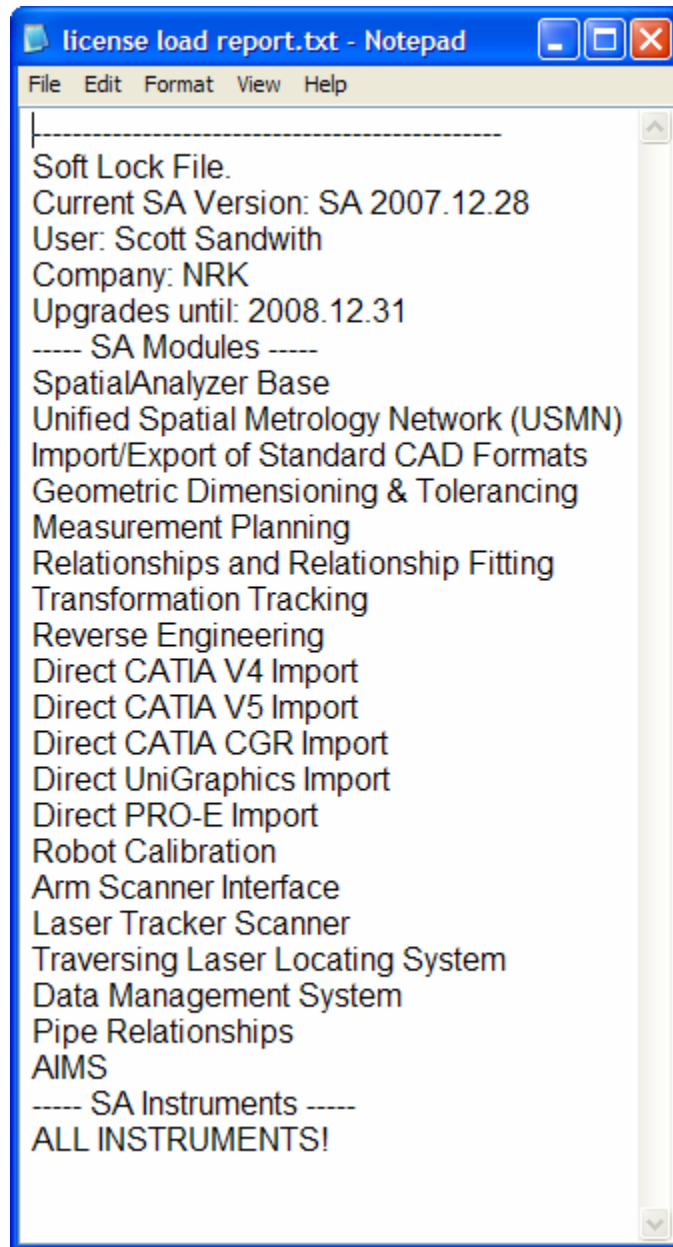


Creating and Saving the NRK Software Lock file

Once you receive the software key file you can move to the third step in the process; Applying a software lock received from NRK. Double click the file on the computer that was used to create the request file and SA will install it for you. Re-start SA to being using it.



After installing the softlock nsk file a text file is opened to reports the modules and instruments available for this computer. The figure below shows an example.



```
-----  
Soft Lock File.  
Current SA Version: SA 2007.12.28  
User: Scott Sandwith  
Company: NRK  
Upgrades until: 2008.12.31  
----- SA Modules -----  
SpatialAnalyzer Base  
Unified Spatial Metrology Network (USMN)  
Import/Export of Standard CAD Formats  
Geometric Dimensioning & Tolerancing  
Measurement Planning  
Relationships and Relationship Fitting  
Transformation Tracking  
Reverse Engineering  
Direct CATIA V4 Import  
Direct CATIA V5 Import  
Direct CATIA CGR Import  
Direct UniGraphics Import  
Direct PRO-E Import  
Robot Calibration  
Arm Scanner Interface  
Laser Tracker Scanner  
Traversing Laser Locating System  
Data Management System  
Pipe Relationships  
AIMS  
----- SA Instruments -----  
ALL INSTRUMENTS!
```

License Load Report

The **SpatialAnalyzer** software license is then setup on the computer.

TCP/IP Protocol and Network Configuration

There are two general classes of network configuration for running **SpatialAnalyzer**: configuration for machines with network hardware, and configuration for machines without network hardware. In the case of machines lacking network hardware and where the user only wishes to run the analyzer and any instrument interfaces on the same machine, the Dial-Up Adapter feature of Windows may be used to provide a TCP/IP address.

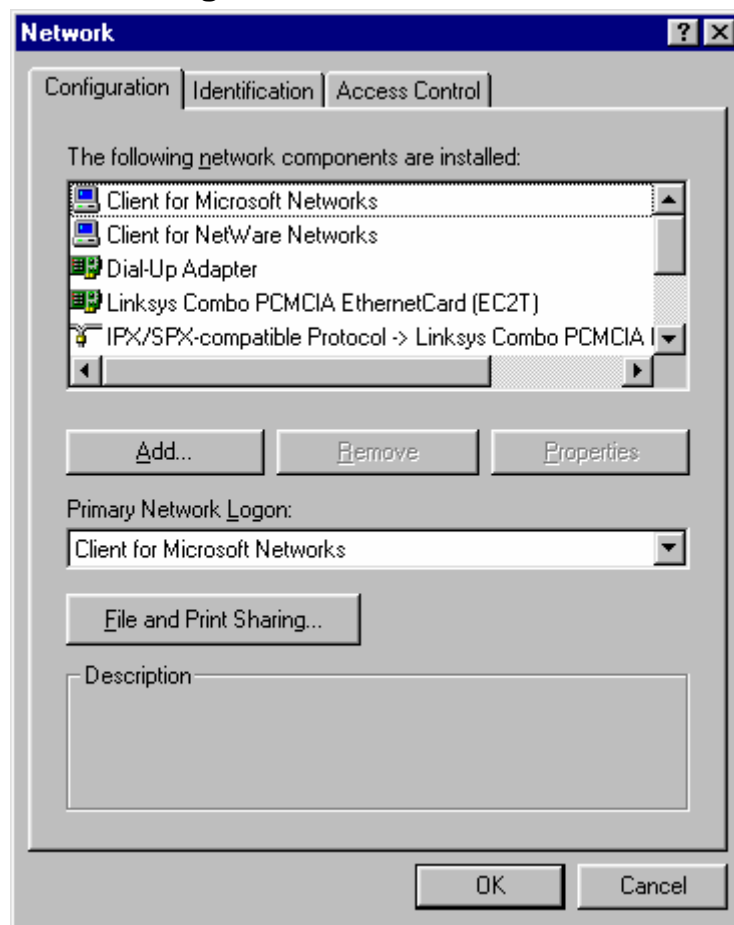
An IP, or Internet Protocol address, is used to uniquely identify each computer on the network. You may check or set the IP address of a computer by going to the Windows Control Panel and selecting the “Network” icon.



Standard Windows Control Panel Network ICON

This will open the network configuration dialog. Depending on the operating system that you are using, you will need to accomplish the following steps.

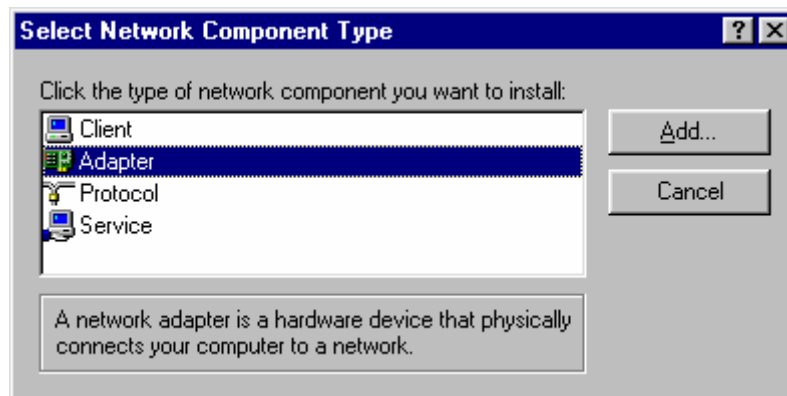
Windows Network Configuration



Windows Network Configuration Dialog

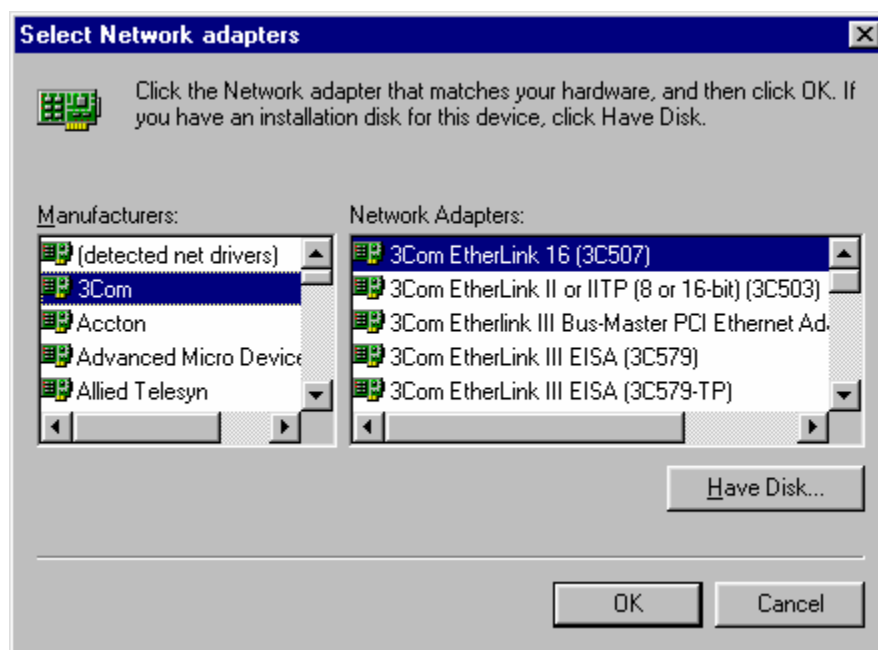
Check with your network administrator to verify the appropriate settings. This section will just illustrate one possible configuration that will function for a stand-alone network between the computer and instrument interface. If necessary, on the Configuration tab, add the appropriate driver for your network card. If the driver is already present, just

proceed to the next step. To add an adapter, hit the Add button. This produces the Add Network Component dialog shown below.



Windows Add Network Component Dialog

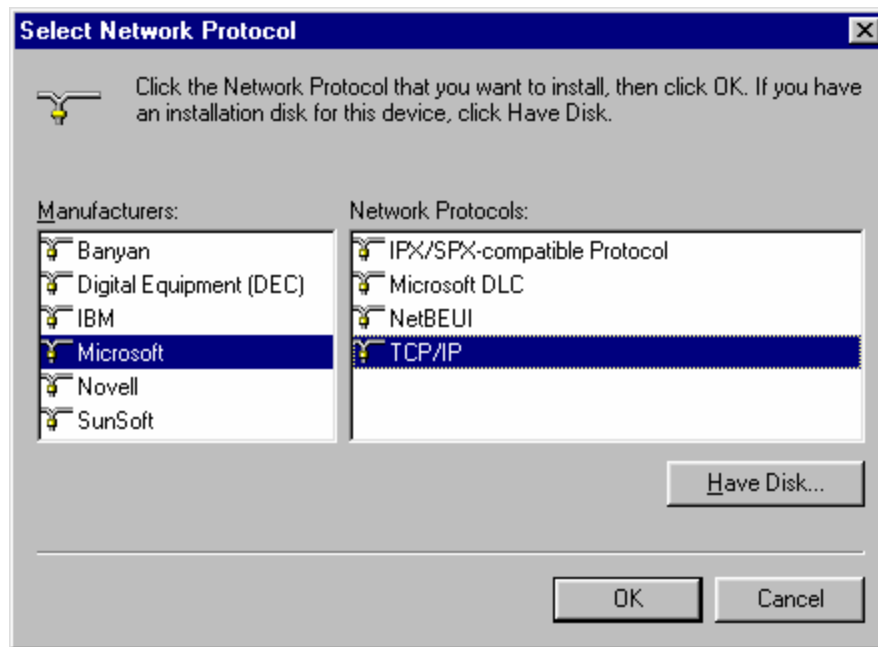
To add an adapter, highlight the Adapter element and hit Add. This produces the dialog shown below.



Windows Select Network Adapter Dialog

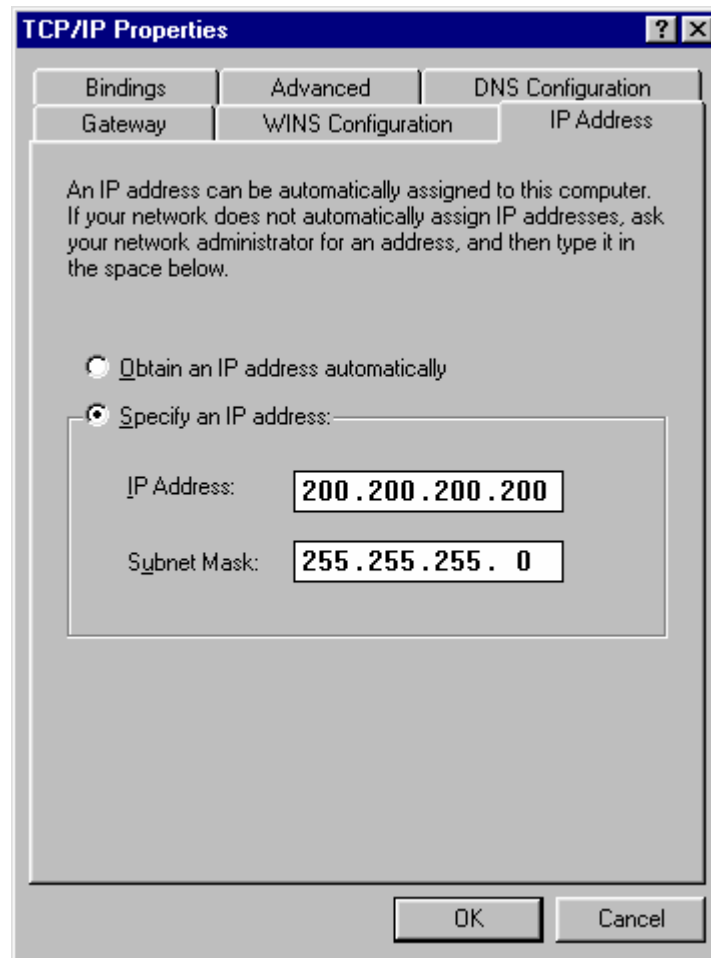
Select your adapter manufacturer and model, and hit OK. Your computer will likely need to be rebooted and may require your original Windows installation disks or CD. If you have no network hardware installed in your machine you can still install a TCP/IP network by installing the Windows Dial-Up Adapter. To do this, select Microsoft as the manufacturer, and Dial-Up Adapter as the adapter hardware. Again you will likely need to reboot and may be prompted for your installation disks or CD.

After the network adapter is installed, return to the Add Network Component Dialog shown in the figure. Highlight the Protocols element and hit Add. This produces the following dialog



Windows Network Protocol Dialog

Select Microsoft as the manufacturer and TCP/IP as the network protocol. Hit OK. Return to the Network Configuration Dialog shown in the figure. Highlight the TCP/IP protocol for your particular network card (Dial-Up Adapter if no network card is present), and hit the Properties button. This produces the dialog shown in the figure.

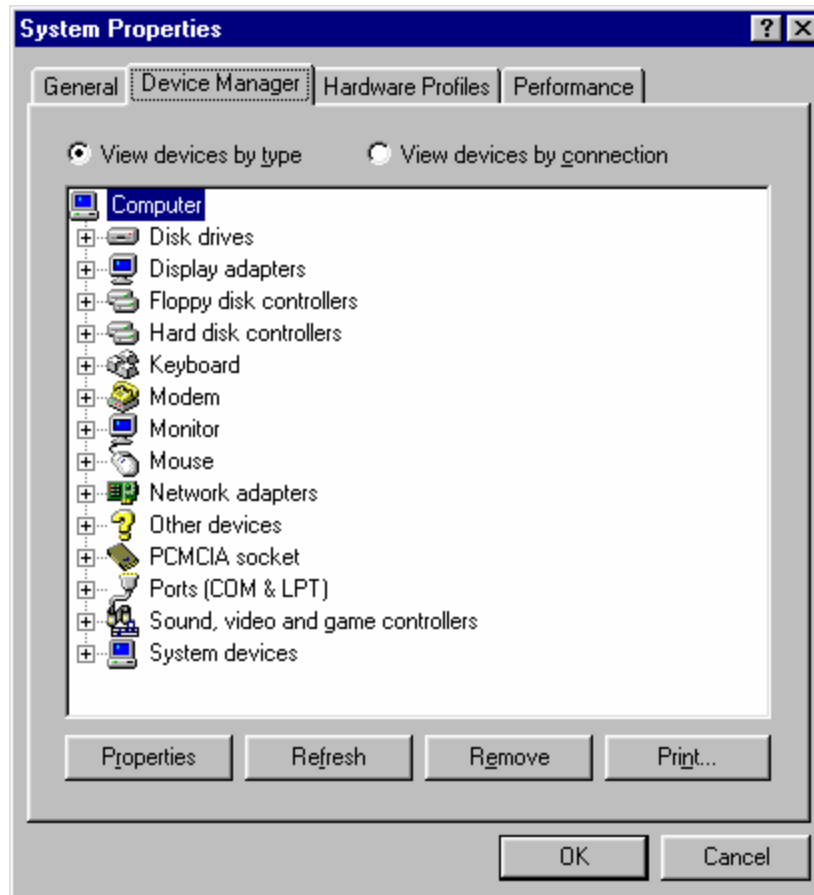


TCP/IP Protocol Properties Dialog

The settings shown should be sufficient if you are not connected to another greater network. If you are, consult you network administrator for configuration information. Reboot your machine, and the network should be ready to operate.

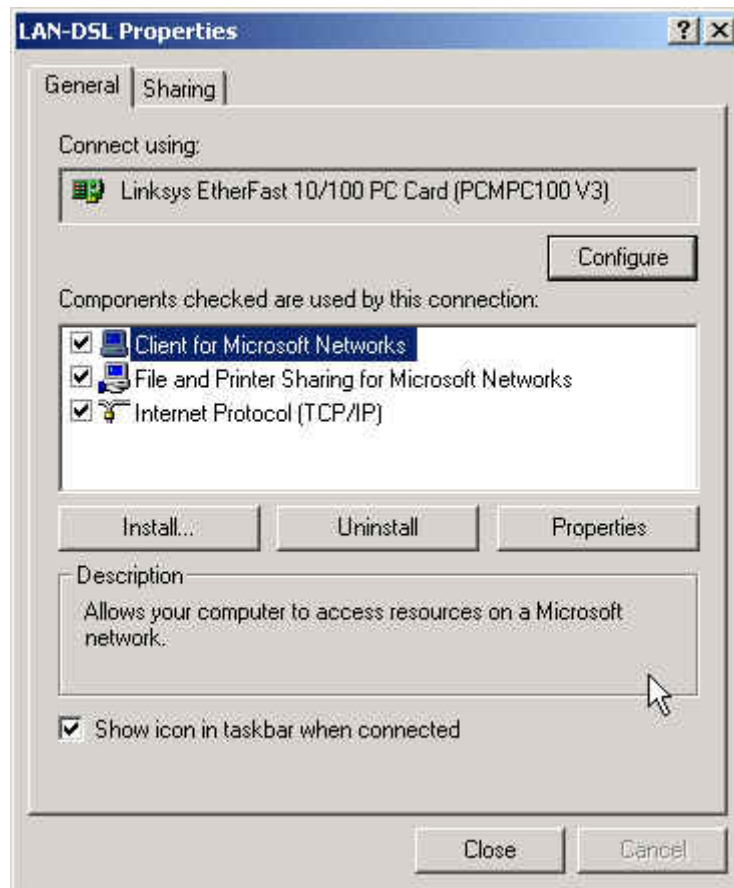
Diagnosing Problems

If Windows detects a problem during the Plug-and-Play start-up phase of rebooting, it will flag any offending devices and highlight the conflicts in the System Properties, Device Manager Page. To access this, right click on the "My Computer" icon and select the Properties option. You will see the dialog shown in the figure. Any problem devices will have an "X" superimposed over the device's icon. To delve deeper, highlight the device and hit the properties button.



Windows Device Manager

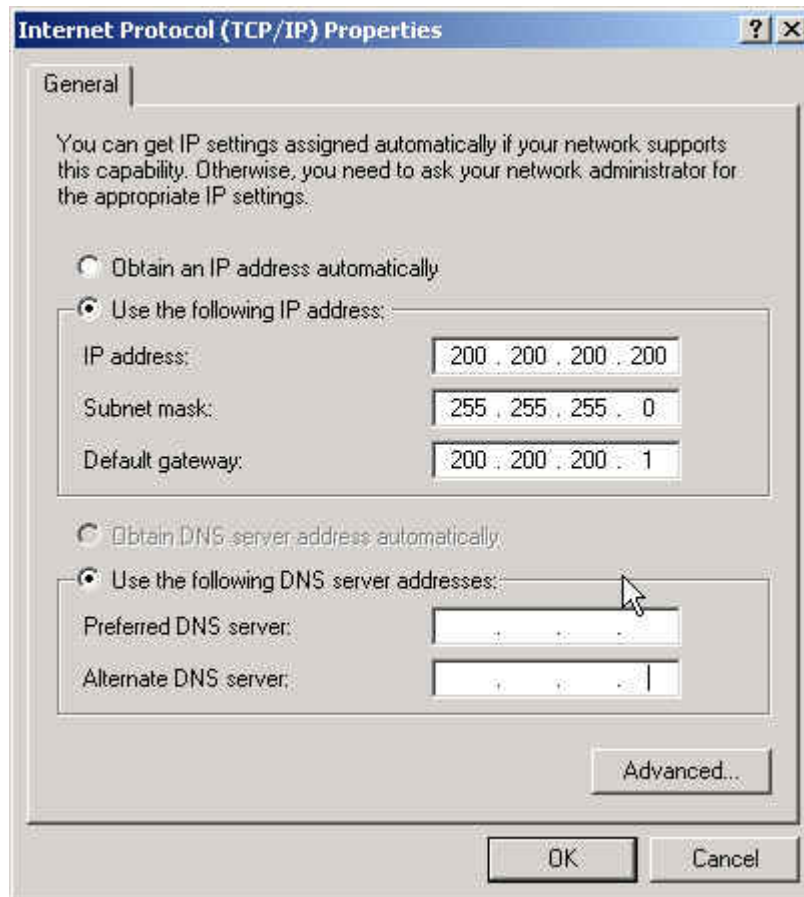
Windows2000® and XP® Network Configuration



Windows Network Configuration Dialog

The figure shows the standard Windows Network Configuration Dialog for Windows 2000. This section will just illustrate one possible configuration that will function for a stand-alone network between the computer running Windows2000 and an instrument interface. We will assume you already have a network adapter configured as shown in the Connect using: panel in the figure below. If a network card is not installed, please follow the documentation provided by the card manufacturer. Check with your network administrator to verify the appropriate settings.

With a properly installed network card, the next step is to install a Network Protocol. The Network Configuration Dialog in the figure shows that the TCP/IP protocol has been installed. This Protocol was added using the helpful installation wizard activated by hitting the Install... button. The process leads you through several easy to follow steps. Setting the properties for the protocol is an important step that requires further explanation.

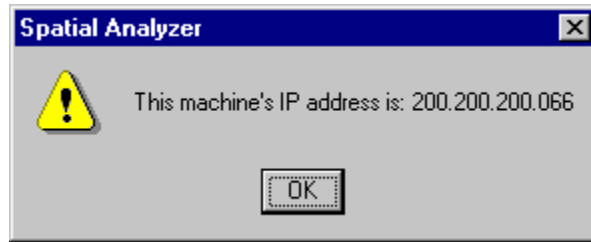


Windows 2000 Network Protocols Tab

First, if your computer will participate as part of an established network (e.g., your company's network), check with the network administrator before proceeding. They will likely have a set of specific numbers that will specify your computer's address within the established network. The typical setup will use a variant on the following example. Make sure the "Use the following IP address" radio button is set. Set the IP address to "200.200.200.200" and the subnet mask to "255.255.255.0". These are just settings that should work for the majority of systems. Switch to the DNS tab and configure it appropriately for your network. If your computer is not part of a greater network then you can skip the DNS tab all together.

Checking the Final Configuration

Make a note of the IP address of the host **SpatialAnalyzer** machine. This can be done from within **SpatialAnalyzer** by selecting the , menu option. This results in the dialog shown in the figure.



IP Address Check Dialog

System Configuration

The following are recommended settings for configuring a metrology computer system. These recommendations are to be used as a starting point for the computer system. Later, once the system has proven to be fully operational, other software may be added if the software does not interfere with the system. Start simple / basic, get the system working, then carefully add any non-standard system software.

- Uninstall ALL firewall software.
- Uninstall ALL antivirus software.
- Uninstall Embassy Trust Suite Software.
- Modify system Boot.ini file to support Large Address Aware applications.

SpatialAnalyzer software is LargeAddressAware. This allows for handling of larger CAD files in XP and Win2000 if your OS is configured for /3GB option in boot.ini.

For instructions on how to edit the boot.ini file.

<http://support.microsoft.com/?kbid=289022#E0IC0ACAAA>

The boot.ini file needs the string "/3GB" in a line that begins "multi(0)disk(0)". A typical example is shown below.

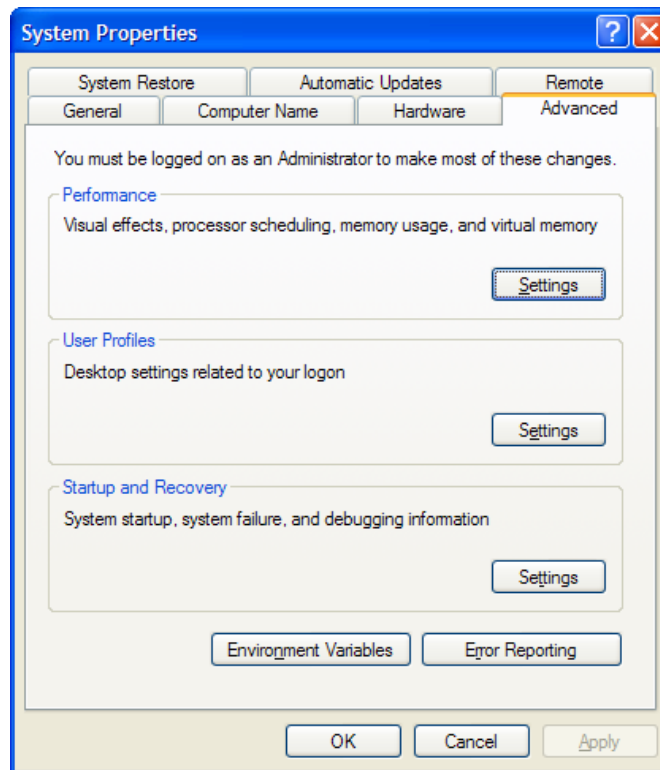
```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(2)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WINNT="???" /3GB
```

To read more about the reasons behind this, see:

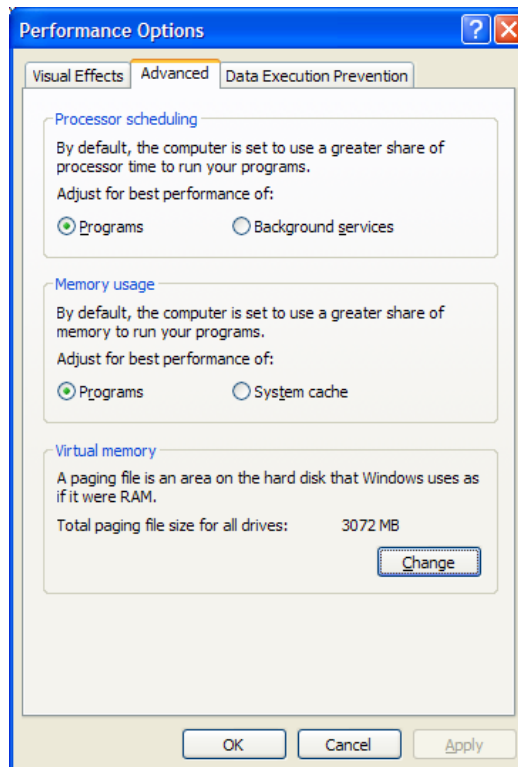
<http://www.microsoft.com/whdc/system/platform/server/PAE/PAEmem.msp>

Virtual Memory

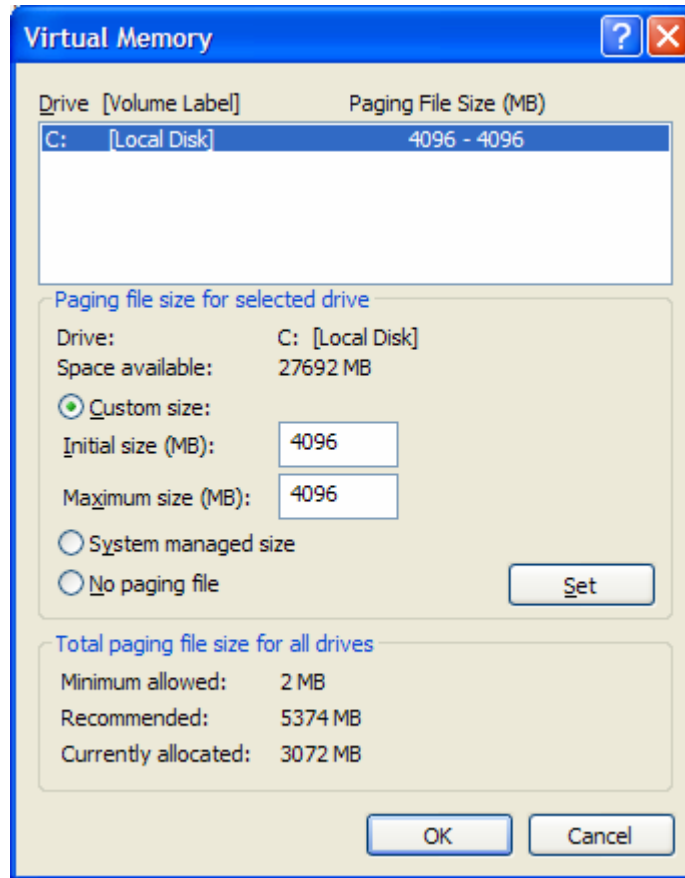
From the Control Panel / System Properties / Advanced tab, select the Performance Settings button.



Performance settings / Advanced tab, select the Virtual memory Change button:



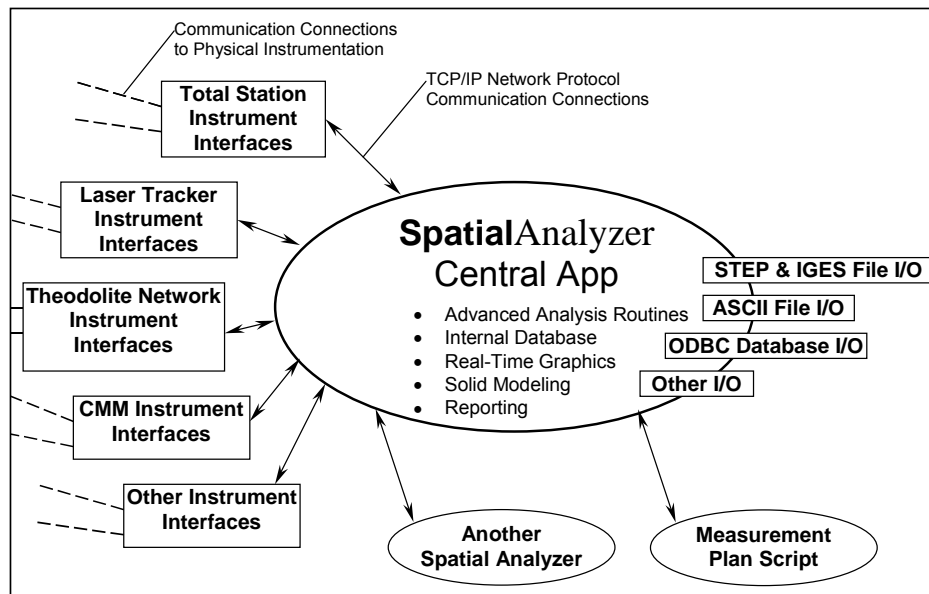
Adjust the Paging file size to become “Custom” with the initial / maximum sizes matching the “Recommended” value or the largest allowable system size:



Reboot the system to apply changes to the available virtual memory.

SYSTEM ARCHITECTURE

SpatialAnalyzer is a flexible and general spatial metrology tool. This chapter will highlight some of the features and illustrate how adaptable **SpatialAnalyzer** is for a variety of spatial metrology configurations. **SpatialAnalyzer** and all of its components were developed using the object-oriented programming concepts. This results in a code-base that is modular in nature. A benefit of this feature to users is that it ensures that **SpatialAnalyzer** will continue to grow without limitations that might otherwise be imposed by the current instrumentation requirements. In short, **SpatialAnalyzer** is designed for instrumentation and analysis both today and tomorrow.



SpatialAnalyzer Metrology System

SpatialAnalyzer System Overview

The figure above illustrates one possible configuration of **SpatialAnalyzer** Metrology System. Note the system is comprised of several components integrated with a reliable communications network. The key components illustrated above are the **SpatialAnalyzer** Central Application, an array of instrument interfaces, various I/O (Input/Output) modules, and another copy of **SpatialAnalyzer**. Other SA sessions can remotely monitor any other SA job within the same network. With **SpatialAnalyzer** you can acquire data simultaneously from as many active instruments as you would like. These could all be one type of instrument (all laser trackers), or could be a mix of instrumentation (2-laser trackers and two portable CMMs).

As illustrated, the instrument interfaces communicate with **SpatialAnalyzer** by a robust TCP/IP network protocol. This affords several distinct advantages. First, these connections are extremely fast and reliable. Second, since this is a “network” protocol, the instrument interfaces may be run on either the same computer as **SpatialAnalyzer** or on another computer on the same network. Third, and perhaps most impressive,

since this is the “Internet Protocol” (IP), the computers running the instrument interfaces could be in the same room, across the country, or around the world. Likewise, if activated, it is possible to remotely monitor a measurement job in progress using this same TCP/IP connection over the Internet or on a local network.

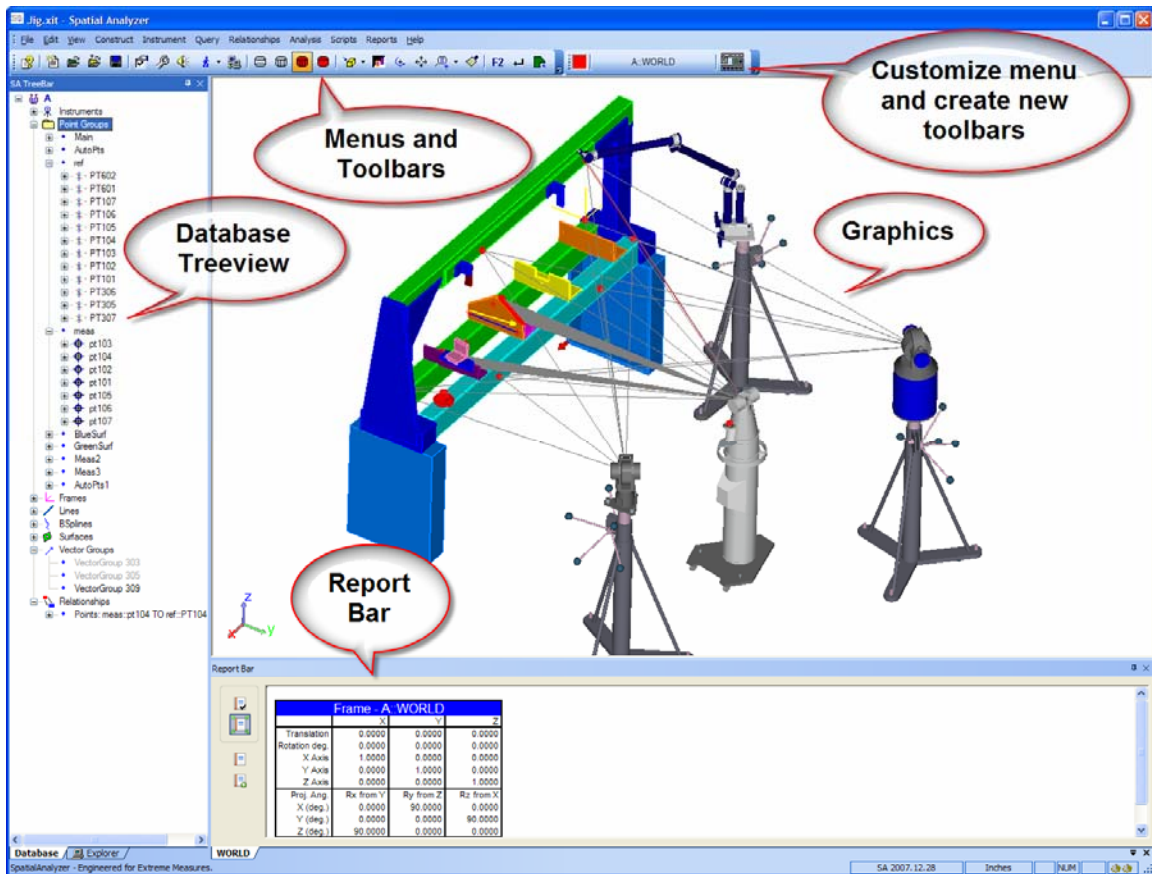
The following sections will describe in greater detail the rolls of each component and the benefits of the selected architecture.

SpatialAnalyzer Central Application

The actual executable code we refer to as **SpatialAnalyzer**, is a modern, C++ (completely object-oriented) application developed specifically for 32-bit Windows operating environments. The **SpatialAnalyzer Central Application** encompasses all of the graphics, analysis, reporting, internal database, and security features of the **SpatialAnalyzer** system. Additionally, the **SpatialAnalyzer Central Application** manages all of the parametric primitives and objects within the internal database.

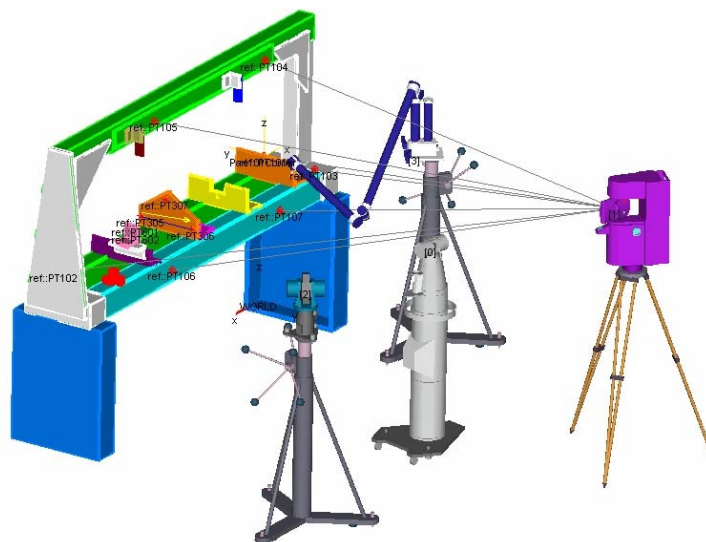
User Interface Features

The figure below shows the **SpatialAnalyzer Graphical User Interface (GUI**, pronounced gooeey). This provides a user-friendly way to access all your measurement data. You may choose any of several methods of access, allowing you to select the best method for a given task. Let's begin our exploration of the **SpatialAnalyzer** architecture with a look at the various objects involved and their relationships. The **SpatialAnalyzer** window is the top-level window in the application. It has a resizing border, caption bar, system menu, and minimizes, maximize, and close boxes. This window has a number of different ways to present the data in **SpatialAnalyzer** to the user. Each of the different presentations is called a view. There are different views including graphics, tree, reports (in a variety of forms), and charts. Each view shows the data in a unique way and together they form a powerful interface for quickly and easily turning the CAD and metrology data into information.



SpatialAnalyzer Graphical User Interface

Graphical View




SpatialAnalyzer Graphical View

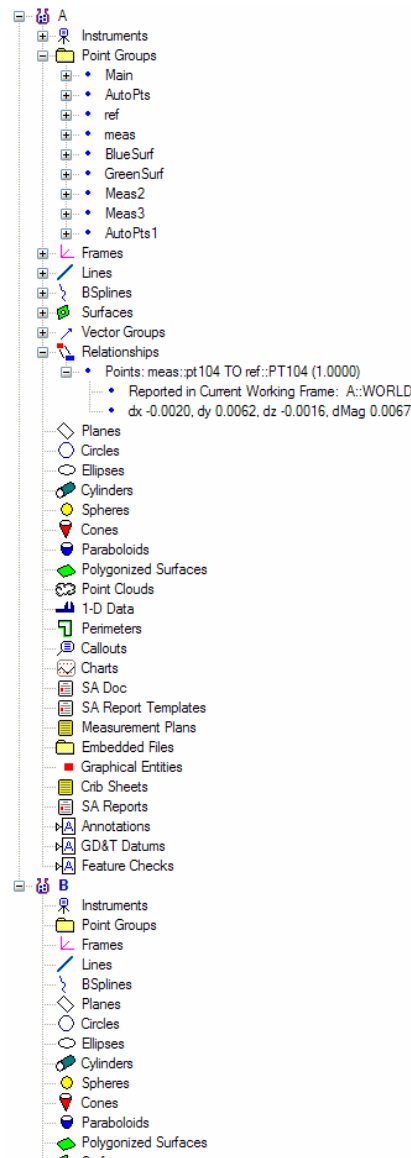
Using the **SpatialAnalyzer** Graphics View you may view the CAD geometry (e.g., solids, surfaces, curves etc.), the measurement sensor(s), their measurements, and the points computed from those measurements all at the same time. The view is a full functioning 3D environment supporting shading, zooming, panning, and rotation of the viewer's perspective. You can get in as close as you need, then back off to see the detail in perspective, and look at it quickly from as many vantage points as you need to understand the information. This presents a practical way to orient a user to the job they are performing. In many situations, a quick glance at the graphical view can confirm proper operation or immediately highlight gross operator errors. It is simple to confirm the results of complicated operations, such as a theodolite and/or tracker network bundle, simply by looking at the graphical view and confirming that this depiction matches the physical set-up in front of the operator. The graphical view is contained in the main **SpatialAnalyzer** window, which houses the basic components that help the user control the view.

Toolbars and Menus

The multi-layered menu system of **SpatialAnalyzer** is configured to somewhat guide the user to the proper functions. The highest level of menus presents encompassing categories of operations and each successively lower menu gets more and more specific. Following standard Windows conventions, **SpatialAnalyzer** Toolbars are dockable and floating. They may be either in the middle of the workspace or attached to the Frame at any point. Toolbars may be activated and deactivated from within the user options pages.

Create customized menus and toolbars with the  button control. A following section covers the details in the User Interface Profiles functionality.

Database Tree View



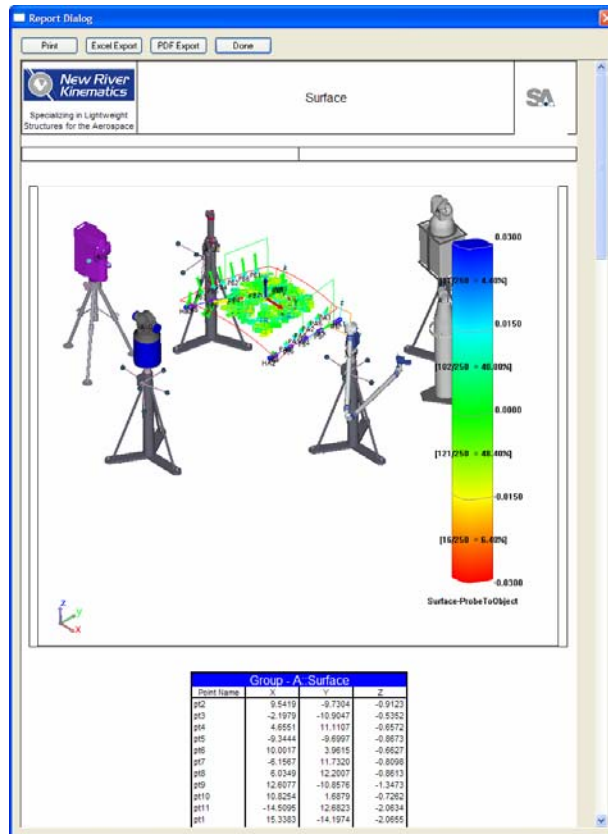
SpatialAnalyzer Database Tree View

The **SpatialAnalyzer** Database Tree View allows a user to quickly obtain data pertaining to a selected feature. In addition, the user may also use this view to control the graphical and other properties of the selected item. Left clicking on any item in the Tree View explodes the associated information contained within that entry. Many Tree entries go several layers deep and present information that could be obtained in other areas as well. Right clicking on an entry presents a menu of functions that enable the user to adjust certain properties or control its graphical representation.

Quick Reports and Report Bar

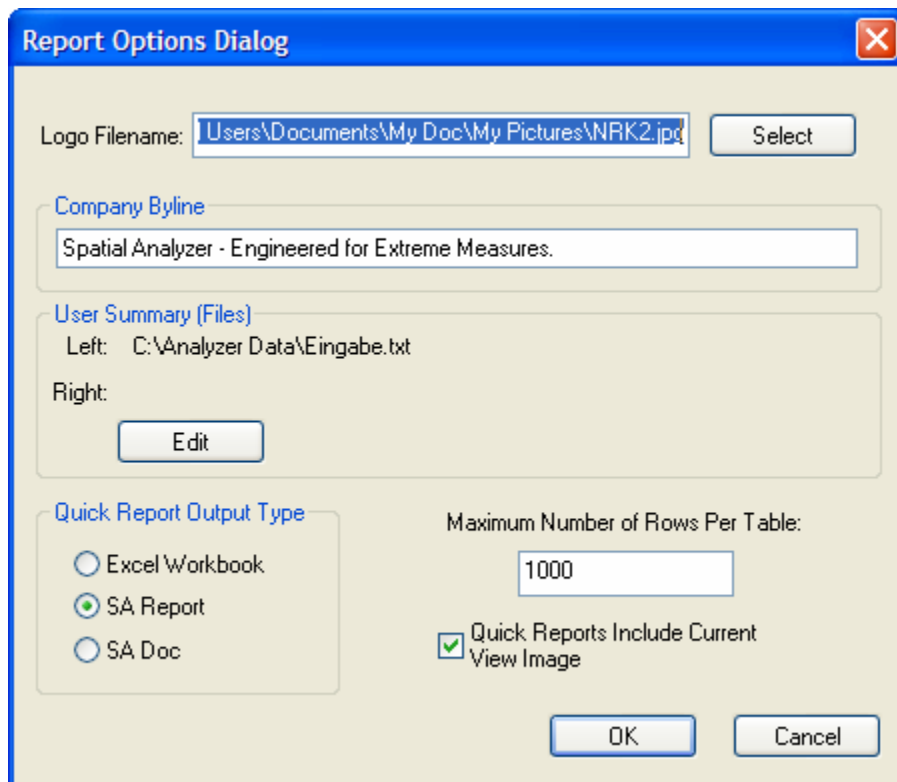
Quick Reports within **SpatialAnalyzer** are saved on the treeview. The Report Bar shows pertinent properties for entities. Each object type has its own Quick Report that presents the

user with a screenshot and a table of parameter values for the specific object. Printing and print-preview functions allow the user to out put reports for archival purposes.



SpatialAnalyzer SA Quick Report

Quick Reports can be configured to output SA Reports, SA Docs, and Excel Spreadsheets. SA Docs are MS Word RTF files that are embedded on the SA treeview. Header images, logo settings, and fonts are user settable from within the Report >> Options dialog.



SpatialAnalyzer SA Report Options dialog

Special User Interface Features

The purpose of this section is to familiarize the user with the special features of the **SpatialAnalyzer** user interface. Many of these are unique to **SpatialAnalyzer** and are unfamiliar to even experienced Windows® users. A few minutes spent learning these methods now will save much time during regular use.

Cursor Prompting

One of the innovative features of the Spatial Analyzer is the extensive use of cursor prompting. Anytime there are a series of operations or selections that the program requires of the user, a prompt will appear and follow the cursor around the screen. This prompt will contain instructions as to the action that the user should follow. Once the action is complete, the next step of the operation will appear. This provides a user-friendly means of obtaining complex series of information from the user.

During a cursor-prompted operation, there are several operations the user may perform to backup or assist in the operation. These are as follows:

Abort: If you wish to end the current function, press the ESCAPE key, and the prompt will disappear.

Pick Points or Objects from a List: When you are prompted to select multiple points or objects from the model, you may press F2 and a selection dialog will appear. Using this dialog, you may include or exclude points from your selection using a list format.

End Multiple Selections: When you are being prompted to select multiple objects or points, press either ENTER or SPACE to end the prompting sequence, indicating you have selected all of the entities you are interested in.

Mouse Functions

The mouse may be used to change the point of view that the user has on the model. By moving the mouse in combination with its buttons and keyboard keys, you may perform the following functions:

Rotate the viewpoint: Hold down the RIGHT button on the mouse and drag across the screen. The model view will rotate about the current working frame. This works for both side-to-side and up and down motions. At first, many people are uncomfortable with this method of view rotation. We've found however, that most users after only a half-hour of regular use prefer this method to the point of view dialog. A view port dialog is also available as a selection under the View Toolbar drop down menu.

Spin the viewpoint: Hold the Control key and right mouse button down to spin the view point about an axis normal to the current screen.

Zoom view using a rectangle: Move the mouse to one corner of the zoom rectangle, hold down the LEFT button and drag the rectangle to encompass the portion of the model you wish to see in more detail. When you release the left button, the view will zoom in. Press the toolbar button or select , to zoom out.

Zoom view with the Mouse Wheel: When your mouse cursor is in the graphics view the wheel on your mouse can be used to control the zoom of the view. Rolling the mouse wheel back zooms out. Rolling it forward zooms in.

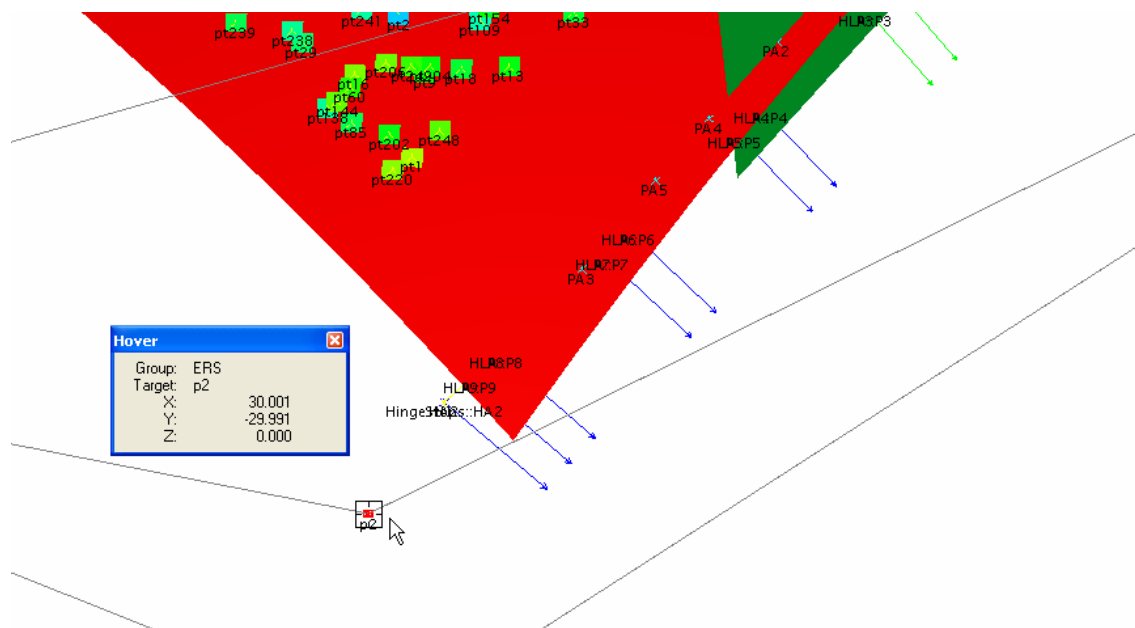
Zoom view: Hold down the CONTROL key and the LEFT mouse button. Drag the mouse up and down on the screen, and the view will zoom in or out based on the mouse motion. You may also use the PgUp and PgDn keys to control this operation.

Pan the view: Hold down the SHIFT key and the LEFT mouse button and drag the mouse. The view will pan as you move the mouse. You may also use the arrow keys to pan the view.

Double-left-click centers the view: Double clicking with the left mouse button in the graphics will center that spot in the graphics view.

Hover Point view

A convenient way to determine a point name and coordinates is with the Hover point tool. To activate this mode either select it from the menu [View >> Hover (nearest point)] or by hold down the control key and the "H" key at the same time.



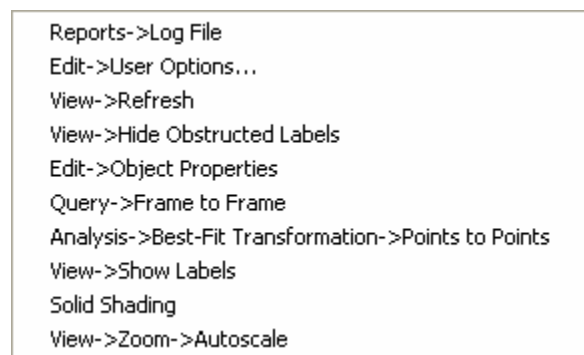
Hover Point view Control

The point properties closest to the cursor are displayed in the window and a square box centers on the point in the graphics view. These two things make it easy to quickly determine which point is which in the graphics view.

Repeat Command(s)

SA has a hot key to help when a command needs to be repeated. To repeat the last command hot the control key down and then select the Tab key on the key board. A menu choice is also available under the Scripts menu.

A history of commands is also available by holding down the control, tab and shift keys all at the same time.



Example Command History List

Toolbar










The Spatial Analyzer makes extensive use of Windows toolbars. These toolbar buttons provide shortcut access to the menu options contained within the system. The toolbars are organized into commonly used and related groups of commands.



SpatialAnalyzer Toolbars









If you move the mouse over any of the toolbar buttons and hold it there for a brief moment, a caption box will appear summarizing the function of the button. In addition, the status bar in the lower left corner of the **SpatialAnalyzer** window will give a more detailed description of the button. By default all of the toolbars are visible. They can be moved and rearranged into an order that you find convenient.

The toolbars are as follows:

Main Toolbar contains commonly used functions for opening new and stored jobs, and saving the current job. It also contains buttons for accessing the SpatialAnalyzer Manual , new job , open a job file , save a job file , user options dialog , object properties , **SpatialAnalyzer** Speech Server , run instrument interface , and a dialog showing all the online **SpatialAnalyzer** network accessible components .



SpatialAnalyzer Main Toolbar

Shading, Background, View, Callout, and Set Object Color Toolbar helps you quickly access a number of different functions. The   buttons allow you to control the shading of objects in the graphics view; they are Wireframe and Solid views respectively. Switching, saving, and setting views is accomplished with the  button. The  button toggles the background color in the graphics view between black and white. Controlling the graphics view rotation center and auto-scaling the view port are accessed using these   buttons. Creating, editing, and storing **SpatialAnalyzer** Callout views is done using the  button. Object color can be set with the  button.



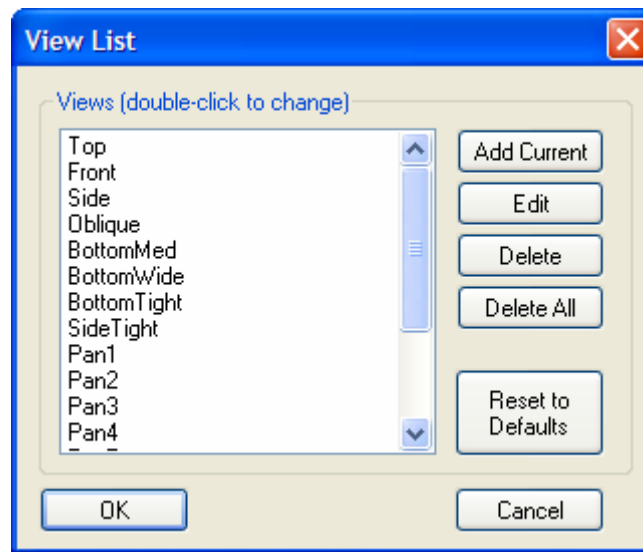
SpatialAnalyzer Shading-Background-View-Callout

View Toolbar allows you to change the graphic's view perspective to a number of commonly used orientations, e.g., Front, Top, Side, and Oblique. The different views appear on a Windows drop-down list that appears when you click on the down arrow on the right side of the control.




SpatialAnalyzer View Drop Down List

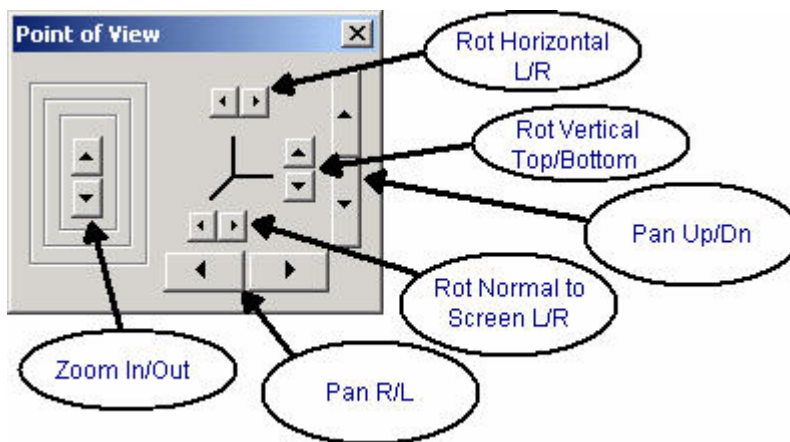
The View Toolbar allows you to name and save particularly useful views for the job. Clicking on the view button on the tool bar brings up the View Settings Dialog as shown in the figure. Once a view is saved, it appears on the pre-defined views list with the name you gave it. Coming back to a view that you need to use to select objects is easy by selecting it from the list of views. You can edit the names of each of the views, even the pre-defined ones. So controlling what you consider to be the front, side, and top view of the objects is available.




SpatialAnalyzer View Setting Dialog

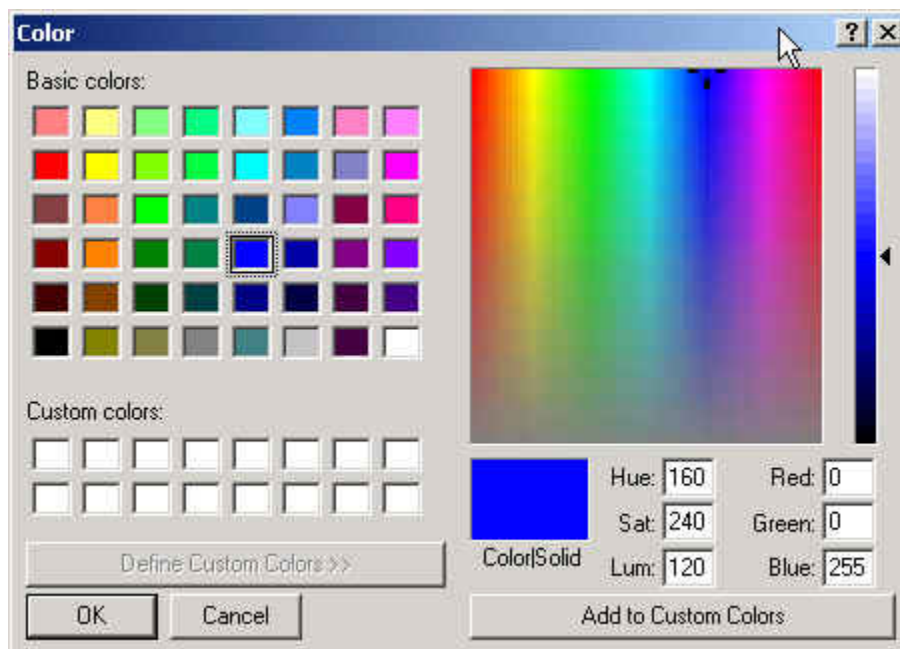
Precisely controlling the view perspective is supported in the View menu using the Point of View Control dialog. Specific incremental adjustments (e.g., rotating 5 degrees about the X-axis², etc.) can be made to the view using this dialog.

² The translation and rotation increment is set in the User Options Dialog's , Units tab.



SpatialAnalyzer Point of View Dialog

Working Color Selection is set using the active color toolbar button . The figure below shows the current working color selection dialog for the open job. It is set with a standard Windows color palette selection dialog. The standard palette control that Windows provides allows you to select the desired color in at least five different ways. They are, select a basic color, select a custom color that has been previously set, click directly on the palette to choose the color, set the Hue, Saturation, and Lumens (i.e., brightness) parameters and to set the Red, Green, and Blue component that form the resultant color. By selecting a different color, new objects and points will be shown in the graphics view using that color.



SpatialAnalyzer Working Color Selection Dialog

Working Frame Toolbar shows the current working coordinate frame as the label for a button on the toolbar. By clicking on the button, an easy to access list of the available coordinate frames is shown.

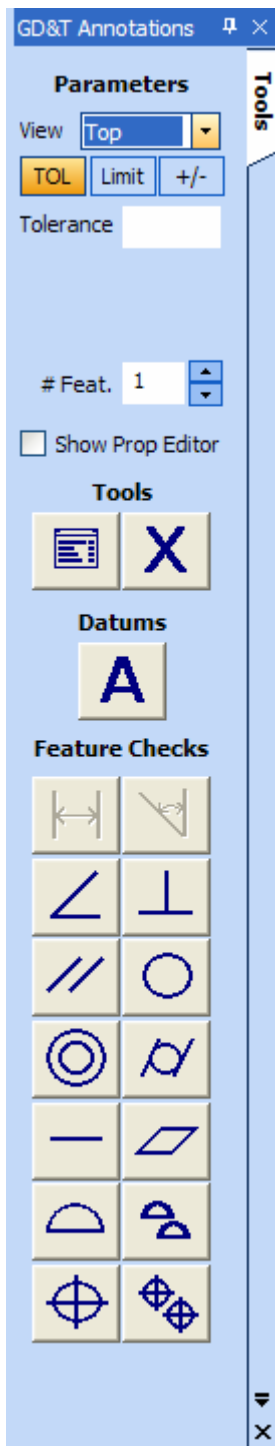


SpatialAnalyzer Working Frame Toolbar

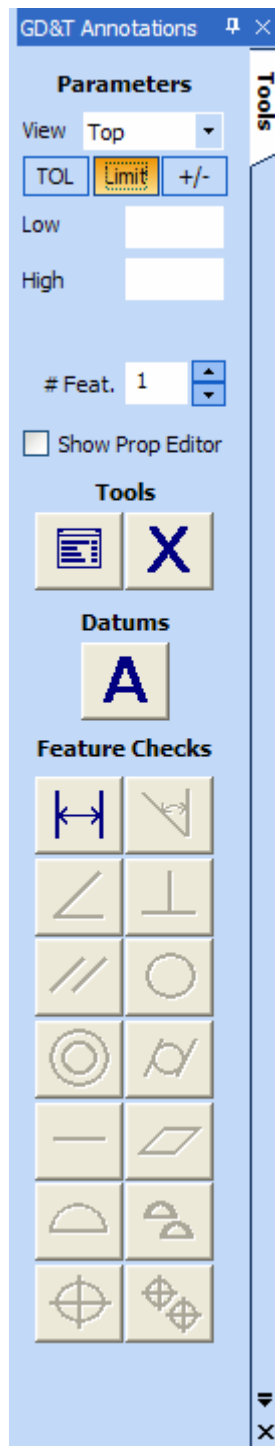
SpatialAnalyzer allows an operator to manage an unlimited number of coordinate systems. Coordinates may be presented and spatial relationships queried with respect to any of the established coordinate frames. By selecting a different frame in the Working Frame Toolbar, reports and analysis output will be presented relative to this frame.

GD&T Annotations Toolbar

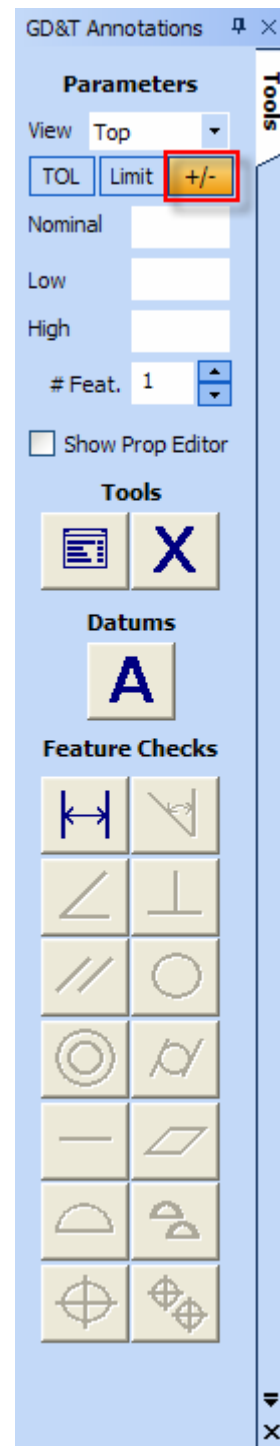
Use the GD&T Annotations toolbar to create datums, annotations, feature checks, limits or simple +/- tolerances. There are a wide



Tolerance Panel



Limit Panel



+/- Panel

The GD&T Annotation toolbar support customized annotations with the “Show Prop Editor” feature option.

Geometric Objects within SpatialAnalyzer

SpatialAnalyzer's internal database manages and maintains an array of parametrically defined geometric primitives. The following discussion pertains to the internal **SpatialAnalyzer** database. This should not be confused with either the external databases supported by **SpatialAnalyzer** or the other "utility databases", such as the , and databases.

Most database entity types are classified as objects. For example are all objects. You can edit the properties of these objects by selecting the edit object function or by right clicking the object in the Database Tree View.

There is a distinction designed to make manipulating large amounts of data easy. A group is an object containing an unlimited number of points or targets. Both points and targets can belong to the same group.. With this distinction in place, you may perform certain operations on a group basis, e.g., set a group's color, move a group of points together, select a group of targets, etc. This seems natural for many operations. For example, we may measure a group of points all lying on a paraboloid and place them in a group named "Parabolic Dish". Now we may fit a paraboloid to this group rather than having to select each individual point. Grouping points and targets allows you to organize the data into logically consistent sets. When you need to perform specific functions on only this set of points or targets it is easy select those and only those points and targets.

An instrument object contains the basic information and properties for each instrument type and it placed within the workspace. Measurements are used to store the observation information from the instruments.

Every entity with **SpatialAnalyzer** has properties that are stored in the internal database. For example, a circle has a center point, radius, normal, and color. A measurement has the type of shot, and the actual shot data (e.g., az, el, dist, date, time, temperature, pressure, user, measurement reporting parameters, target offset, etc.).

The Special Case of Groups, Points, and Targets

The Spatial Analyzer treats measured data differently than constructed data. A Target is defined as a point that is determined directly from measurements. Points are similar to targets but are either created by the user, imported from a CAD file, fit to data, or generated by moving targets. This architecture is enforced to prevent the user from misinterpreting information.

In addition, both Targets and Points are stored in Groups. These Groups may contain any number of Points and/or Targets. The purpose of this grouping is to allow the user to organize data. It is important to note, however, that users who are not interested in using the grouping functionality may simply place all points and targets into a single group. Refer to Chapter 4 for more information regarding groups, points, and targets.

Instrument Interfaces

The modular instrument interfaces are the basis of the data acquisition aspect of **SpatialAnalyzer**. These provide a easy to use and friendly interface for the control of instruments during a measurement job.

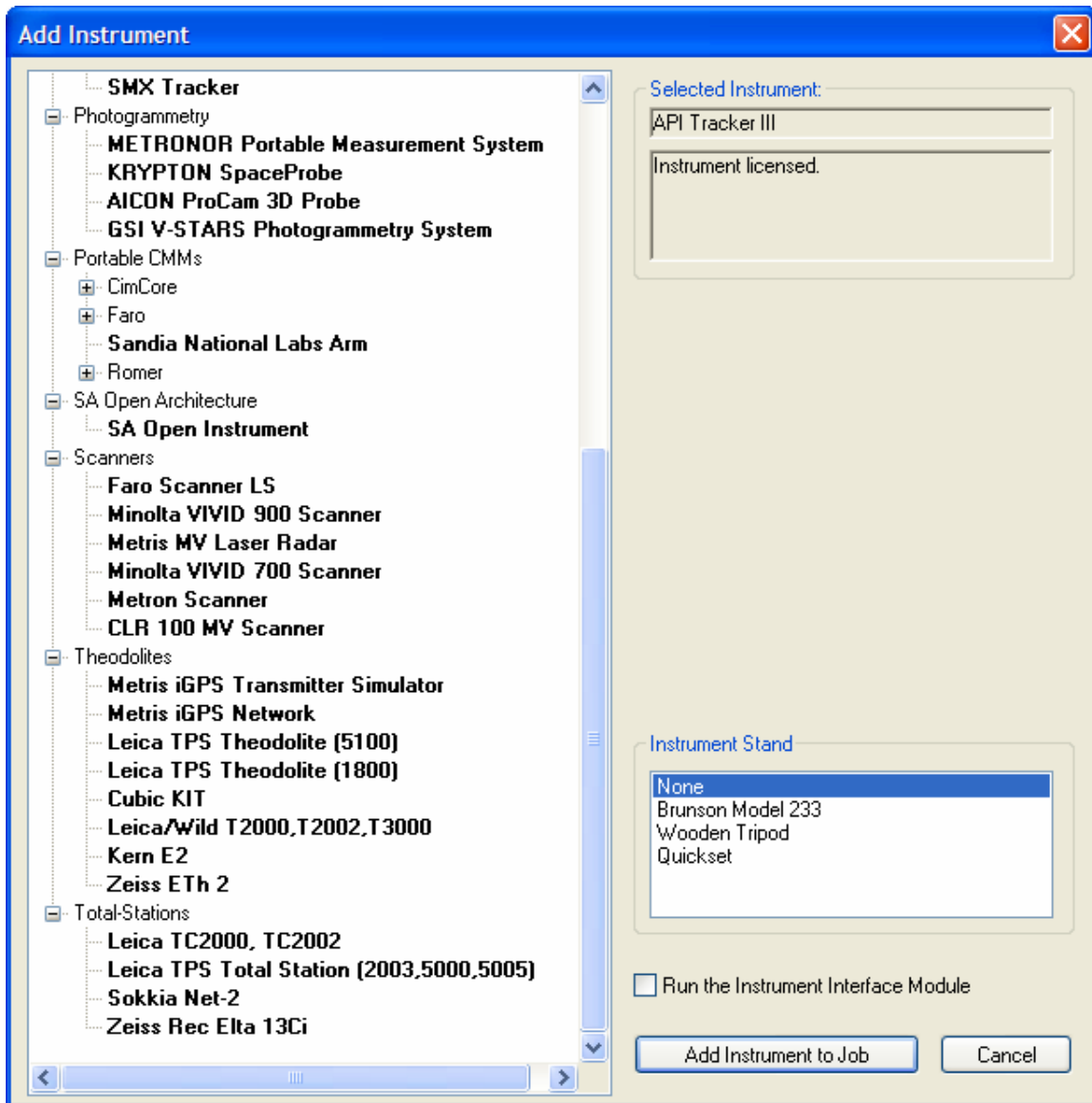
Establishing the instrument interfaces as separate entities in the architecture affords several key advantages. First and foremost, this provides flexibility to adapt to tomorrow's instrumentation. The number of instrument interfaces available for **SpatialAnalyzer** is ever expanding, much like printer drivers and other peripheral devices for the Windows operating system. Second, this approach enables us to provide a common interface for each type of instrument. Thus a portable CMM manufactured by Company A would have the same top-level interface as a portable CMM manufactured by Company B. This allows users to easily select the instrumentation best suited for a particular task. It also allows users to easily transfer the expertise they have gained with one instrument to another. Thereby improving their flexibility when moving from task-to-task and reducing the training time for a user to become productive on a different instrument. Third this allows for a more objective view when it comes time to replace or purchase new hardware since re-training is minimized by the common instrument interface in **SpatialAnalyzer**.

In all cases, the instrument interfaces perform four key functions.

- They provide a common and simple user interface that can be used under **SpatialAnalyzer** control to guide a user through a particular measurement task.
- They communicate over a TCP/IP network to convey essential measurement data to **SpatialAnalyzer**.
- They communicate via a native format to the instrument. For example: serial RS-232 connections for many theodolites, Network protocols for many laser trackers.
- They provide an efficient means to collect the raw measurements, the conditions and techniques under which they were collected, and to perform operation checks on the instrument to establish that it is meeting its performance specifications. It also supports simulating measurements and measurement tasks before the job is performed. By simulating a measurement process before the job is perform, accuracy and time estimates can be made to help you make the best choices relative to optimizing the placement of the instrument(s) relative to the object.

Starting an Instrument Interface

An operator starts an instrument interface by going to the Instrument menu item in **SpatialAnalyzer**. Select the first item on Instrument menu; Add... A dialog of all the instruments interfaces is shown in the figure below.



SpatialAnalyzer Add Instrument Dialog

Select the instrument that you want to work with from the list on the left of the Add Instrument dialog. You can choose a stand from a list of stands on the right. If you are not using an instrument stand, then choose the “None” option in the Stand list. If you have previously connected to the same type of instrument from the computer running the **SpatialAnalyzer** application, then you can choose to automatically run the instrument interface by checking the appropriate box on the dialog. If this is the first time that you’ve run this instrument type from this computer, then you have to add the instrument to the job, then run the interface from the **SpatialAnalyzerstart** menu. When you have completed your instrument and stand selections, hit the “Add Instrument to Job” button. Some instrument interfaces must always be run from the Start menu.

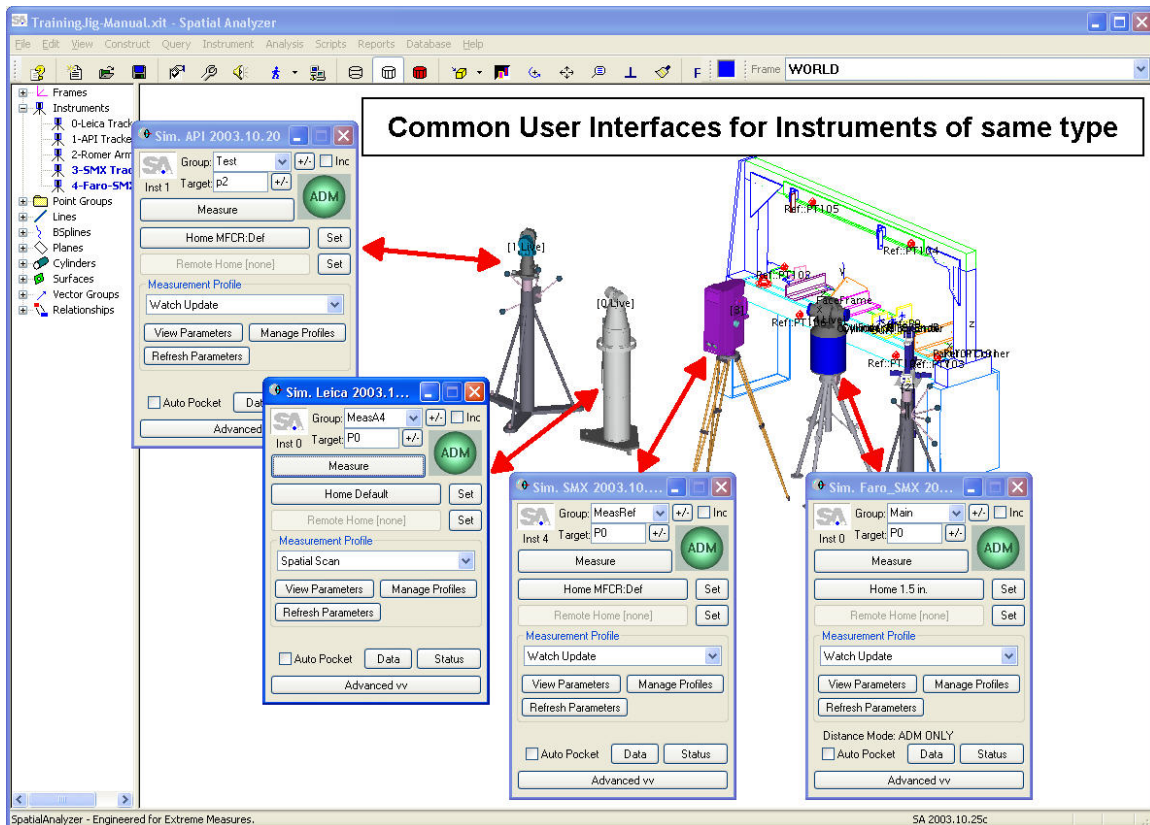
All instrument interfaces require some basic information to function properly at start-up. Note that the operator is prompted to enter their user name, the instrument ID corresponding to the index within the **SpatialAnalyzer** model (these indices will start with 0 and proceed upward as more and more instruments are added.)

The screenshot shows a dialog box titled "Spatial Analyzer Tracker Interface Logon". It has a blue header bar with the title. Below the header, there is a "User Name:" label followed by a text box containing the word "User". To the left of the text box is a small icon with the letters "SA". Below this is a section titled "Connection to Spatial Analyzer" in blue. It contains a checkbox labeled "Show All Network Components" which is unchecked. Below the checkbox is a list box showing a tree structure: "NRKWEST3 (192.168.0.2)" is the root, with a sub-item "Relationships4.xit", which in turn has a sub-item "1 Leica Tracker :3327:". The "1 Leica Tracker :3327:" item is highlighted with a blue selection box. Below the list box is another section titled "Connection to Tracker" in blue. It contains three checkboxes: "Start Tracker" (unchecked), "New Job (clear data)" (unchecked), and "Connect to Tracker, Lana #" (checked). The "Connect to Tracker, Lana #" checkbox is followed by a text box containing the number "1". Below these checkboxes is the text "Lana numbers on this PC: 19 18 14 8". Below this text are three radio buttons: "Connect from PC Lana #" (selected), "Auto Connect from PC Lana" (unselected), and a third radio button that is unselected. The "Connect from PC Lana #" radio button is followed by a text box containing the number "9". At the bottom of the dialog box, there is a "User Notes" text box, and below that are "OK" and "Cancel" buttons.

A typical instrument interface logon screen

Typically the instrument interface and **SpatialAnalyzer** will be run on the same computer. However, the TCP/IP socket connection between the instrument interface and **SpatialAnalyzer** allows the operator to run the interface on one computer and **SpatialAnalyzer** on a separate computer. While not typically used when connecting to just one instrument; when two, three, or more instruments are needed on a job simultaneously or in remote areas, the connection technology in **SpatialAnalyzer** is able to run them all dependably across your company's network. The connection technology is the same as that used to connect computers across the Internet. **SpatialAnalyzer** is architected to allow you to take advantage of your company's network.

When the operator hits OK on the instrument interface logon dialog, the selected interface will start and perform routine checks of the network and hardware. In all subsequent starts of the instrument interface, this information will be retained and the fields will be populated with this default information. The figure below shows three instrument interfaces running on top of **SpatialAnalyzer**. Menu options from within each instrument interface allow the user to control the tracker's functions, change units, and other instrument specific functions.



Instrument Interfaces Running 'On Top' of SpatialAnalyzer

Gathering Data from multiple instruments

One of the **SpatialAnalyzer**'s key strengths is its ability to gather data simultaneously from multiple instruments in the same job file. This can be done with any combination of instruments and users may connect and disconnect instrumentation at will. The process for adding a second instrument is exactly the same as adding the first. Just follow the procedure outlined in earlier sections.

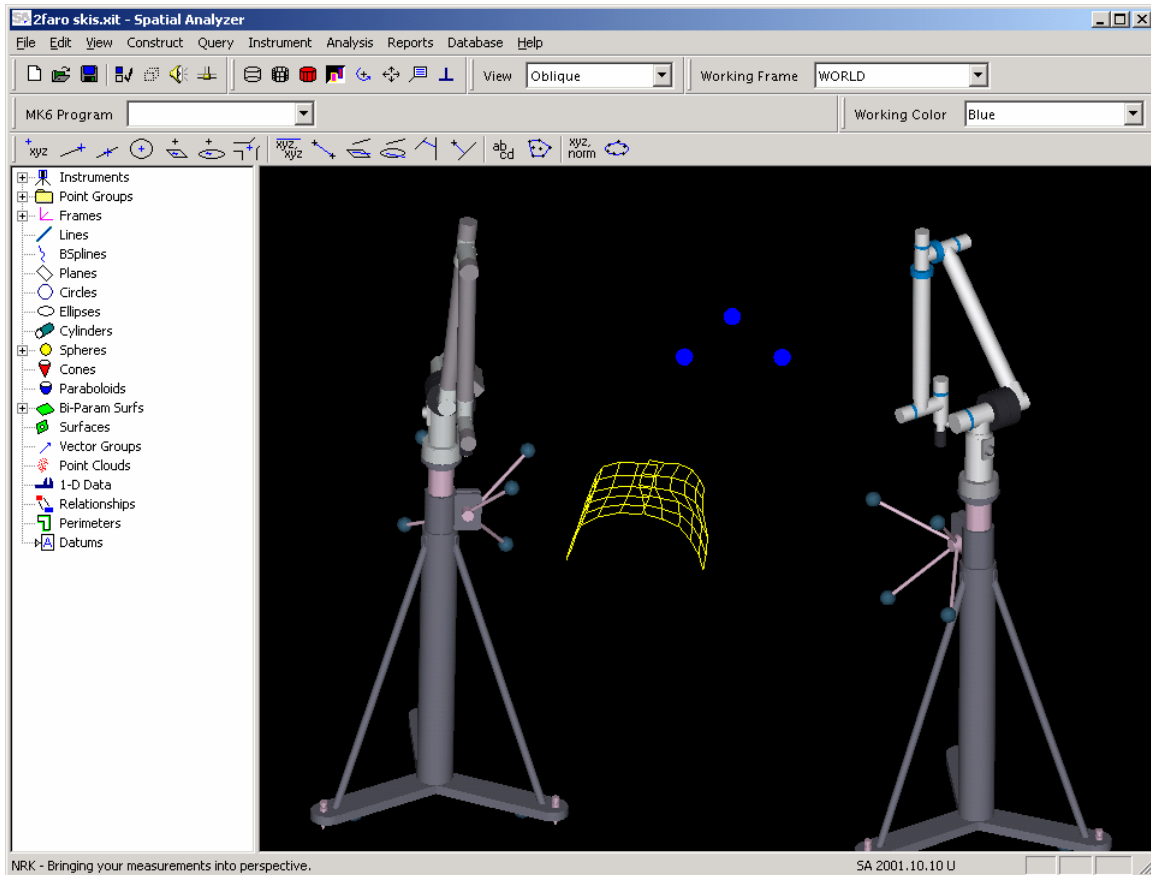
There are a few unique opportunities that arise from having simultaneous instrumentation. We will present just a few of these here.

Multiple instruments working independently

Often multiple instruments are needed to simply expand the possible extent of a job or to work around visibility obstructions. In this case each instrument gets located relative to

some reference system (like nominal design points) and they gather data that contributes to the same job. This is in many respects just like a job where an instrument is moved from one position to another during a survey except that time is saved by gathering data simultaneously and there is the potential to use two instruments of differing types.

The following figure presents a job in which two faro arms are gathering data and creating surfaces of a common part. The two instruments were located relative to each other by measuring three tooling balls and then best-fitting the resulting sphere centers. In this case one positioning of an arm could not reach the entire part.



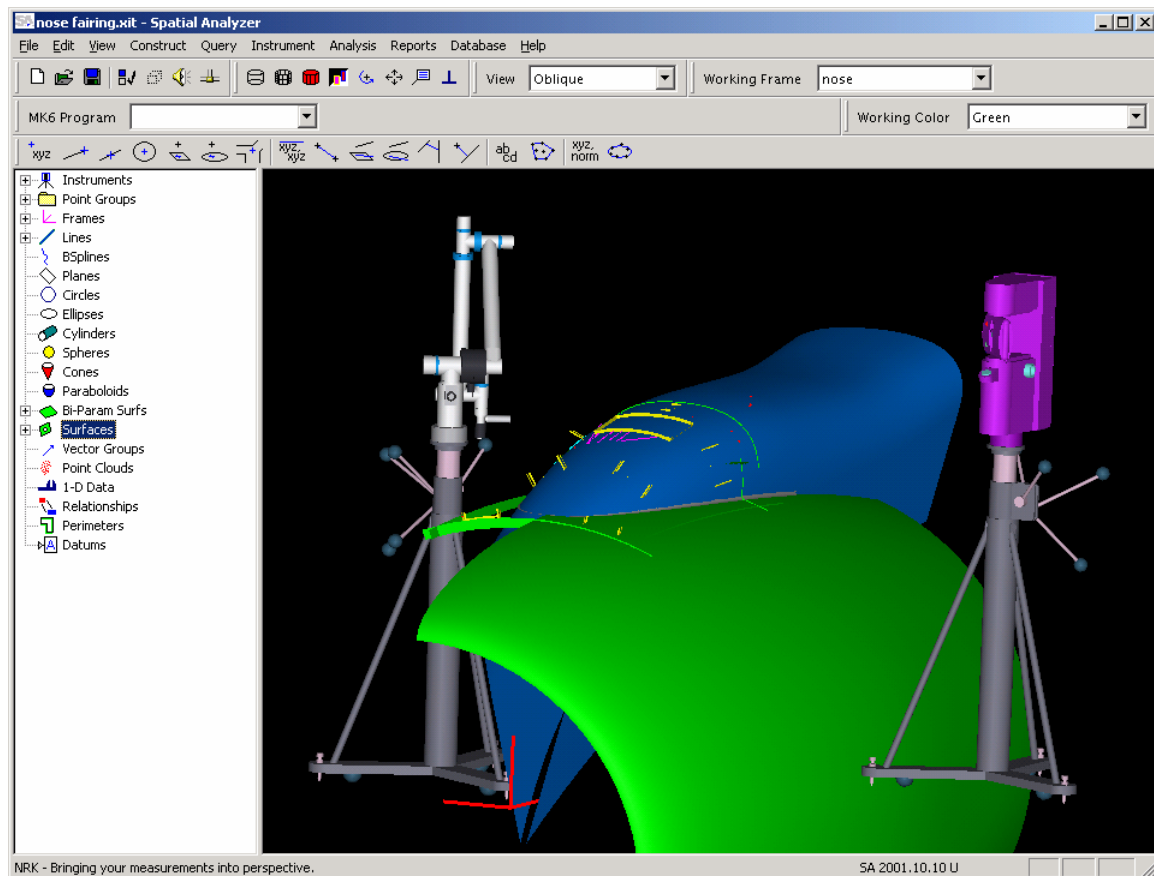
Two FARO Arms Gathering Data Simultaneously and Creating Surfaces on a Common Part

Cooperative instrumentation example (locating one instrument with another)

This is a case that truly lets us exploit the favorable attributes of each instrument to construct a measurement “system” that exceeds the usefulness of either single instrument. As an example consider the dexterity of a Faro or ROMER portable CMM for measuring holes or pockets in a complicated structure. These devices are however, limited in terms of their ability to tie into convenient references and to reach distant objects. If we combine an arm with a laser tracker, we now have in the tracker an ideal

device for performing tie-ins into a large part, and monitoring the position and orientation of the portable CMM. AT the same time we maintain the flexibility of the arm to inspect parts without the requirement of direct line-of-sight.

The following job was performed with multiple placements of a Faro arm around the part. The arm was located using the tracker which was tied into a reference system located more than 1000 inches away from this assembly. SA also contains tools for locating the arm by replacing its touch probe with an SMR. The SMR can then be measured by the tracker in several different positions and the arm automatically located relative to the tracker. This operation uses synchronized measurements from each device to ensure the greatest accuracy of the resulting positioning.



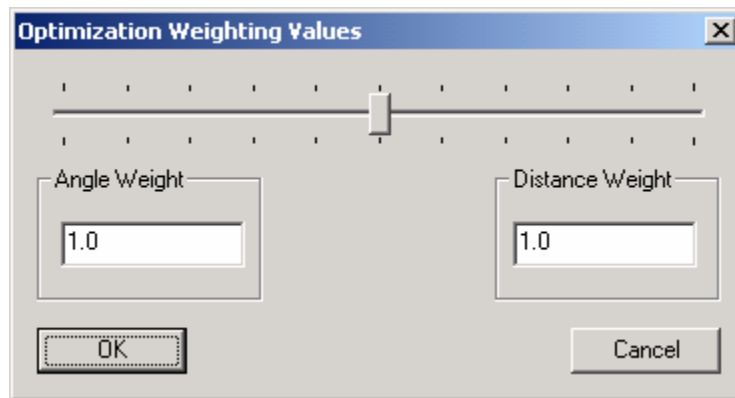
FARO Arm orients to the part with SMX tracker measurements

Cooperative instrumentation example (real-time monitoring position and orientation of a part)

Another use for multiple instruments is to immediately gather more data than a single instrument is capable of. For example, to determine the position and orientation of a part with a tracker requires that the operator measure at least three targets. This can be done instantaneously if three trackers are available. Further this position and orientation information could also be used to drive other devices like robots or cranes to control the relative position and orientation of two objects.

Cooperative instrumentation example (multi-lateration for extreme precision)

In cases where extreme precision is required, it is possible to use three or more laser tracker to monitor a single SMR. In this scenario, the trackers effectively form a network much like a typical theodolite network. When bundling theodolites (total stations) it is common practice to disregard the distance measurements since the angle encoders provide greater accuracy. In SA's bundle, users may vary the relative importance of angular or distance data accordingly. For an EDM theodolite bundle you would discount distance and favor angular data. For a tracker, you would discount angular data and weight heavily the distance data since the interferometer is much more accurate than the angular encoders. These relative weights are adjusted in the following dialog located within the units tab of the user options.



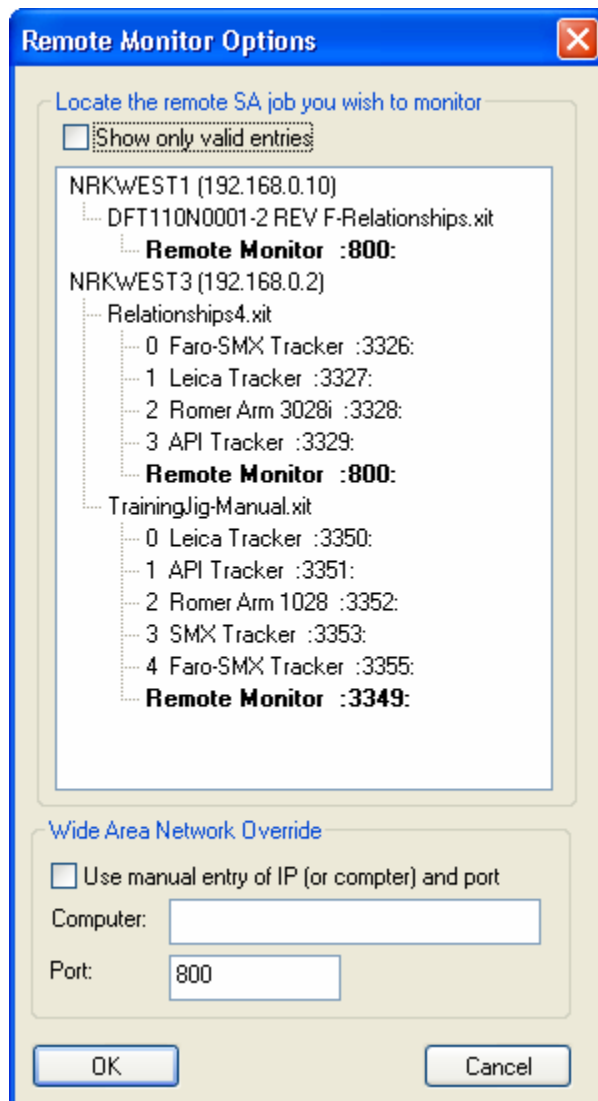
Relative Angle vs. Distance Weighting Dialog

For bundled trackers, when only distance values are used measurements can be taken with 5 micron (0.000,005 inch) one sigma uncertainty. The need for this kind of accuracy does not occur often, but it is impossible to obtain this currently without multiple instruments.

Remotely Monitoring a Measurement Session with another copy of SpatialAnalyzer

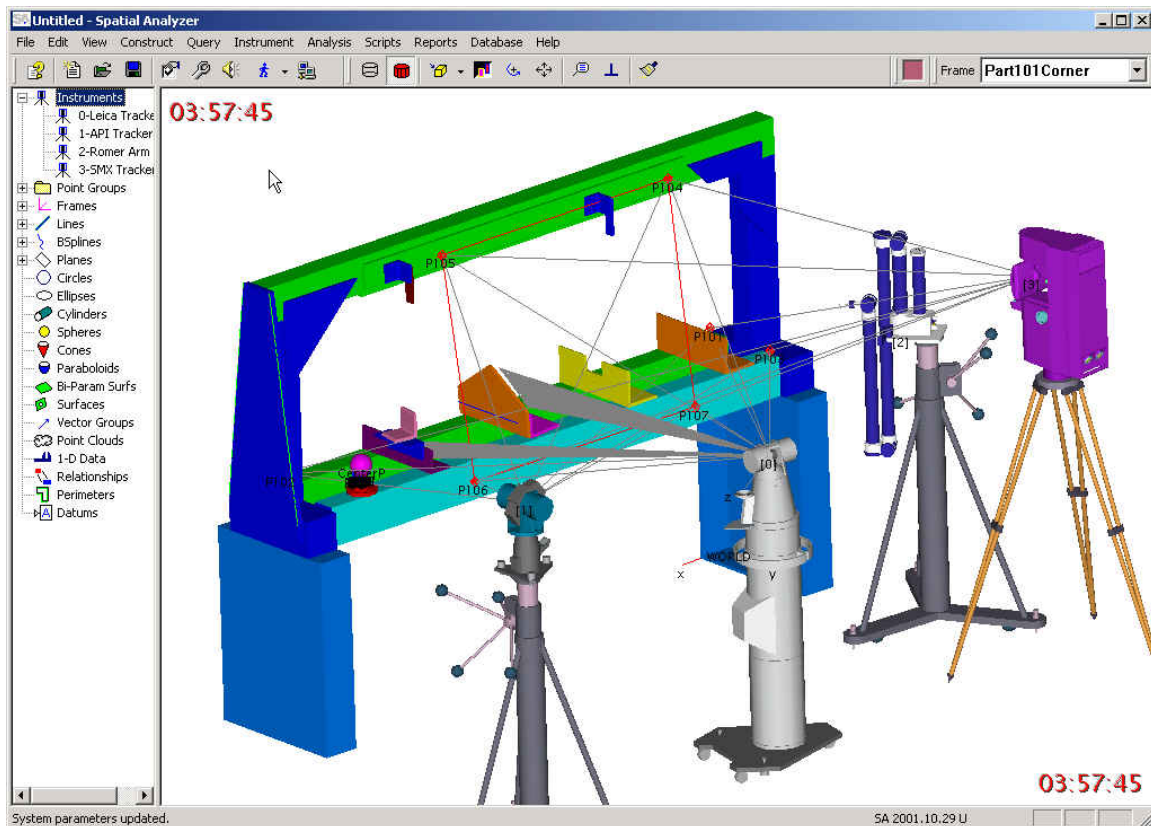
One unique feature of **SpatialAnalyzer** is the ability to remotely monitor a measurement job in process. The general nature of **SpatialAnalyzer** architecture enables this unique feature. The remote copy of **SpatialAnalyzer** is treated simply as if it were yet another instrument type gathering coordinate and geometric data.

To remotely monitor another copy of **SpatialAnalyzer**, open a new file with the File, New menu option. From the File menu, select Remote Monitoring another **SpatialAnalyzer** menu option. This will pop up the remote monitoring configuration dialog.



Remote Monitoring Configuration Dialog

From the network tree view select the **SpatialAnalyzer** job that you want to remotely monitor. Hit OK to begin monitoring. In nearly all respects, the monitoring **SpatialAnalyzer** is identical to the original. The one exception is that the corners of the graphical view will bear a time stamp of when the monitoring was last updated.



Remote Monitoring with SpatialAnalyzer

Users should note that the monitoring job is in fact a true copy of the original. You may perform analyses and construct geometries in this copy. This job may be saved just like any other **SpatialAnalyzer** job.

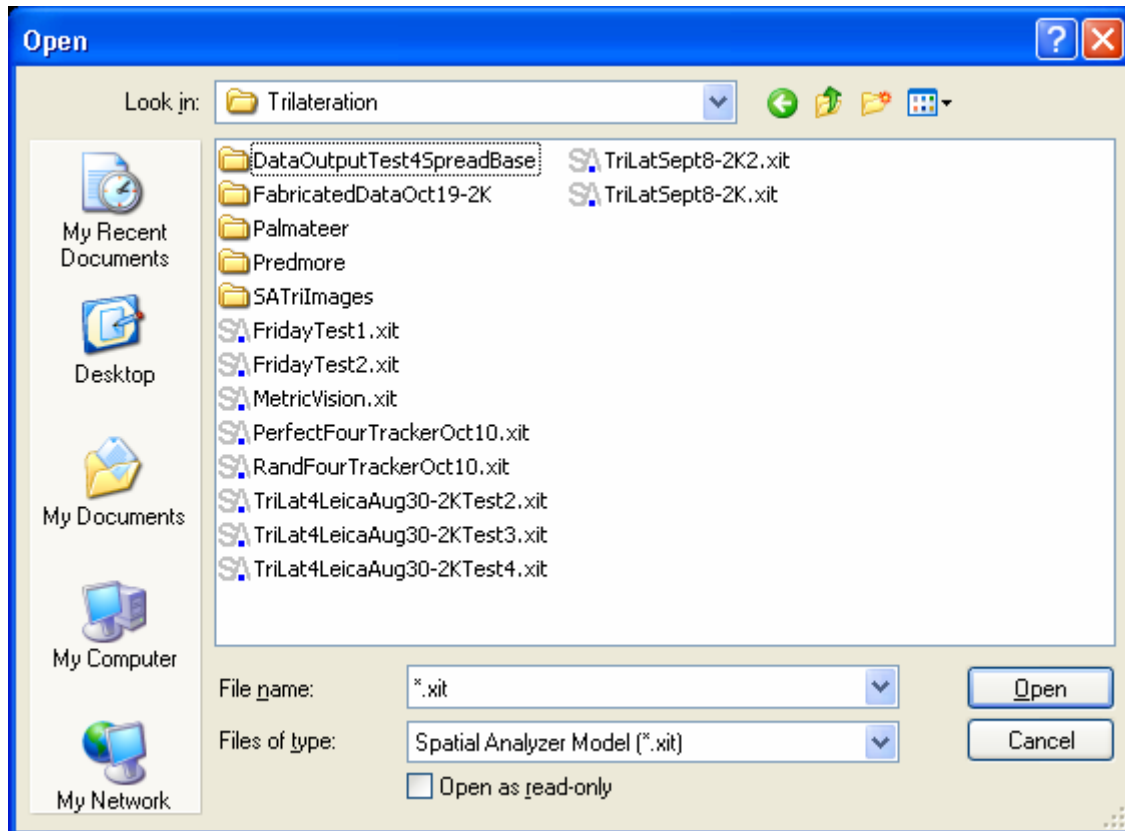
Survey templates

Since there are a wide array of settings and customizations a user may configure within the SpatialAnalyzer, template files are necessary to avoid repeating the same option selections each time a new job is created. To save a template file simply select the File->Save as read only template. This will write the file and all it's setting to disk to be opened later as a template file. Template files are essentially standard **SpatialAnalyzer** files except that when they are opened the job name is automatically modified to be **SpatialAnalyzer** job<today's date and time>.xit. This is done so operators can easily open templates and not have to worry about overwriting or corrupting the templates with data.

You can set the default job file by **SpatialAnalyzer** opens to a particular template file by naming it "default.xit." Put the "default.xit" file in the [Analyzer Data\Templates] directory. SA will automatically look for this file and load it every time a new file is loaded. If it does not exist, SA loads its default job each time a new job is requested.

Import/Export Modules

In addition to receiving measurement data from instrument interfaces, **SpatialAnalyzer** is also capable of exchanging data with any of several standard file types. The most common of these file exchange types will be discussed here. These functions are accessible on the File, Import and File, Export menu selections. All of these file import/export functions utilize the familiar Windows file selection dialog.



Windows File Selection Dialog

Refer to the Menu Reference chapter for information regarding the particular options for any one type of import or export. One such file import operation that warrants a discussion here is the **SpatialAnalyzer** .XIT file format itself. These files are in a compressed binary format that is used by **SpatialAnalyzer** for efficient storage. This enables a user to effectively merge or combine data from multiple measurement sessions. After combining the data the user would simply treat the data as if it all were gathered at the same time with different instrument stations. To orient one instrument relative to another, the user will have to either best-fit coordinate data or bundle the instrument observations.

Importing Data

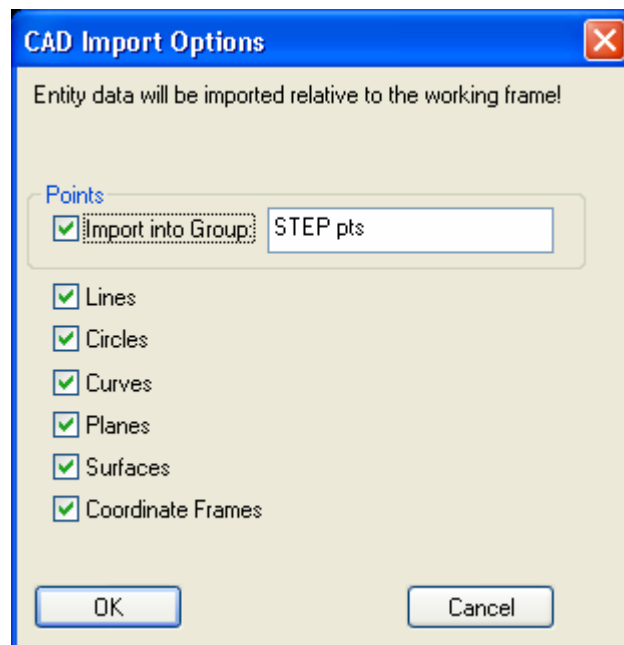
SA is capable of importing points, geometry, and process information from many types of data files. This section will provide examples of this process and explain the various options and conventions. The first type of import is the CAD model import. This is used

to get geometry from a design CAD system into **SpatialAnalyzer**. This is typically done so that measurement data can be quickly compared to the design CAD model.

There are many file formats CAD systems use to transfer information. These are typically standard formats that all other CAD systems can read. **SpatialAnalyzer** can read and write STEP, IGES, and VDA. STEP is the most modern standard and gives the best results when using CAD systems such as CATIA.

Importing CAD Data

To import a CAD file (IGES or STEP) into **SpatialAnalyzer**, go to File->Import then pick the format you are interested in. You will next be prompted to pick the file. Once that is done, the following dialog will appear:

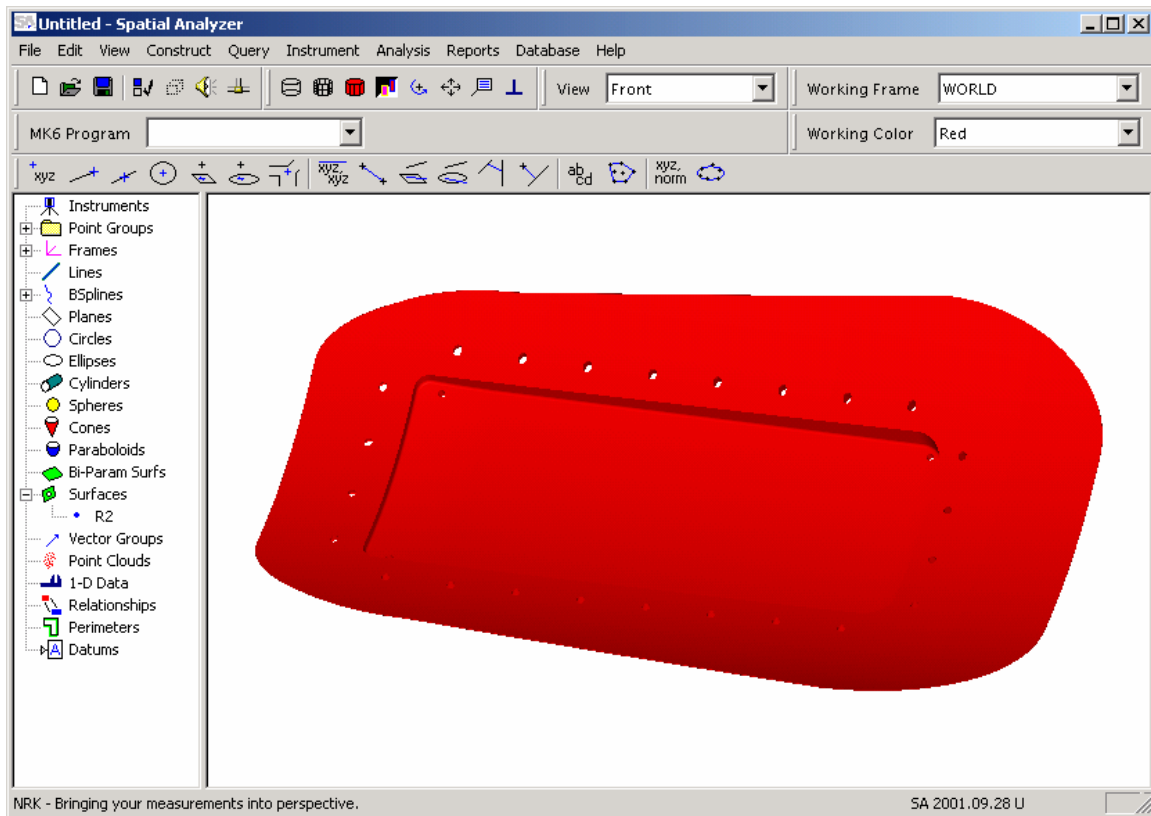


CAD Import Options Dialog

This dialog allows you to filter out certain types of objects from the import. This is useful since many CAD programs export a multitude of extraneous objects that were used in the construction of the geometry. Generally speaking you will want the surfaces from the CAD file and possibly the points.

Notice that you are prompted for the group to put the points in. This is necessary since CAD files do not group points the way **SpatialAnalyzer** does. All the points in the CAD model will go into the **SpatialAnalyzer** group name you specify.

When you hit OK in this dialog, **SpatialAnalyzer** will import the CAD file. This may take some time depending on the complexity of the model. In this case, after about 20 seconds, the model appeared in **SpatialAnalyzer**:

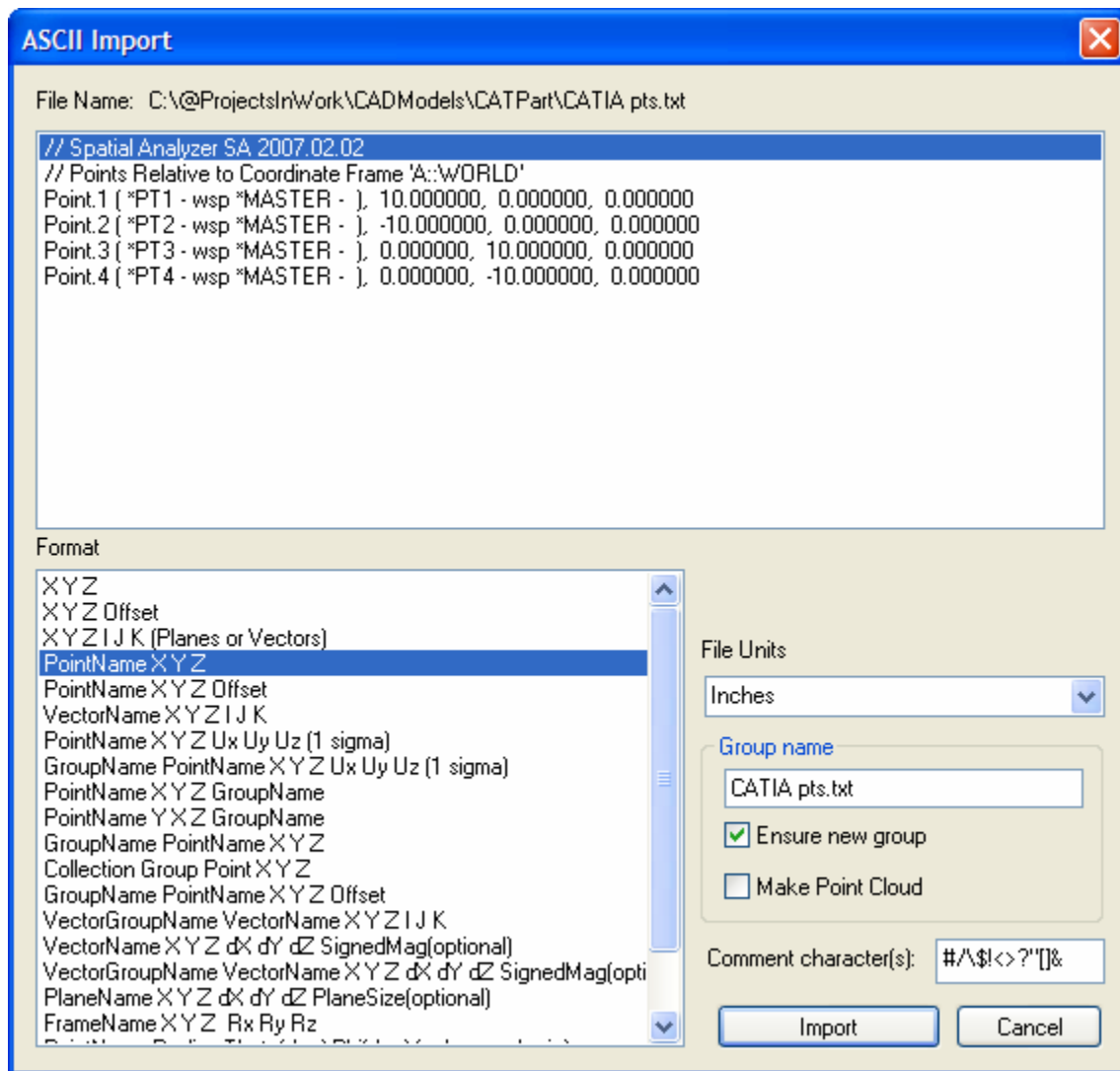


SpatialAnalyzer's Graphics View of the imported CAD file

Importing ASCII Data

ASCII text files are common in coordinate metrology. They provide a standard, human-readable format that many software packages can easily work with. SpatialAnalyzer supports many, many formats of ASCII data. These are imported using the File->Import menu and selecting the "ASCII: Predefined Formats" option.

As with the CAD import, you are first prompted to select the file you wish to import. Once you do that, you will see a preview of the file and several format options:



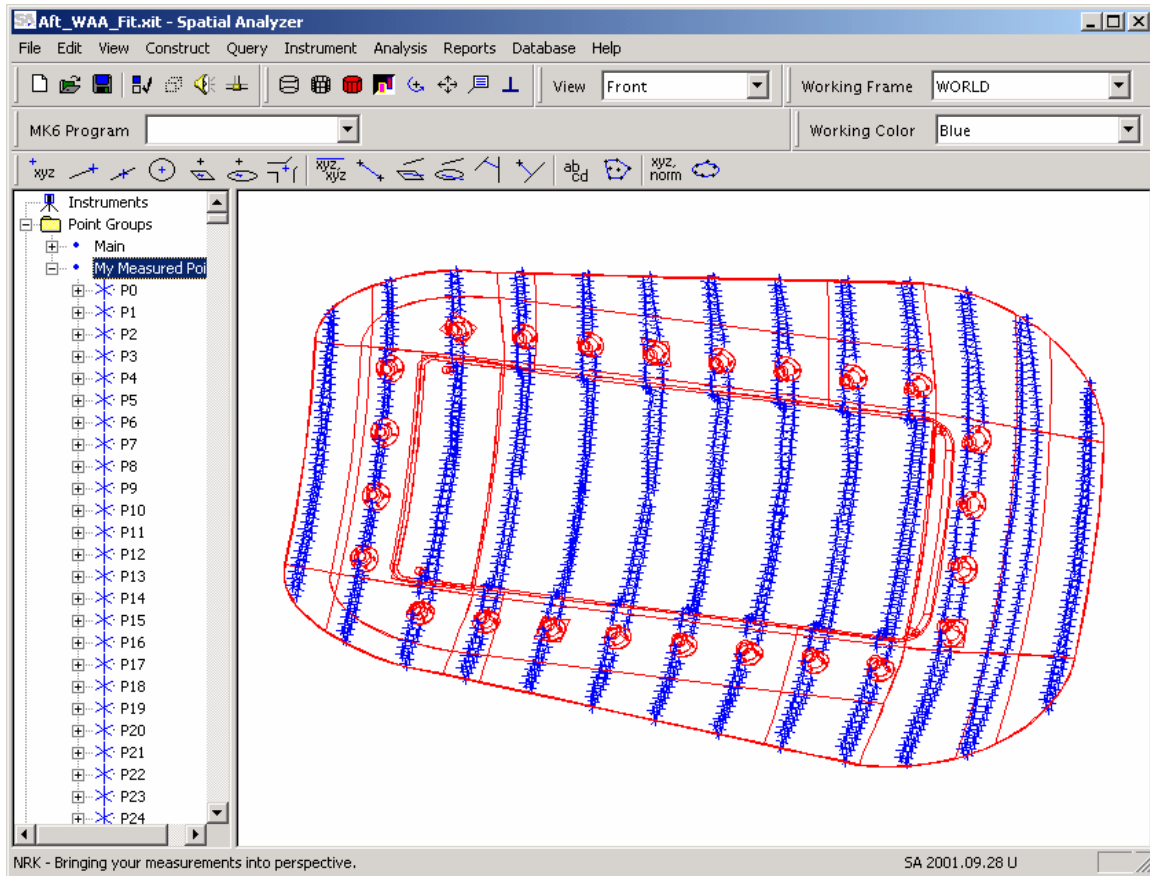
ASCII Import Dialog

In the preview window, highlight the line where the import should begin. This allows you to easily skip over any comments at the top. Also note the list of comment characters at the bottom of the screen. Any line that begins with one of these characters will be skipped.

Next, select the file format. In this case, we want “PointName X Y Z”. Then, select the units of the data in the file. If these are different than the units of your **SpatialAnalyzer** job, **SpatialAnalyzer** will automatically convert the points during import.

You can also pick a name for the point group that will contain the points. This field is ignored if you pick an import format that includes group name. Along with the group name are options to force the creation of a new group and create a point cloud.

When you are finished selecting the options, press Import and your points will be added to the **SpatialAnalyzer** job. Here is an example of what you would see:

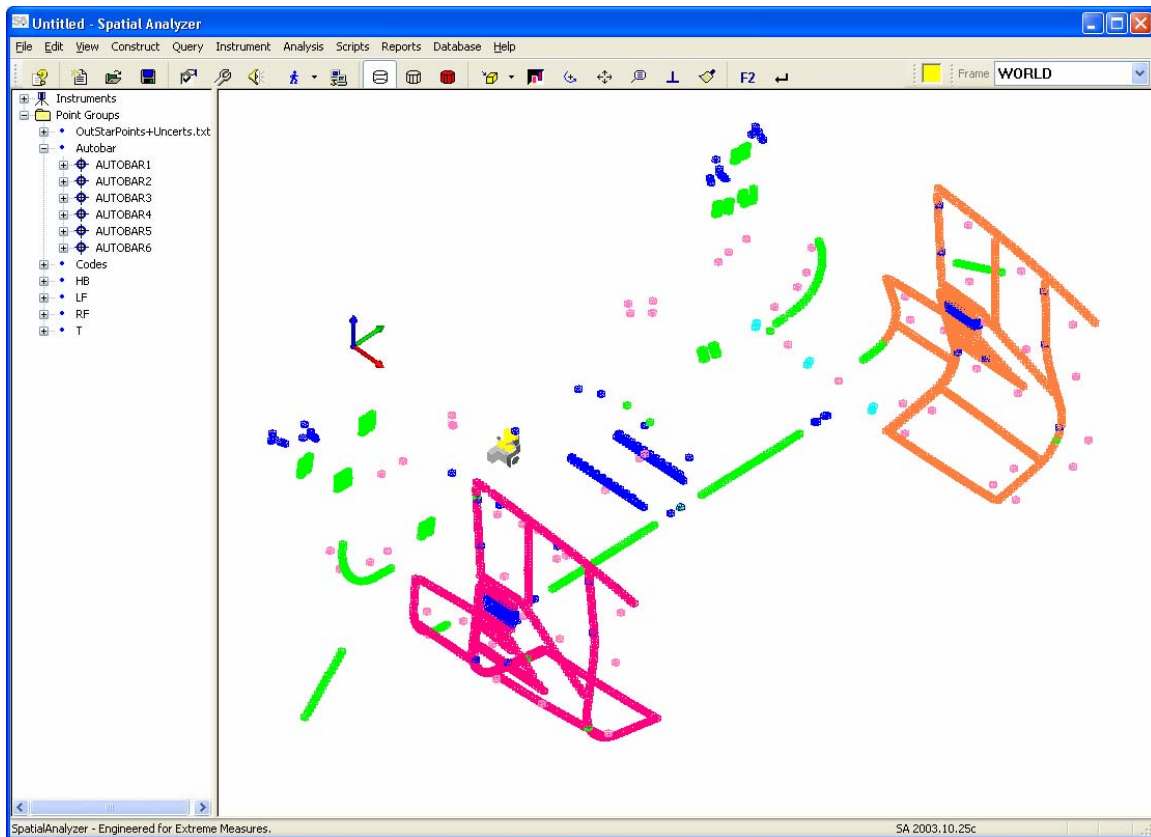


SpatialAnalyzer's Graphics View of the imported CAD and ASCII files

Notice the new point group on the tree-view. It contains the points that were in the ASCII file.

Importing V-Stars Data

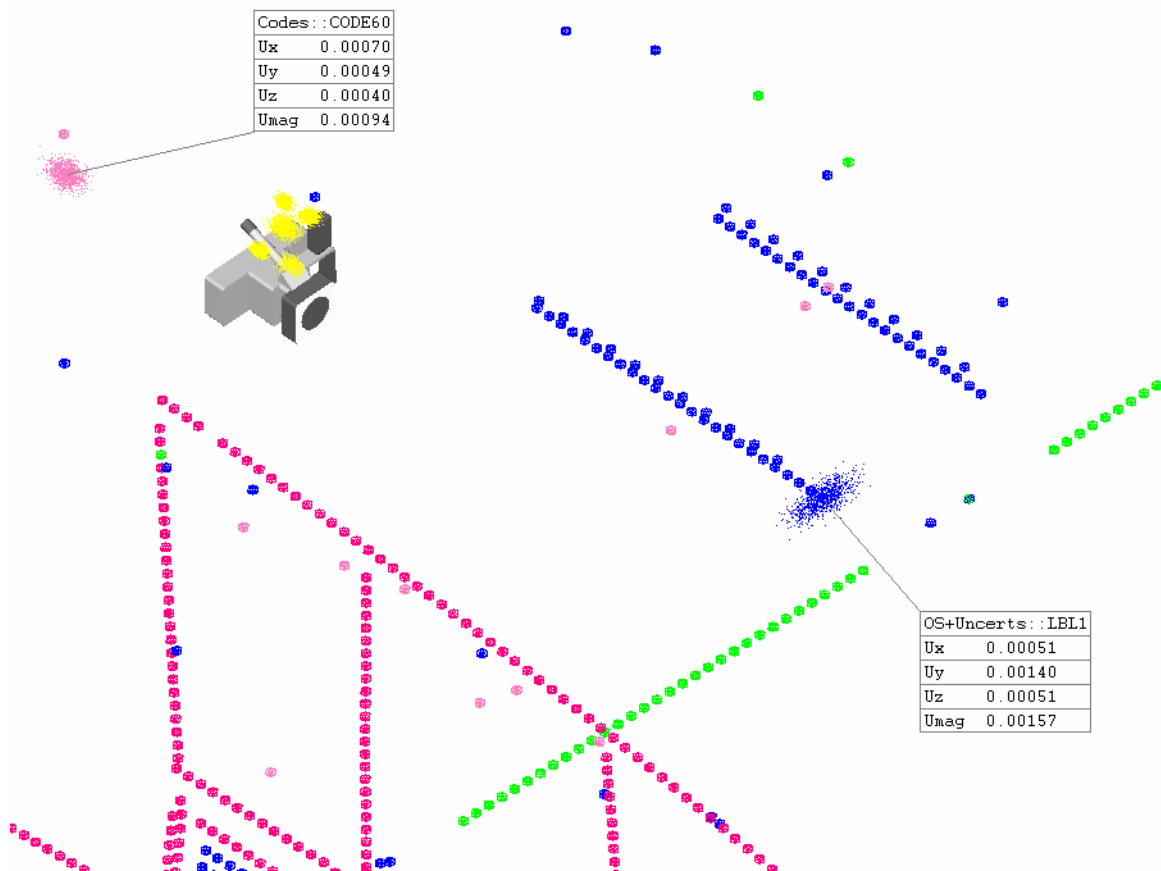
The VSTARS .xyz format is directly supported in SpatialAnalyzer. You can use this format by selecting File->Import->Custom Formats->VSTARS .xyz. Next, you will be prompted to select the file you wish to import. Once the selection is made, the file will be read and a message box will tell you how many points were imported. In this case, the file, Bundle15.xyz was selected. After the import, the 880 points appear in the **SpatialAnalyzer** model:



Graphics view of V-Stars points

Notice that the points are placed in the “Codes” group. Also notice that the point names from the VSTARS file are preserved.

You can also import a VSTARS file using the standard ASCII import utility presented in the previous section. This utility allows you to select a predefined format. In this case, use “PointName X Y Z UX UY UZ” to import the points and their respective limiting accuracies from the OutStar.txt file. With the limiting accuracies in SA you can create uncertainty fields and visualize their respective sizes. The figure below show several uncertainty clouds.



Graphics view of V-Stars with Uncertainties

The result will be the same, but you will be allowed to select the group name for the points as opposed to having it automatically selected by the filename.

Exporting Data

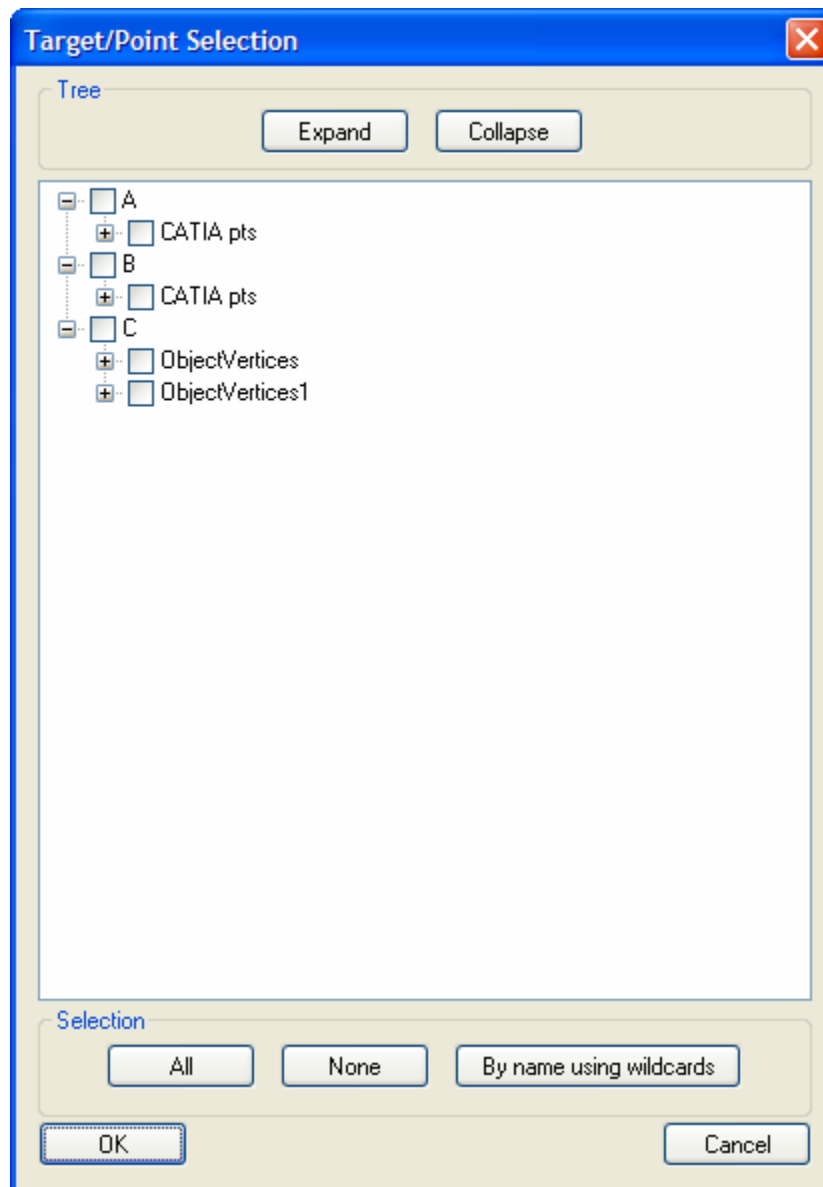
Exporting ASCII Data

To export an ASCII file, go to File->Export->Points (ASCII File). You will be prompted to select the points you wish to export. There are several ways to pick the points:

Hold down SHIFT and the left mouse button. This will allow you to drag a window over the points you wish to select.

Double-click the points individually.

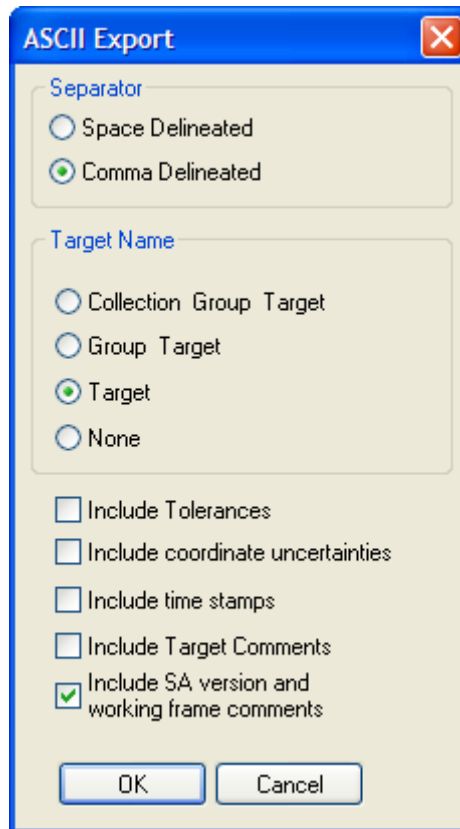
Press F2. You will be prompted with a point selection dialog:



Selection dialog

Using this dialog, you can select entire groups or individual points. Notice that the number of points in each group is listed beside the group name. In this case, we selected 7 points, 3 from one group and 4 from another.

After using one or a combination of the above selection methods, you will see the selected points highlighted on the graphical view. You will also have a cursor prompt letting you know that you need to press ENTER to end the selection. Once you press ENTER, you will be asked to enter a filename for the export. Next, the point export dialog will appear:



ASCII Export format selection dialog

There are several options that control the formatting (and the information) of the output file. In this case, we selected points from 2 different groups so it makes sense to include the group names in the export. After pressing OK, the file will be written. The result is:

```
// Spatial Analyzer SA 2001.10.08 U
// Points Relative to Coordinate Frame 'WORLD'
measured::r1   -5.126804    2.546770    -5.206153
measured::r2   -7.483444    5.164647    -7.856990
measured::r8    6.147343    7.618336    -1.481063
Bulkhead 4::r1   9.866939    3.274941    -6.859645
Bulkhead 4::r3  -0.719932    -6.633198    3.630177
Bulkhead 4::r4   1.864986   -9.689932   -5.153050
Bulkhead 4::r5   3.642384    3.975646    -8.991668
```

Exporting Graphic Images

SpatialAnalyzer provides several methods for exporting the graphical views. This is useful for conveying measurement information in a report or other document. Often, the views that are captured contain whisker plots showing the discrepancies between the CAD design for a part and the measured “as-built” object.

Still Images

One basic method that is built into Windows 2000 operating system is the Print-Screen key. You can use SHIFT + Print Screen to capture the entire screen, or ALT + Print

Screen to capture the active window. This is an easy way to capture images. To the clipboard so you can then paste them into Word or another application.

There is also an option on the **SpatialAnalyzer** menu: Edit->Copy Graphics to Clipboard. This places the graphical window (excluding the **SpatialAnalyzer** frame and menus) on the clipboard.

SA Quick Reports put the current graphics image on each report. These reports are automatically saved on the SA treeview. Right-click on the object report in the treeview to view the report. Printing or other output options are available on the report dialog.

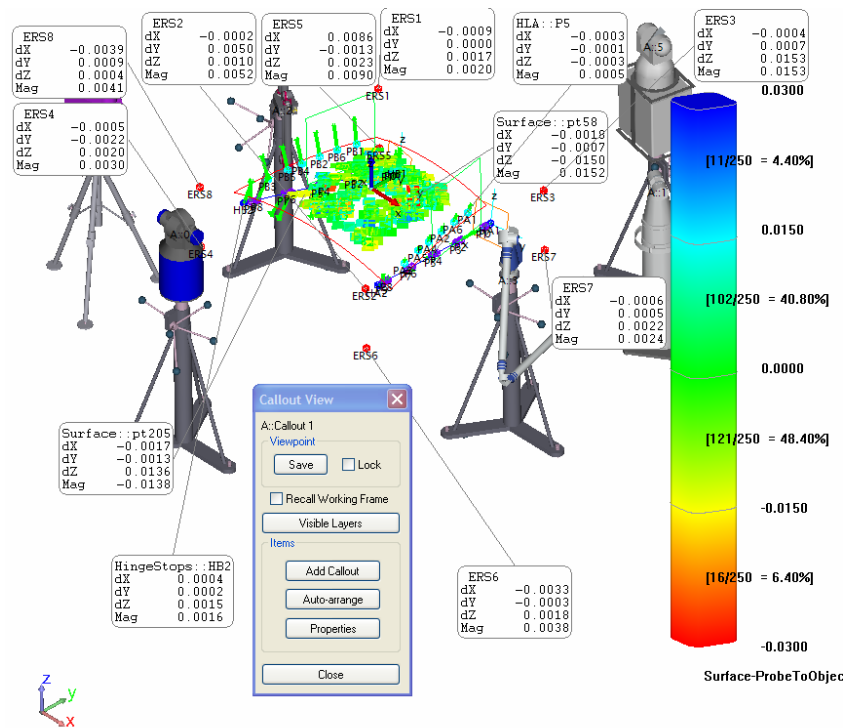
Another method for exporting the graphics is to use the MS Office XP Reporting integration in **SpatialAnalyzer**. This is an advanced feature, but it is simple to use. From the Reports menu, select MS Office XP Automation. Then, initialize a Word report. Once this is done, you can selectively dump things directly from **SpatialAnalyzer** into MS Word. One of the options is "Add Graphics Window". This will capture the graphics in **SpatialAnalyzer** and add it to the Word report. After adding to the report, select the Save and Close option to close out the file. Then, you can open the Word report to see the graphical views and other information. Also, note that these capabilities are available through the SA Measurement Plan. This will be presented in more detail in later sections. It will allow you to automate the process of changing views, and exporting the graphics to Word.

Callout Views

Often, you want to annotate the graphical views created in SA. To do this, use the Callout-View functionality. To do this, press the Callout-View icon on the SA toolbar:

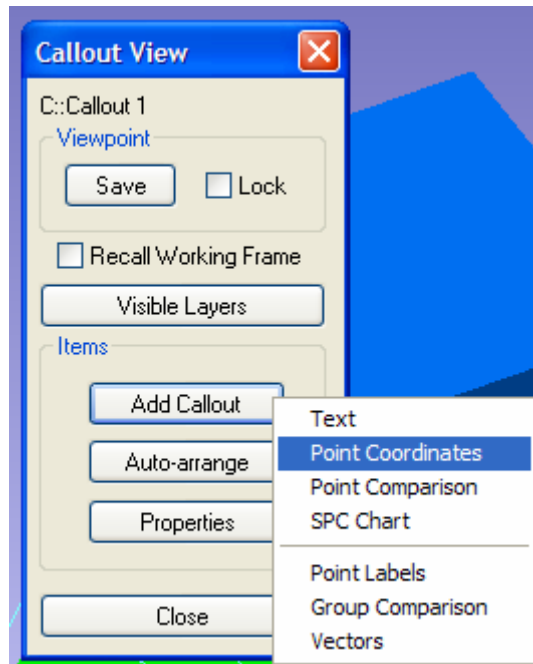


. This will prompt you to name the new callout view:




Callout view naming dialog

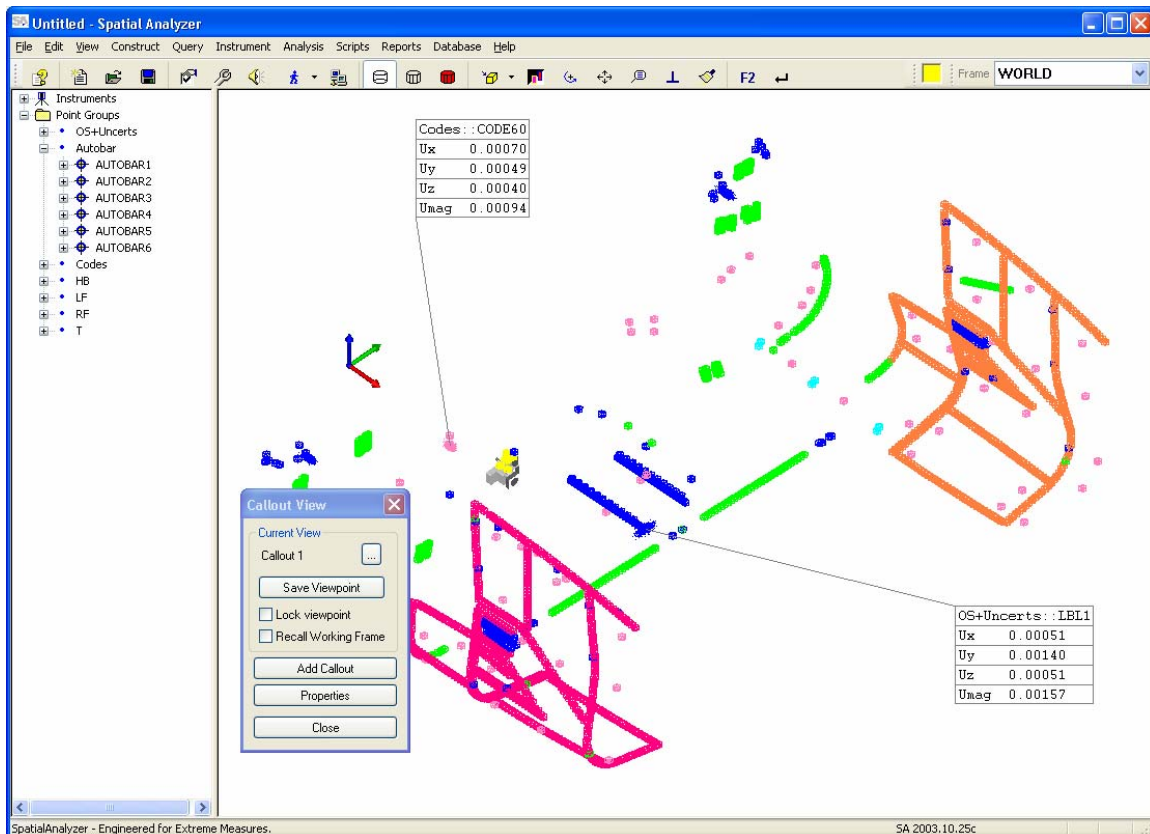
After selecting a name, the callout dialog will appear:



Callout view selection dialog

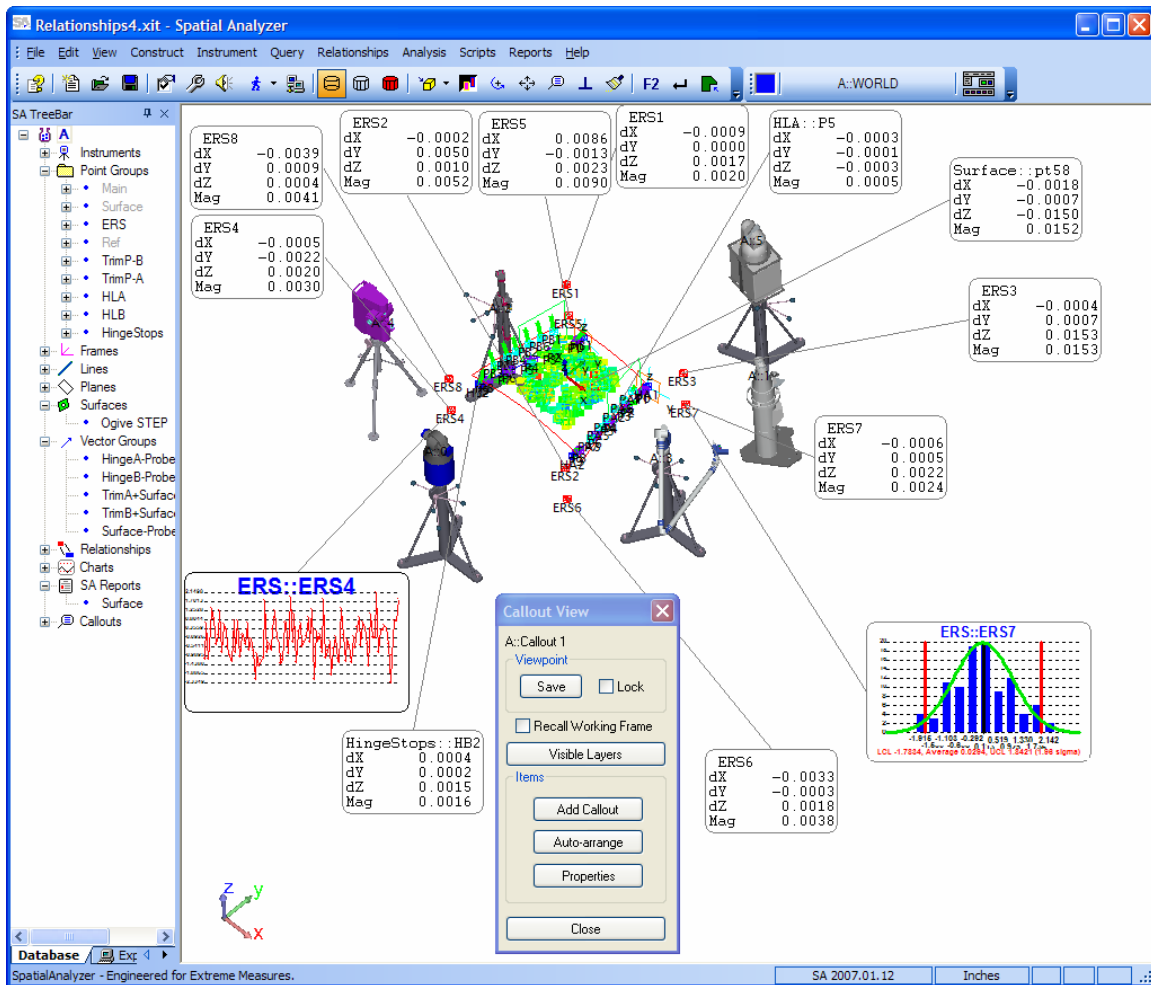
The callout views are saved on the SA treeview. Right-click on the callout view to show, rename, or delete it. This is convenient for providing a “slide show” of a job with various portions highlighted. Not only does each callout view save the callout objects, it also can save the view rotation and zoom setting. Whatever view you leave the callout in when you close the callout dialog is the view it will come back in when you revisit the callout.

To add callouts to a model, select . This will give you a popup menu with several callout choices. In this case, we'll choose “point coordinates”. It will then prompt us to select a point in the **SpatialAnalyzer** graphical view. After that, it will prompt for a callout location. Then, the callout will appear on the graphical view:



Callout view in the graphics window

There are other types of callouts that may be added. These include text notes, point comparisons, SPC charts, point labels, Group Comparisons, and vectors. Examples of these are shown below:



Callout views with an example of an histogram

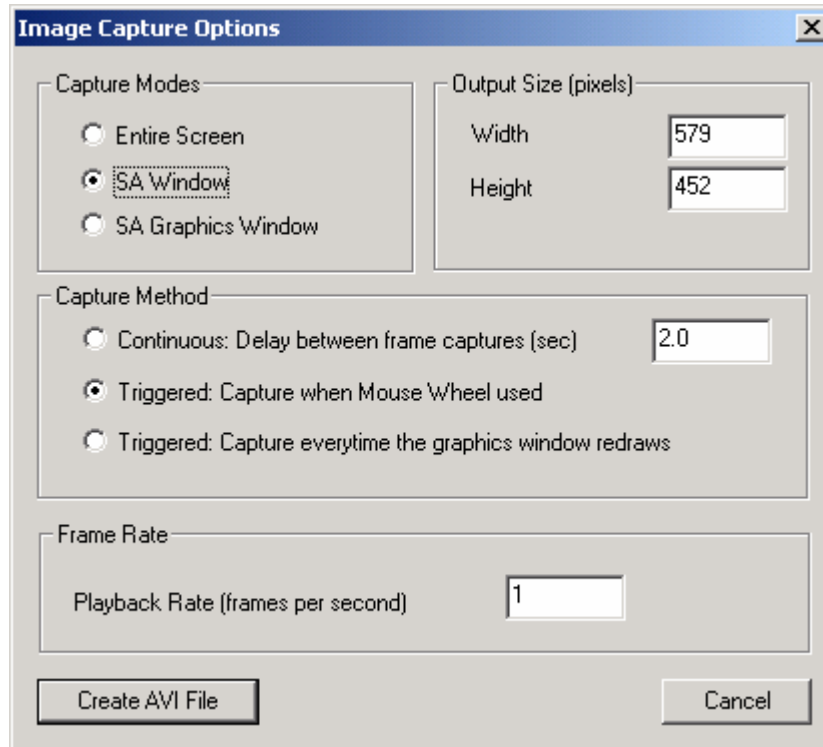
When you move your mouse over a callout, you can left-click (with the mouse button) on it to move it on the screen. If you right click, you can edit the callout properties, delete it, etc.

Once you have annotated your view with callouts, you can use any of the standard methods to capture the image and place it in a report or other document.

AVI Animations

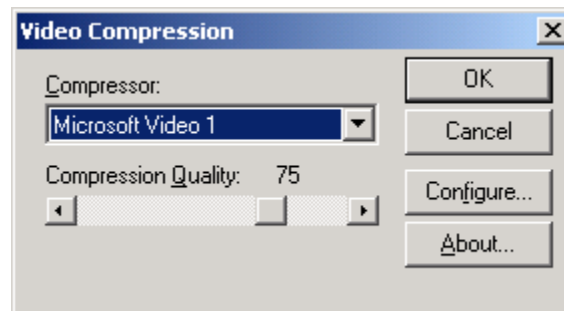
SpatialAnalyzer also allows you to create an AVI movie of either the graphical window or the entire **SpatialAnalyzer** screen. These are useful for conveying a process or visualizing various views of a measurement job.

To create an AVI movie, go to View->Create AVI Movie. There are several options under that menu. First, select New File. After picking the filename, you will see a dialog with several options:



AVI Image Capture Options Dialog

This lets you control various options for the capture and playback. After setting these options, hit the Create AVI button. You will then see a dialog asking you to choose a video compression algorithm:



AVI Video Compression Dialog for the Microsoft Video 1 CODEC

We have had the best results with Microsoft Video 1. The main reason is that it is installed by default on most Windows machines.

Then, capture frames using the trigger you selected. When you are done, select Close File from the AVI menu. To view the AVI, just open the file you created and it will auto-play.

USER INTERFACE CUSTOMIZATION AND PROFILES

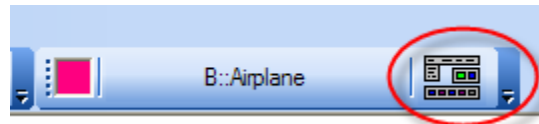
This feature allows users to create their own customized user interfaces. While SA is in customize mode, users can create new toolbars and/or menus which contain the commands they use most often. They can also remove toolbar buttons and/or menu items which they do not need. It is also possible to re-arrange the existing menus and toolbars to better suit the user's needs.

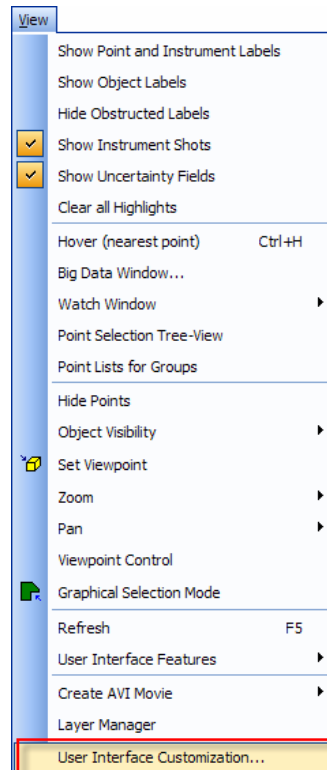
User interface customizations can be saved as profiles. Once saved, user interface profiles can later be recalled. The currently selected profile becomes the default used when SA starts. It is also possible to switch to the special "Factory Default Settings" profile which strips all customization and restores the original, unmodified user interface.

User interface profiles are stored as files with a ".saprofile" extension in the "Analyzer Data\Templates" directory. This allows easily sending a profile to another user. When receiving an ".saprofile" file, the user can simply copy it into their "Analyzer Data\Templates" directory and it will then be available.

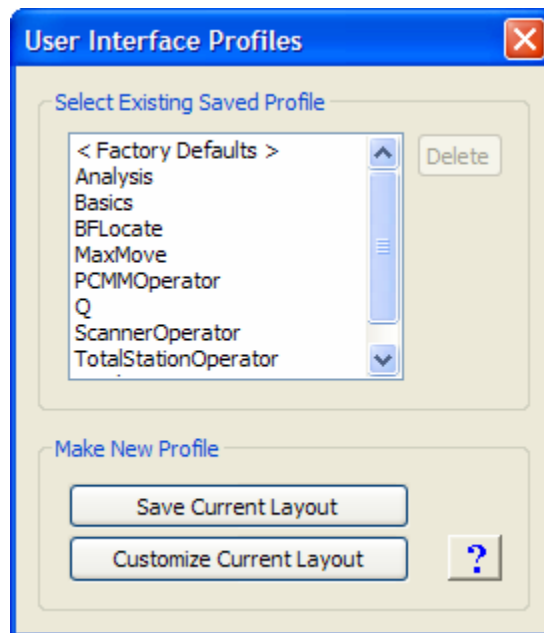
Customizing the Interface

To begin customizing the interface, use the toolbar button or menu item shown below.

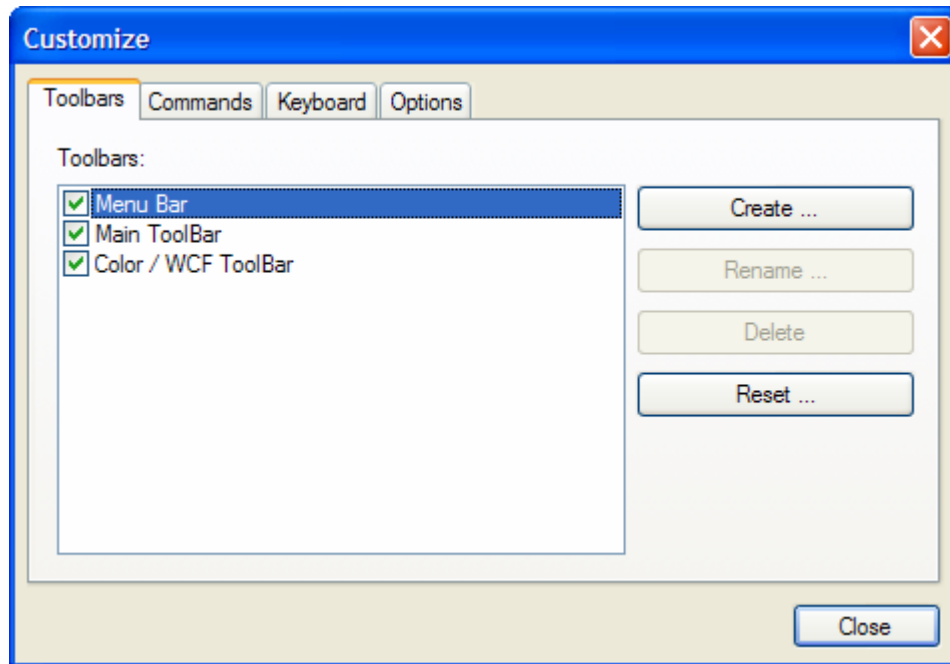




This will open the “User Interface Profiles” window shown below.



From this window, click on the “Customize Current Layout” button. This will open the user interface customization window shown below.



As long as the above “Customize” window is open, SA is in customize mode. While in customize mode, you can not interact with the menus or toolbars as you normally would. Instead, you can drag toolbar buttons around, drag items off of the menus, drag items between menus and toolbars, etc.

When finished customizing the user interface, close the “Customize” window. In the “User Interface Profiles” window shown earlier, you can now save your customizations as a new profile using the “Save Current Layout” button. This prompts you for a name, which will then be added to the list of available profiles.

Recalling a Saved Profile

With the “User Interface Profiles” window open, select a profile from the list (or select the special “Factory Defaults” profile to strip all customization and restore the stock user interface) to load that profile and make it active. Once you’ve selected a profile it will become the default used when SA starts.

Removing Profiles

To remove a profile which is no longer wanted, select the profile from the “User Interface Profiles” window and click the Delete button. This will delete the corresponding .saprofile file from the “Analyzer Data\Templates” directory.

GROUPS, POINTS AND TARGETS

The attributes and properties of Groups, Points and Targets form the basis of the **SpatialAnalyzer**' geometric database. **SpatialAnalyzer** treats measured data, or , differently than constructed data, or . A is defined as a point that is determined directly from measurements. are similar to targets but are either created by the user, fit to data, or generated by moving targets. This architecture is enforced to prevent the user from misinterpreting information.

For convenience, both targets and points are stored in groups. These groups may contain any number of points and/or targets. The purpose of grouping is to allow the user to organize data. It is important to note, however, that users who are not interested in using the grouping functionality may simply place all points and targets into a single group.

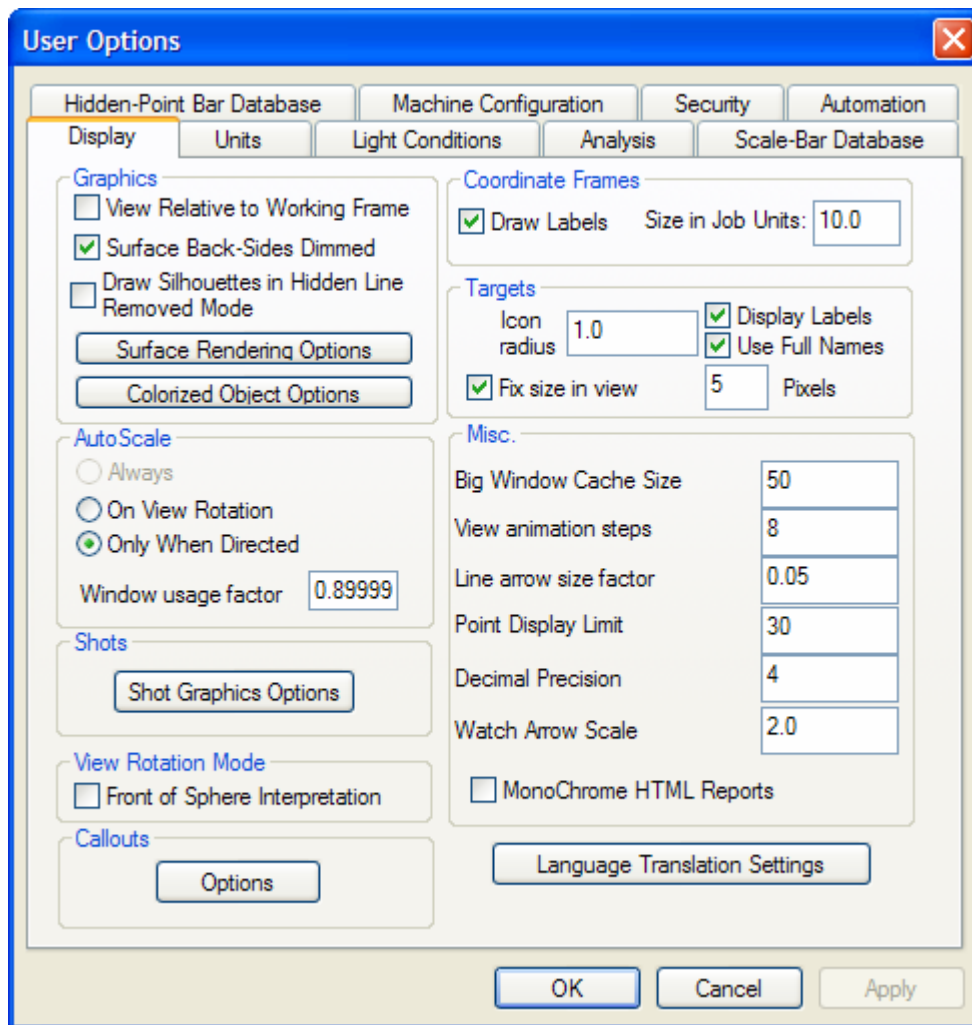
Notational Conventions

Since Points and Targets in **SpatialAnalyzer** have both a Group name and a Point name, it is necessary to specify both names to uniquely identify the point. When the feature is enabled in the graphics window, all targets and points will be labeled with the target and possibly the group name, depending on the setting on the display page. The format is as follows:

<group name>::<target name>

So, if we had a group called "Lower Jig Fixture", and a target named "Ball 3", the name would appear as "Lower Jig Fixture::Ball 3".

From the Display tab of the User Options dialog it is possible to disable "Use Full Names" check box. It is in the Targets section on the Display tab. With the "Use Full Names" option off, only the target name shows in the graphical view.



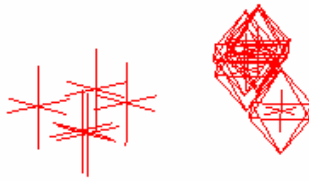
Display Tab on the User Options Dialog

Regardless of whether full names are displayed or not in the graphical view, all measurement reports and other forms of output each point and target will be uniquely identified.

Graphical Distinction between Points and Targets

Points and Targets are depicted differently in the SpatialAnalyzer graphical environment. A point is drawn as a 3-dimensional plus symbol. Targets are drawn as points with a 3-D diamond structure surrounding them.

The figure below depicts both representations.



Points and Targets

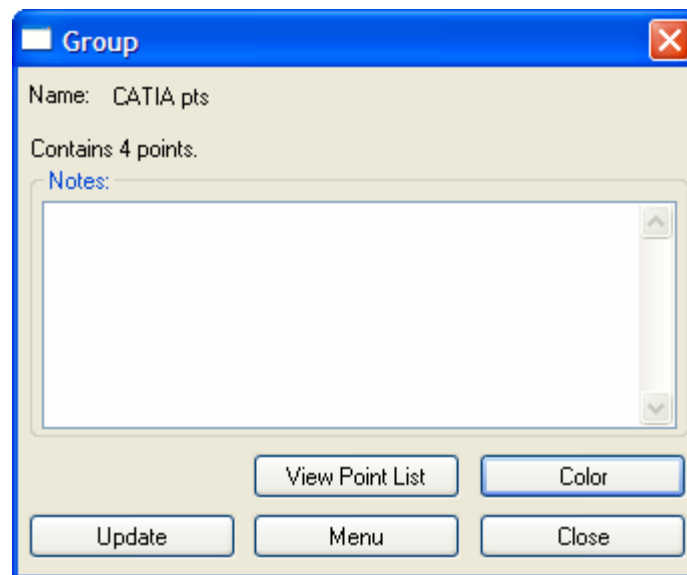
This graphical distinction allows the user to quickly identify and distinguish between measured data and constructed or imported data.

Creating a New Group

There are several ways to create a point/target group. While you are actually collecting data from a measurement device, the instrument interface program will prompt you for a group name and a target name. When a coordinate is added, it will automatically create the specified group (if it does not already exist), and add the measured coordinate to the group. This feature allows the users to quickly organize their data on the fly. Additionally, copying the properties of another pre-existing group of targets and/or points may create groups. File import functions also create groups to contain the incoming data. You can create groups using the Group Manager, which is described in section 0 - Using the Group Manager.

Inspecting the Properties of a Group

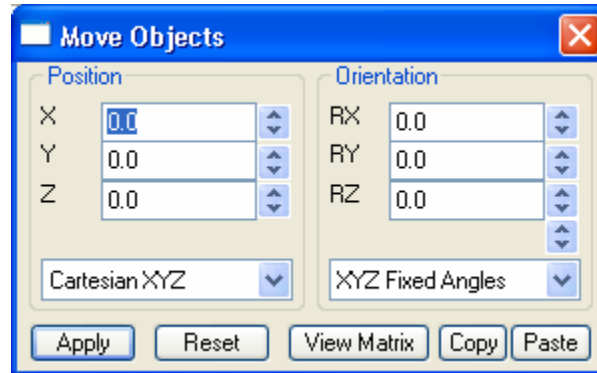
Groups, like other objects within **SpatialAnalyzer**, have properties. To access these properties select the Edit, Object Properties menu selection. With this dialog a user may enter a note for a group, or may change the color of a group.



Group Properties Dialog

Moving, Copying, and Deleting Groups of Data

Once Points and Targets have been organized into Groups, it is possible to move or copy an entire group. To do this, select Move Objects, Copy Objects, or Delete Objects from the Edit menu. Next, you will be prompted to select the object that you wish to move or copy. Click on the Group – which you want to move or copy. The Move Objects dialog appears to allow you to move the Group. If you need to put the group back where it was originally to start over, simply press the Reset button.



Move Objects Dialog

Notice that when a Target is moved, it will be drawn as a point. This is because the data is no longer based on the measurements. This behavior is important to maintain the integrity of the information presentation in **SpatialAnalyzer**.

If you press the Reset button in the Move Objects dialog, the group of targets will return to their original location, but they will not be drawn as targets until you re-compute them.

View Point List

To view a group of points select the group on the tree view and then go to the right-click menu function called 'View Point List.' The figure below shows a group of points in the list viewer.

Group: SURF1

Refresh Copy to Clipboard Export to Text File Edit

Name	X	Y	Z
S133	55.0266	-2.8874	-14.7983
S134	53.3905	-4.0209	-15.0029
S135	52.7585	-4.0622	-16.9011
S136	51.9188	-4.2164	-18.7164
S137	51.1236	-4.2801	-20.5574
S138	50.3766	-4.2521	-22.4225
S139	49.4660	-4.3039	-24.2050
S140	48.4806	-4.3604	-25.9562
S141	47.5295	-4.3361	-27.7206
S142	46.5884	-4.2522	-29.4860
S143	45.5363	-4.2048	-31.2016
S144	44.3589	-4.2138	-32.8316
S145	43.1605	-4.1964	-34.4337
S146	41.9550	-4.1275	-36.0373
S147	40.6149	-4.1329	-37.5327
S148	39.2298	-4.1319	-38.9782
S149	37.8492	-4.0691	-40.4386
S150	36.3517	-4.0783	-41.7733
S151	34.8145	-4.0823	-43.0541
S152	33.2403	-4.0637	-44.3131
S153	31.6151	-4.0657	-45.4821
S154	29.8734	-4.1601	-46.4913
S155	28.0967	-4.2313	-47.4711
S156	26.3741	-4.2095	-48.4833
S157	24.6174	-4.1706	-49.4585

View Group List

This dialog supports sorting the list of points by the point name, X, Y, or Z attributes. To sort by a particular column double click the column header. The sort will alternate between an ascending and descending order.

Within this control you can export either the whole group or selected points to a file with the Export to a Text File button. Copying them to the clipboard is also support. Once in the clipboard they can be pasted into any compatible Windows application (e.g., MS Word, Excel, Access, etc.)

Selecting the Edit button will pass the group to the Point List Edit control.

Editing Point List

Editing a point group in a list can be a convenient means to efficiently setup a point group for a subsequent operation. The Edit Point List control (shown below) enables you to work with a spreadsheet list control to edit points.

Edit Group: SURF 1

Apply Changes Cancel

Name	X	Y	Z	Notes
S49	53.4556	-1.9294	-23.0996	
S50	52.6332	-1.9177	-24.9417	
S51	51.7166	-1.9439	-26.7179	
S52	50.7676	-1.9435	-28.4794	
S53	49.8195	-1.8884	-30.2437	
S54	48.8070	-1.8343	-31.9751	
S55	47.7328	-1.7867	-33.6658	
S56	46.5406	-1.7924	-35.2836	
S57	45.2759	-1.8206	-36.8346	
S58	44.0232	-1.7813	-38.4018	
S59	42.7225	-1.7379	-39.9280	
S60	41.3525	-1.7124	-41.3927	
S61	44.8083	-1.7615	-37.5211	
S62	43.3453	-1.9278	-38.8804	
S63	41.9325	-1.9801	-40.3043	
S64	40.4670	-2.0387	-41.6717	
S65	38.8827	-2.1815	-42.9028	
S66	37.2602	-2.3260	-44.0723	

Edit Group List

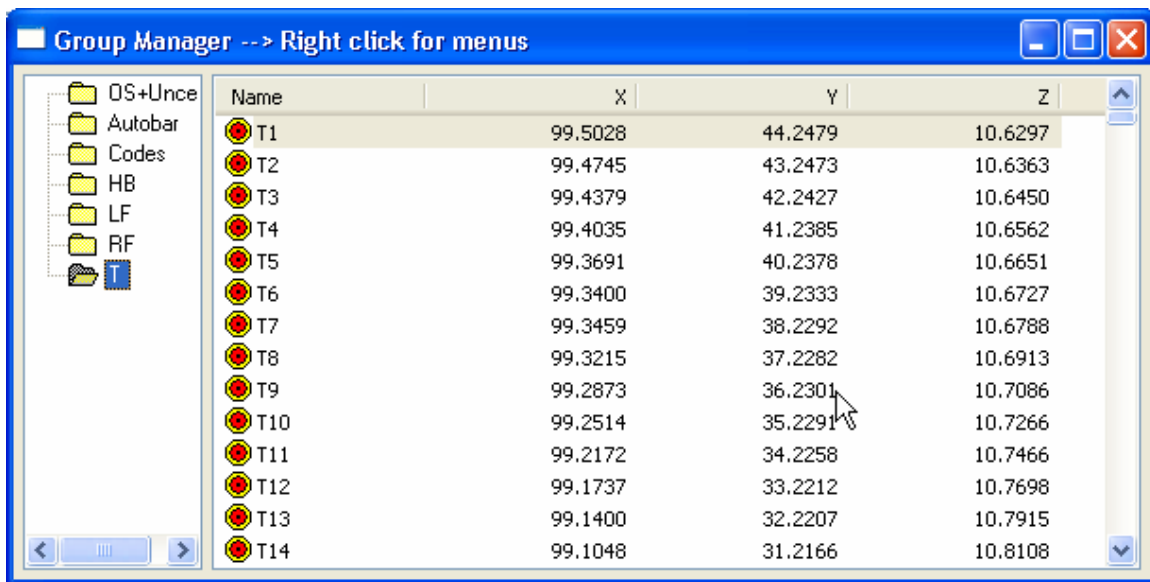
Once you have completed the edits hit the Apply Changes button before exiting the dialog.

Clipboard support is integrated into the Edit Point List functionality. The classic hotkey combinations are supported (e.g., Ctrl-Insert and Ctrl-C for copy; Shift-Insert and Ctrl-V for paste; Ctrl-X for cut; Delete for delete; etc.)

Using the Group Manager

The group manager provides a familiar Windows Explorer like environment for creating, moving, copying and deleting groups from within a **SpatialAnalyzer** job. Operations discussed here are slightly different than those discussed above in that the group manager interface can never modify coordinate values in the way the transform dialog can. Here we are merely discussing the common operation of regrouping or renaming.

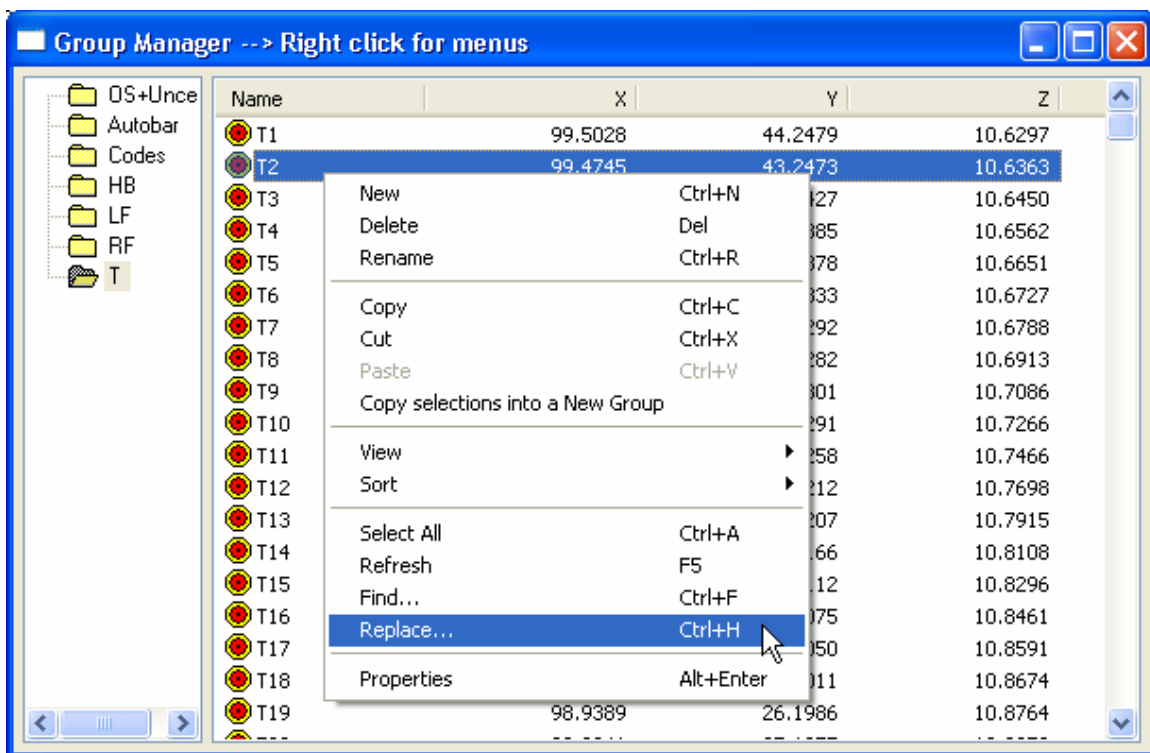
The figure below shows the group manager interface. The left pane contains a list of all groups within the current job. The right pane contains a list of all the targets and points within the selected group. To start the group manager interface, select the Edit, Group Manager menu option. The interface will appear. Clicking on any group will update the right pane to show you the selected group's contents.



Group Manager

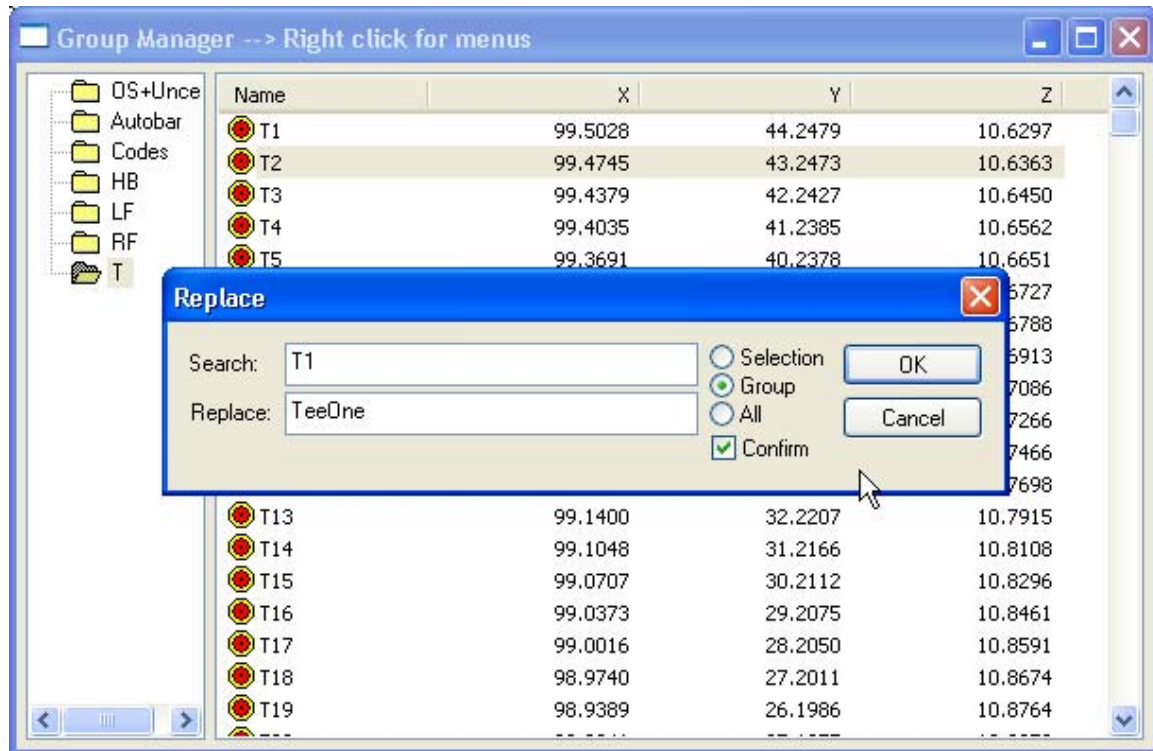
Renaming Points and Targets

Points and target can be renamed with the Replace function within Group Manager. When your mouse is in the right pane either right-click to get the menu or hit the hotkeys Ctrl-H to get the Replace dialog to show up.



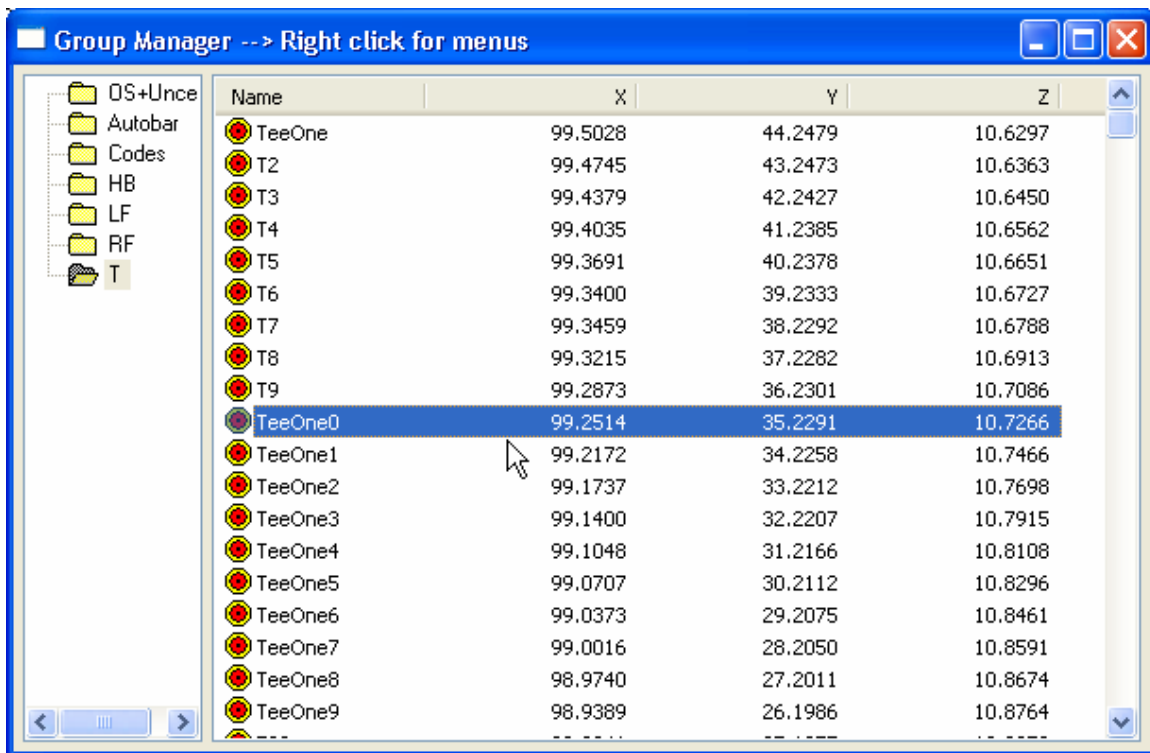
Right-Click menu window

With the Replace dialog up, enter the character string that you want to search for and the string to replace it with into the dialog fields. The figure below shows an example. The function will search all of the targets in the group for the string “T1.” When the string “T1” is found it will be replaced by the string “TeeOne.”



Right-Click menu window

The figure below shows the results based on the request above.

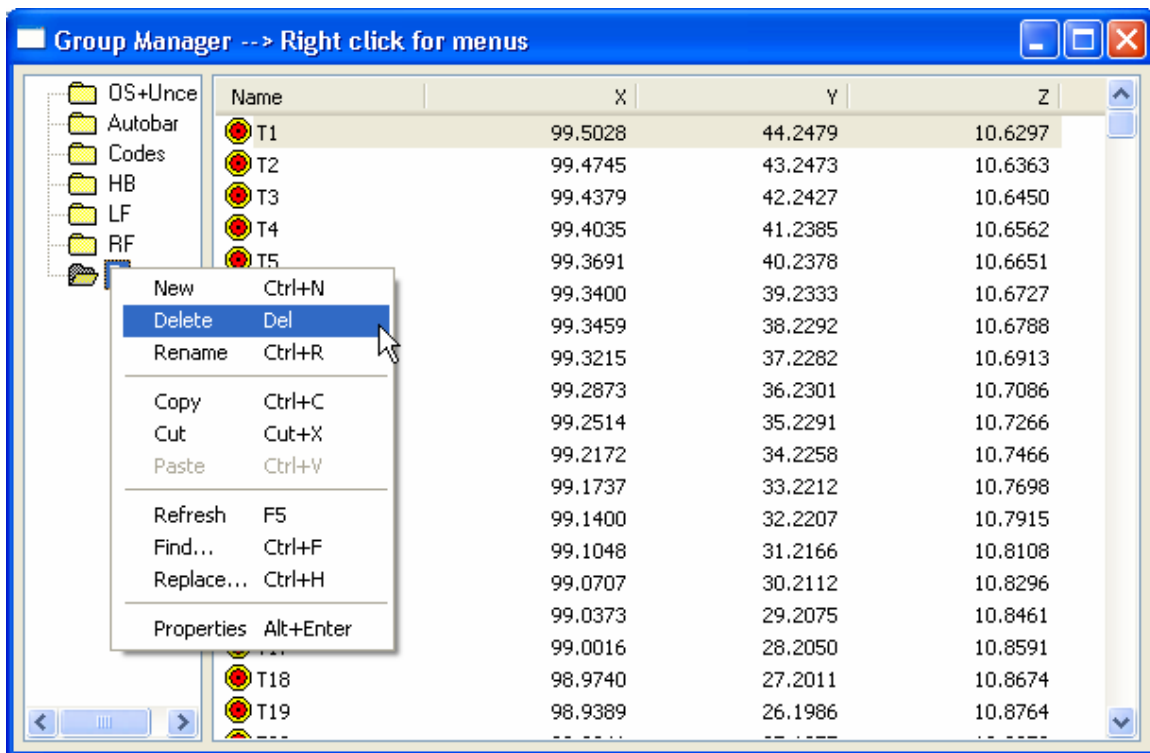


Right-Click menu window

Note that only the targets with the "T1" character were changed. This Renaming process can be expanded to search the entire database or limited to just selected targets with the radio button option for "All" and "Selection", respectively.

Deleting Groups or Individual Points

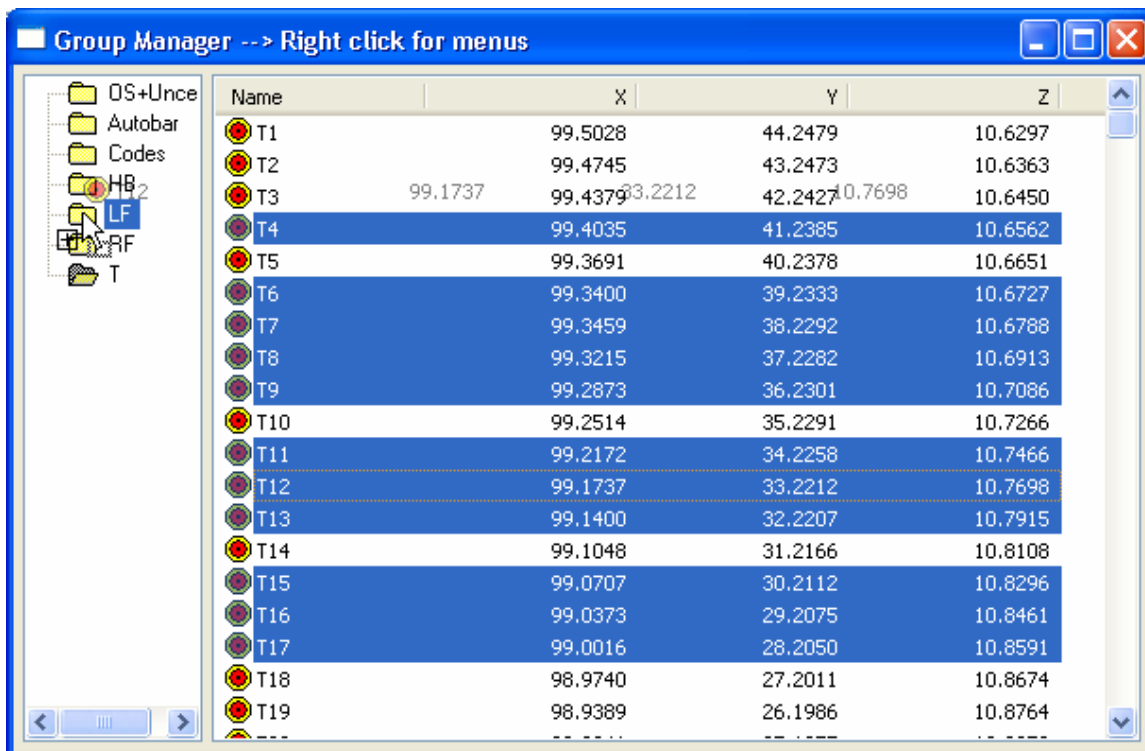
To delete an entire group, highlight the group in the left pane and hit the DEL key. To delete a selection of targets or points, highlight them in the right pane and hit the DEL key. Before the points are deleted, **SpatialAnalyzer** asks you to confirm with a Yes/No message box, which you're sure that you want the points deleted.



Deleting group

Moving Whole Groups of Data

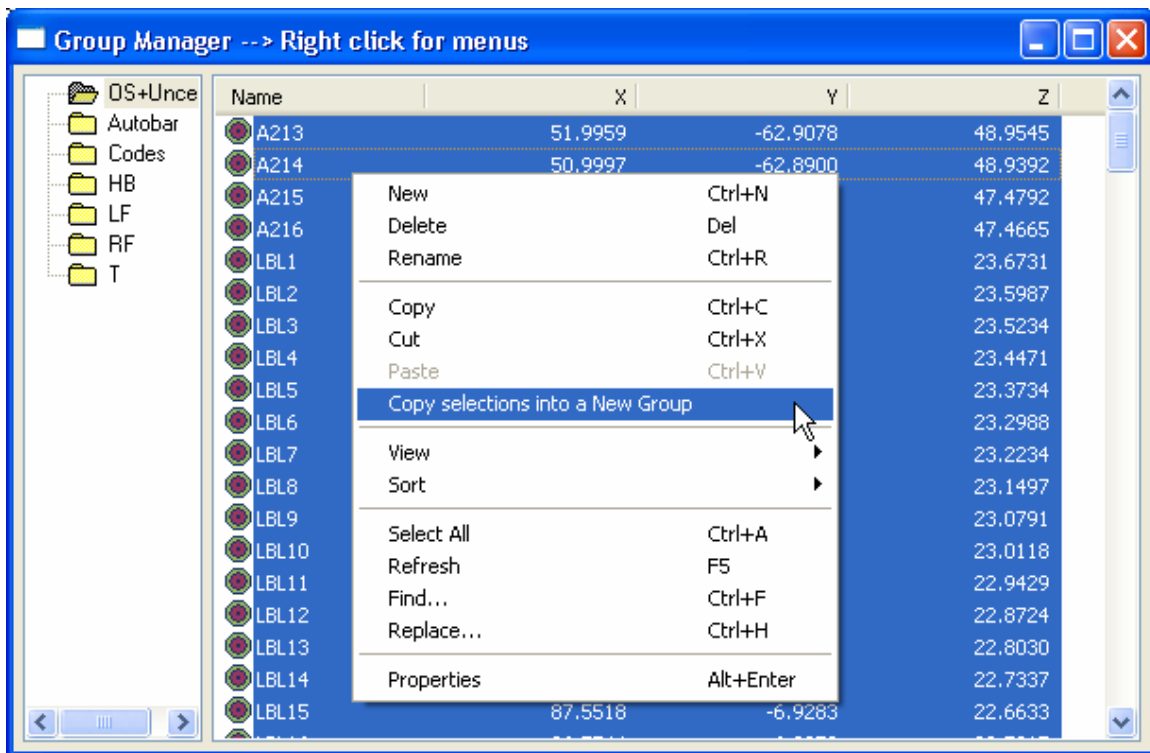
To move an entire group of data, highlight the source group, hold down the right mouse button and drag to the destination group. The figure below shows a number of points being moved to the Main group. Both groups will remain in tact, but the points and targets in the 'ref' group have now been transferred to the Main group. In the case of duplicate names, an asterisk will be appended to the newly arriving point name.



Move group of points and targets from the ref group to the Main group

Copying Whole Groups of Data

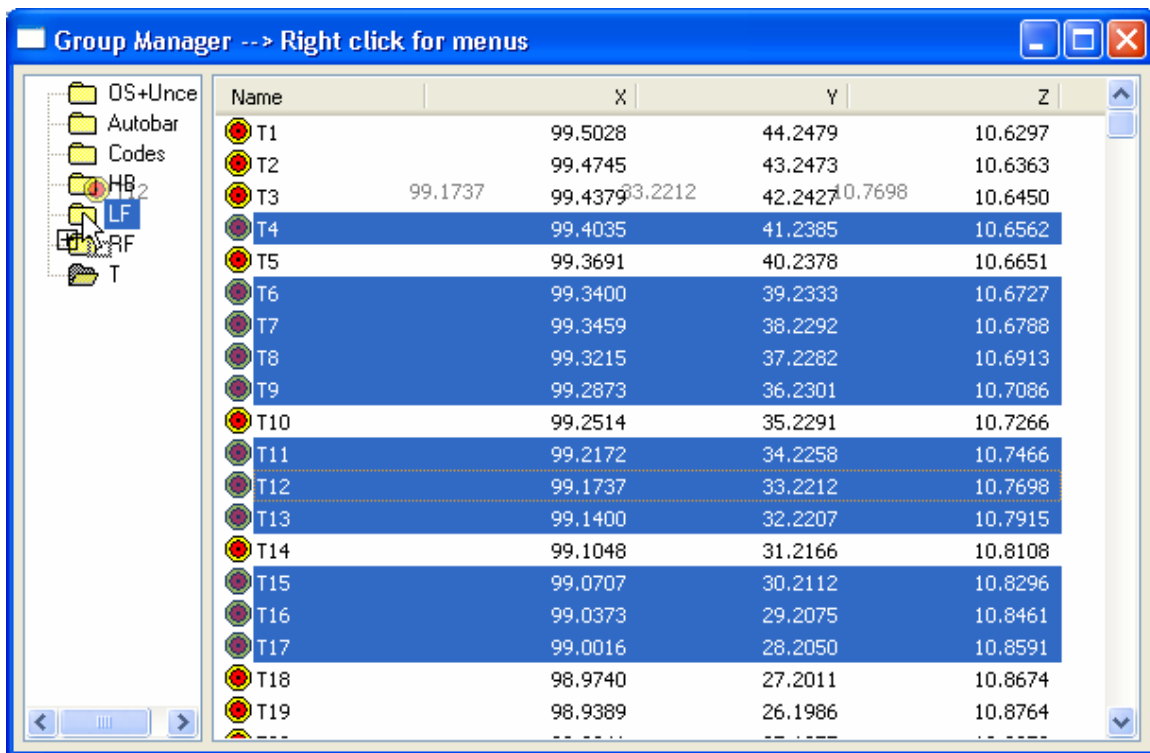
To copy an entire group of data, highlight the source group, press and hold the Ctrl key, and hold down the right mouse button to drag to the destination group. Again, both groups will remain in tact, but the contents of the source group have now been copied to the destination group. In the case of duplicate names, an asterisk will be appended to the newly arriving point name. Note that in accordance with our earlier discussed convention, all copied targets will become points in the destination group since there are no observations of these points from any instruments. The source targets will remain targets.



Coping the entire Ref group into the New group

Moving Individual Points and Targets

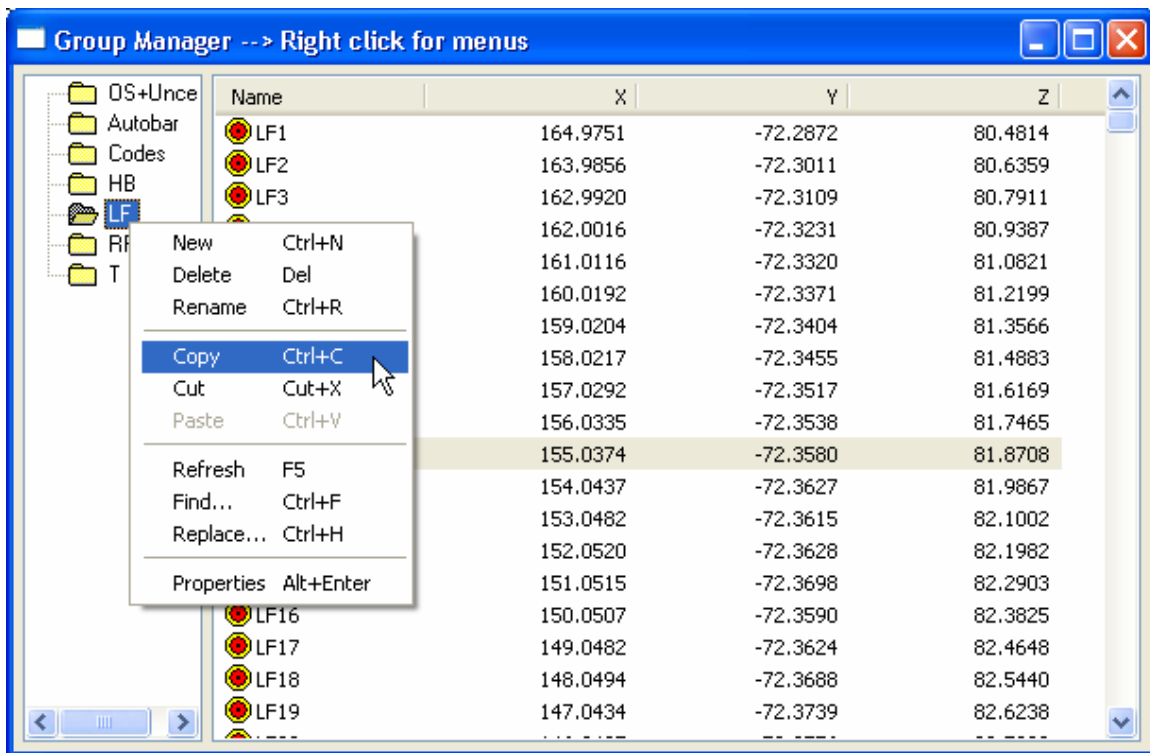
To move selected points and targets, highlight the items in the right pane, hold down the right mouse button and drag to the destination group (in the left pane). The figure below shows a selected set of points and targets moving from the ref group to another group. In the case of duplicate names, an asterisk will be appended to the newly arriving point name.



Coping selected points and targets

Copying Individual Points and Targets

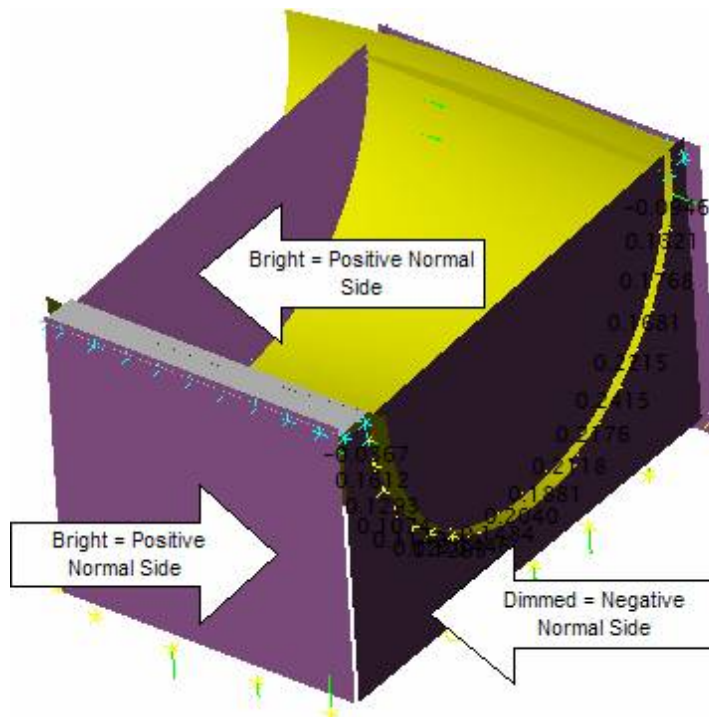
To copy selected points and targets, highlight the items in the right pane, press and hold the Ctrl key and hold down the right mouse button and drag to the destination group (the left pane). The figure below shows the LT point group being copied. The group can then be pasted in the point group window. In the case of duplicate names, an asterisk will be appended to the newly arriving point name. Once again, copied targets will revert to points in the destination group.



Copying a group of points

Surface Normals (Direction)

SpatialAnalyzer shows the sense of the surface's normal direction by presenting the positive side of the surface as Bright and the negative side Dim. There is an example shown below.



Surface Back-Sides Dimmed

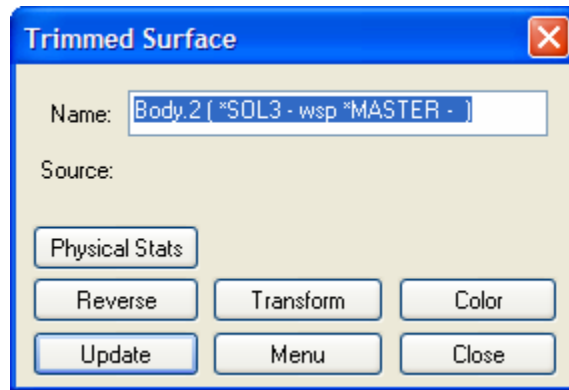
This behavior of showing the positive and negative sense of surface as bright and dim (respectively) can be turned on the User Options dialog. By turning off the Display option for showing Surface Back-Sides Dimmed all of the surfaces are then shown with the Bright color.

☒ Surface Back-Sides Dimmed

User Options >> Display Tab for Surface Back-Sides Dimmed

Reversing Surface Normals

Surface normals can be reversed on the surface's properties dialog with Reverse button. Select the object's properties dialog and then the Reverse button on the dialog. The function is shown in the figure below.

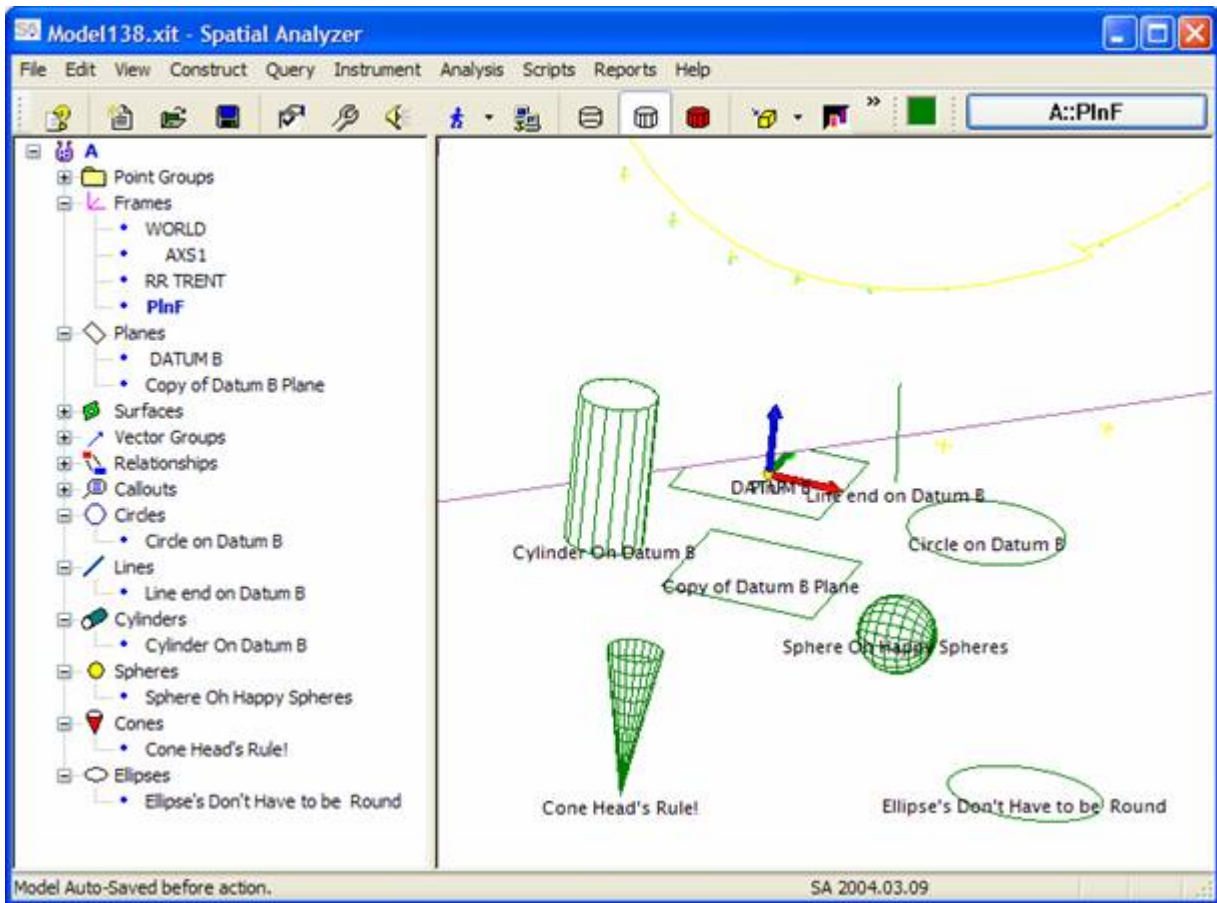


Surface Properties dialog Reverse Surface Normal

There are also functions found on the Surface Normal Conditioning menu option under the Edit menu that help set the Surface Normal directions. The Indicate Point on Positive (or Negative) Side function allows you to select a point and then select the surfaces to condition. The next step is to indicate if the point lies on the positive or negative side of the surface. SA will then project the point on to each surface and if the projection indicates the point is on the alternate side then its surface normal is changed to be consistent with your request. To reverse a number of surfaces at the same time select the Reverse Surface Normal Direction option from the Edit >> Surface Normal Conditioning menu. After selecting all of the surfaces that need to be changed SA reverses each selected surface's normal direction.

Show Object Labels

The show object labels function on the View menu displays object names. An example is shown in the figure below.



Show Object Labels

Querying Point/Target Data

SpatialAnalyzer contains many powerful analytical features. These allow you to represent data relative to any coordinate frame, determine the error tolerances on each measurement, and fit a wide variety of geometric objects to the measured data.

This section will focus on the features related to querying the target data and assumes you are already familiar with acquiring from instruments or importing the data from CAD files.

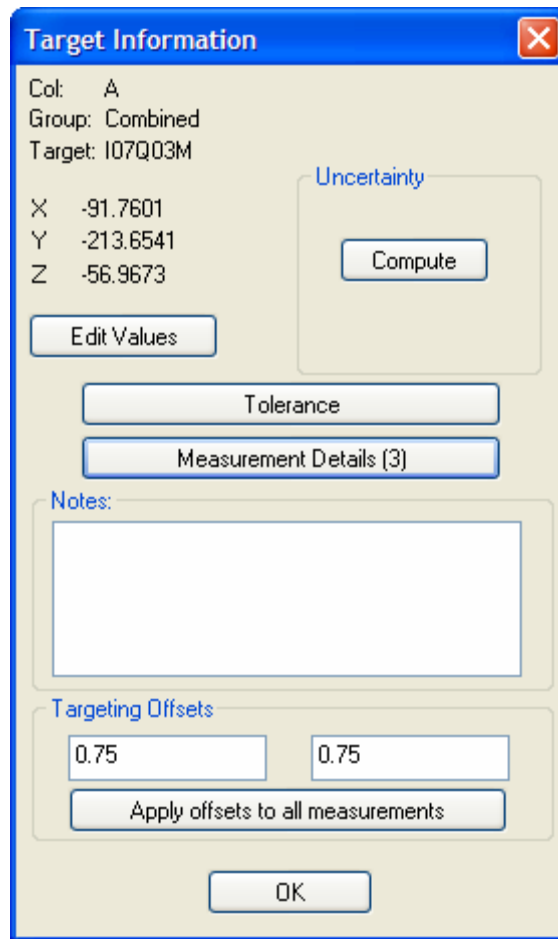
Querying Target Information

Once enough measurement data is available for a given target, **SpatialAnalyzer** will automatically compute the location of the associated target. At this time, a target will appear in the graphical environment. During or after the measurement process, you may query information about the target using one of the following methods.

Querying a Single Target

If you wish to query information about a target, double click in the graphics view near the target and the Information dialog will appear as shown in the figure below. You can also access this dialog from the database tree view, by right clicking on a point and then

selecting Properties from the pop-up menu. This window lists the target's group name, the target name, the XYZ coordinates of the target relative to the current Working Coordinate Frame, and the quality of the measurement.

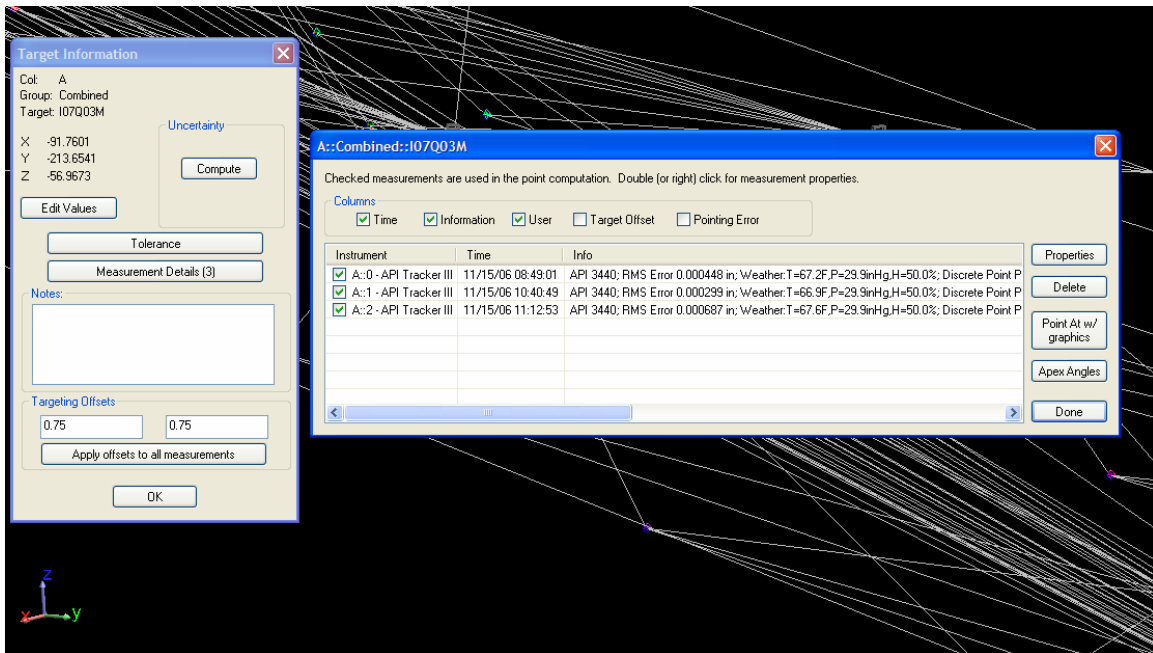


The image shows a software dialog box titled "Target Information" with a standard Windows-style title bar (blue background, white text, and a red close button). The dialog has a light beige background. It contains the following elements: A text area at the top left with the text "Col: A", "Group: Combined", and "Target: I07Q03M". To the right of this is a section titled "Uncertainty" in blue text, containing a "Compute" button. Below the text area is an "Edit Values" button. In the center, there are two stacked buttons: "Tolerance" and "Measurement Details (3)". Below these is a "Notes:" label followed by a large, empty white text box. At the bottom of the dialog is a "Targeting Offsets" section containing two input fields, both with the value "0.75", and an "Apply offsets to all measurements" button. At the very bottom center is an "OK" button.

Target Properties Dialog

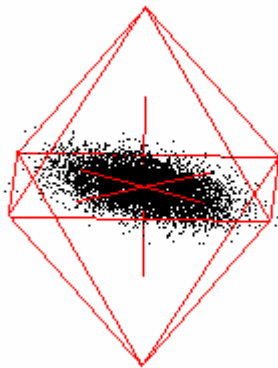
To query the location of the target relative to a different coordinate frame, close the dialog, and change the current Working Coordinate Frame. When you do this, all coordinate data will be represented relative to that frame. This powerful feature allows the user to instantly verify measurement data relative to a variety of meaningful reference systems.

You have seen that by double clicking on a target you can obtain its location and measurement quality value. From the target information window, you may also press the Details button for more in-depth information about the target. The Details button brings up another dialog as shown in the figure below. This information includes the error sensitivity for each of the coordinate axes and a list of each measurement used to compute the target location.



Target Details Dialog

Note that the confidence interval used to determine the error bounds is displayed along with the number of samples used to form the statistical data. The coordinate uncertainties for the X-Y-Z target location are displayed in the current working frame. To see this information relative to another coordinate frame, close the window, change the working frame, and double click the target again. The target details dialog also provides the user with an easy way to deactivate and activate individual observations of a target. In addition, the user may also hit the draw button and produce a graphical representation of the coordinate uncertainty at the point. The figure shows the uncertainty cloud for the target referenced above.

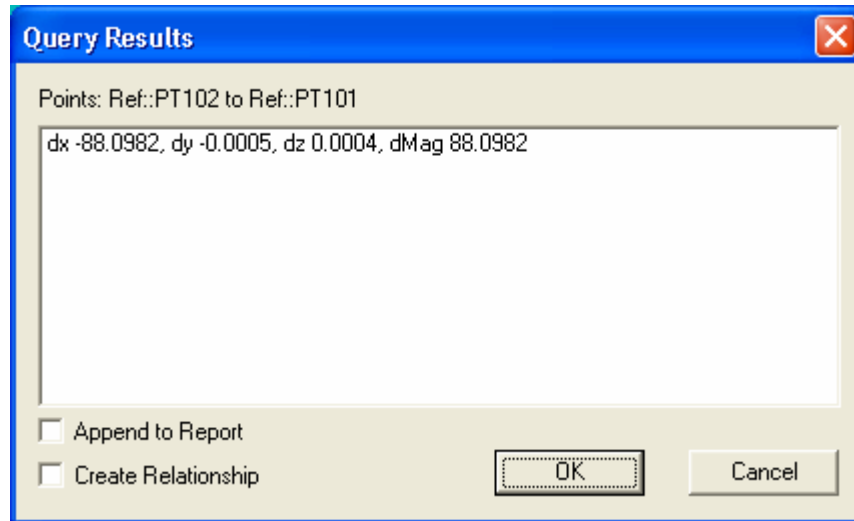


Uncertainty Cloud at a Target

Querying Relationships between Targets

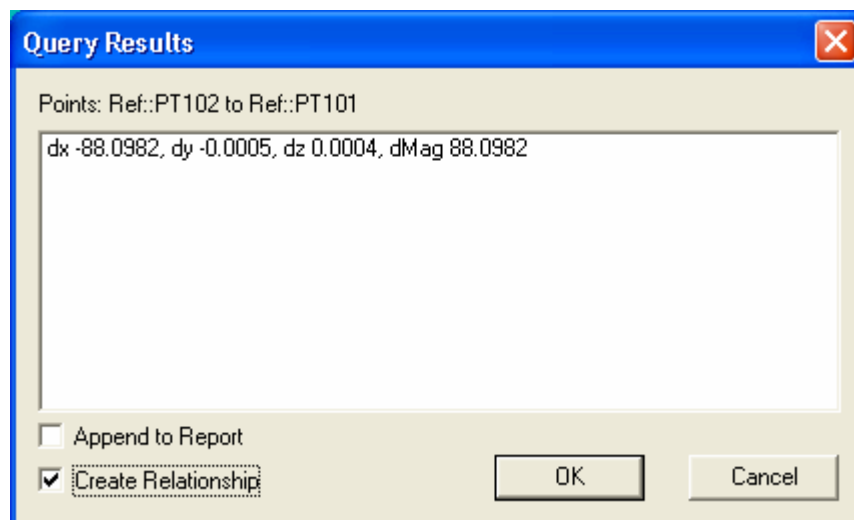
You may use the query menu to determine the vector and distance between any two of the measured targets in the workspace. Note that this function will return the information relative to the current working frame.

Select Query, Point to Point (Relationship). You will be prompted to select the points you are interested in. Once you have selected the points, **SpatialAnalyzer** will display the XYZ distance from the first point to the second point and the magnitude of the vector in a Query Results dialog.



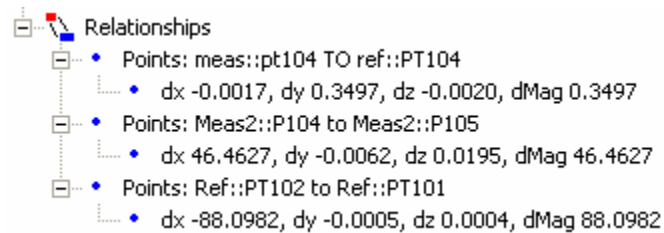
Point to Point Query Results Dialog

SpatialAnalyzer allows you to save this query in the form of a “Relationship.” By checking the box in the lower left corner of the Query Results dialog, **SpatialAnalyzer** creates a Relationship object that saves the query.



Create a Point-to-Point Relationship

Relationships are objects that remember specific queries. Once a query is saved as a Relationship, you don't have to repeat the process of selecting the individual entities every time you want to know the results. Relationships are accessible on the database tree view and in reports. The figure below shows an example of the Relationships tree view with several Point to Point relationships.



Point-to-Point Relationships in the tree view

If any of the entities within a Relationship change, **SpatialAnalyzer** updates the results automatically. Relationships are useful for more than point-to-point queries, so we will cover them in more detail in a subsequent chapter.

COLLECTIONS

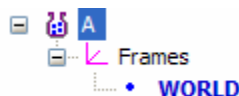
The term Collection is used in **SpatialAnalyzer** to refer to a grouping of objects that you can manage together from the database tree view. A collection can contain any of the objects within a **SpatialAnalyzer** job (e.g., Frames, Instruments, Point Groups, Surfaces, Vector Groups, Charts, etc.) Effectively each collection can be a separate and distinct job. However since each **SpatialAnalyzer** job can contain numerous collections you can also use them together to organize an entire series of measurements of a common object. As an example, a collection with nominal references is used by a whole series of collections each one with a complete set of measurements and analysis of the object. Yet another collection in the same job file can then track the variation of key features between each measurement set.

The concept of collections of objects in **SpatialAnalyzer** is architected in a general way. Several of the primary applications for Collections include:

- Managing a series of measurement jobs each in its own collection in single **SpatialAnalyzer** job file. Collections help organize data to either track an object's changes through a process or track variation between different units of a part.
- Managing the nominal vs. actual dimensional configuration for a series of components in within an assembly. Each component and its measurements can be defined as collections in a job. Analysis and best-fit optimization directly against critical fit and performance features can be done between the parts and their CAD Model.
- Measurement network dependency can be structured in collections. When using measurements from one system to orient another it is often useful to keep the job file well organized. Using one collection to orient the primary instruments and other collections to orient the secondary systems is a useful technique for keeping the relationship between the systems manageable.

Adding Collections

The figure below shows the tree view for a default **SpatialAnalyzer** job. By default the first collection is initially named A.



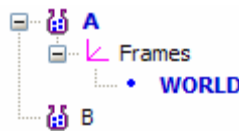
Default Collection → A

In this case the collection only contains the World frame. By right-clicking in the database tree view you can create a New Collection from the right-click menu. The figure below shows the right-click menu (click in an empty area of the tree view.)



Right-click menu to create a New Collection

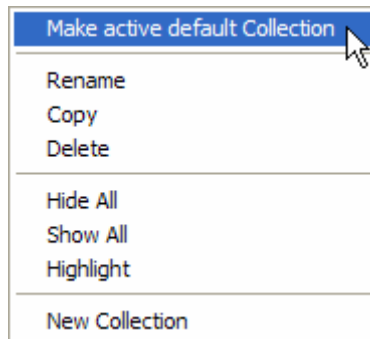
Adding a second collection, i.e., collection B, to the job the tree view look like that shown in the figure below.



Added second Collection

Activating a Collection

Only one collection is considered the active one at any point in time. Once active any new items added to the database are put into this collection. To activate a collection, first select it on the tree view and then use the right-click menu option to 'Make active default Collection.' The figure below shows the right-click menu for making a collection active.



Menu to Make active default collection

The active collection is shown with a bold font in the database tree view. The figure below shows collection B as the active collection.



Collection B as the active default collection

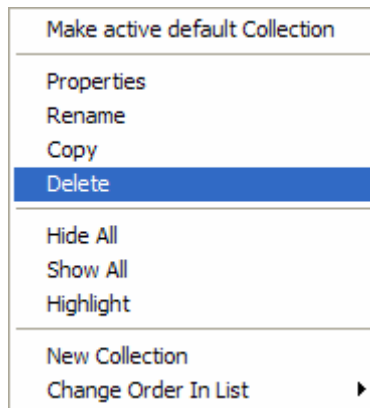
While B maybe the active collection the working frame is defined from a frame in collection A. Even through there can be numerous collections, there is only one working frame active in SA. All coordinates information and reports are relative to the working

frame. Any new objects (e.g., Points, Instruments, Planes, Lines, and Surfaces etc.) are added to the active collection.

Deleting Collections

The process for deleting a collection is consistent with other object types in SA. While the mechanism to delete a collection is similar; care should be taken when deleting a collection with instruments or measured targets from instruments in other collections. The deletion of an instrument with measurements in another collection deletes the measurements to the targets. The targets measured by the instruments are converted to points.

The Delete function for collections is accessible on the right-click menu when it is selected. The figure below shows the right-click menu with the Delete function highlighted.

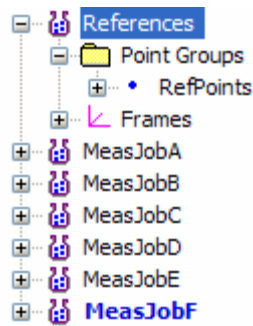


Delete Collection Function

If the deleted collection was the active one, the active flag is moved to the first collection in the database.

Renaming Collections

The default names for collection follow the alphabet. Renaming a collection is accomplished with the Rename function on the right-click menu. A collection name is an alpha-numeric string of characters (including spaces). The naming restrictions for objects in SA are the same as those for file names in the operating system. The figure below shows an example of collection naming.



Renamed Collections

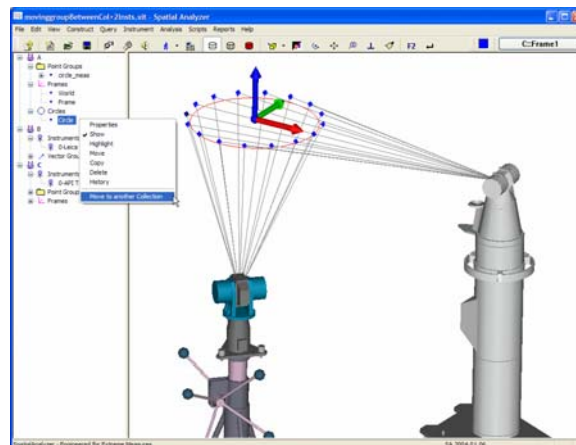
Importing Jobs into Collections

Importing one SA job into another is a common process when different surveys need to be compared. To import an SA job file into the current job use the Import >> SpatialAnalyzer Job File function found on the File menu. A new collection is added to the job for each one in the file being imported. This mechanism keeps the objects in each file organized into their own collections. While they are initially imported into separated collections the objects can be accessed for analysis or auto-measurements.

Copy and Move data between collections

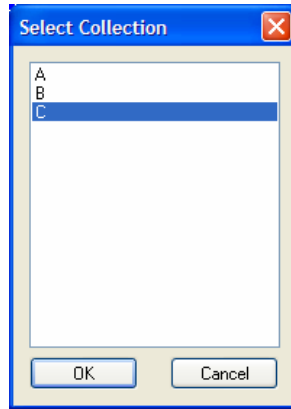
There are a number of methods to copy and move objects between collections. When objects are copied (with any of the copy methods in SA) the active collection is the destination for copied objects.

Moving objects from one collection to another is accomplished with the right-click menu available for each object. The figure below presents an example where the 'Circle' object in collection A is going to be moved to collection C.



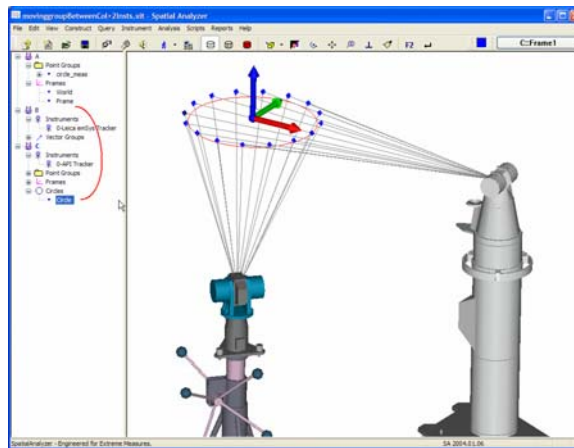
Moving Objects between Collections

After selecting the Move to another Collection function select the collection that you want to move the object to from the Select Collection dialog. The figure below shows an example.



Collection selection for Moving Objects

The results of this object move are shown in the figure below.



Moved Object between Collections

CONSTRUCT

All of the geometric entities in **SpatialAnalyzer** are derived from strict mathematical definitions. Each entity has a set of parameters that uniquely determine the attributes of that object. For example, a point has X, Y, and Z coordinates; a sphere however, requires X, Y, and Z coordinates and a radius. In addition to these geometric parameters some entities require graphical parameters that are only used to alter the way the geometry is portrayed. For example, the number of facets used to represent a sphere, or the color an object is rendered in the view.

For each of the entities below, SA provides:

1. Summary table listing all pertinent information, useful as a quick reference for experienced users
2. Concise mathematical formulation of that entity
3. Parameters needed to define the geometric entity
4. Description of the different ways in which these entities can be created

Properties Common to all Geometric Entities

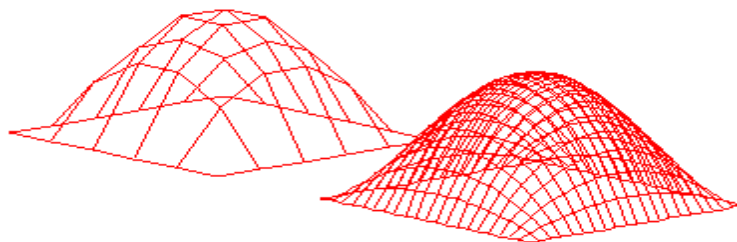
Naturally, geometric entities of different type share many of the basic features needed to define an object in general.

Color Property

All geometric entities have a color. We distinguish between the two only to highlight those Colors can have surface characteristics such as a shinny or flat surface finish.

Facet Count

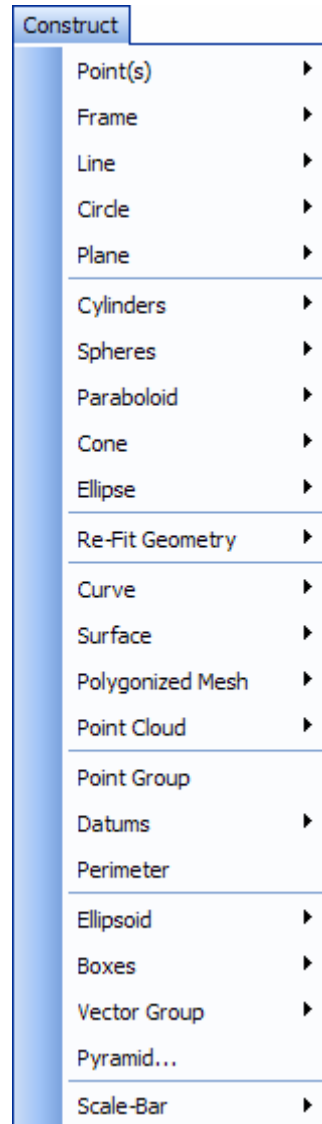
All solid objects are also drawn using facets as the basic elements. Facets are what you see if you view an object in wire frame graphics mode. The figure below shows the effect of increasing facet count for a basic surface.



Effect of Facet Count

Construct Geometries

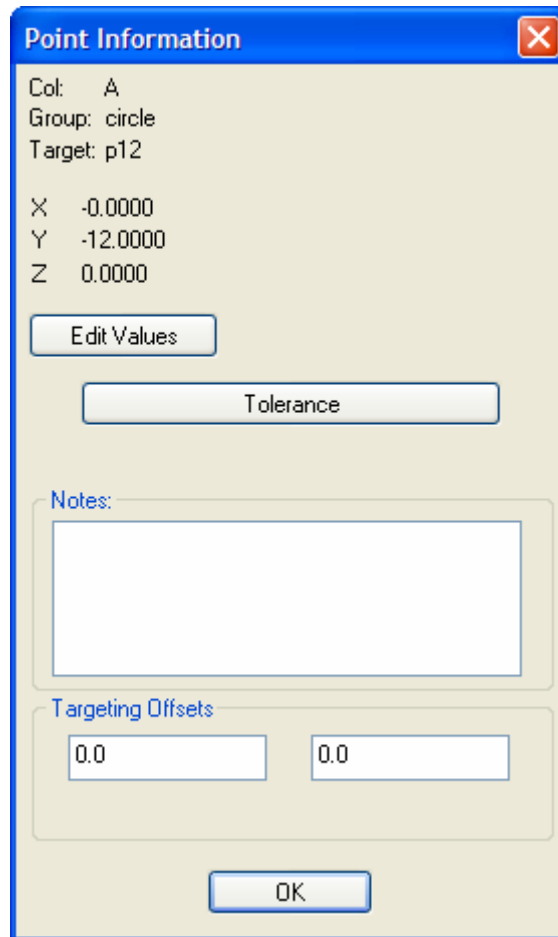
SpatialAnalyzer has a vast array of geometry construction methods that allow you to create geometry from measurements, imported CAD data, or hand-entered geometries. The figure shows the Construct menu in **SpatialAnalyzer**. The sections following this one will cover each construction method.



Construct Menu

Points and Targets

A point is simply defined as an X, Y, and Z coordinate in space. The property dialog for a point is presented in the figure.



Point Information

Col: A
 Group: circle
 Target: p12

X: -0.0000
 Y: -12.0000
 Z: 0.0000

Edit Values

Tolerance

Notes:

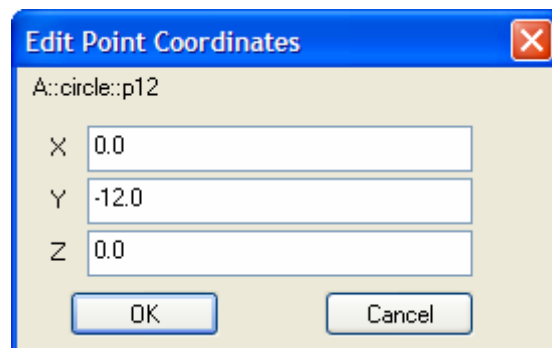
Targeting Offsets

0.0 0.0

OK

Point Properties Dialog

Hitting the modify button in this dialog will result in the edit point coordinates dialog of figure.



Edit Point Coordinates

A::circle::p12

X: 0.0
 Y: -12.0
 Z: 0.0

OK Cancel

Edit Point Coordinates Dialog

Points and targets share many of the same attributes with one clear distinction. Points result from computed data, targets result from measured data. For the following

discussion of points and targets we will refer to both simply as points, drawing a distinction between the two types only when necessary.

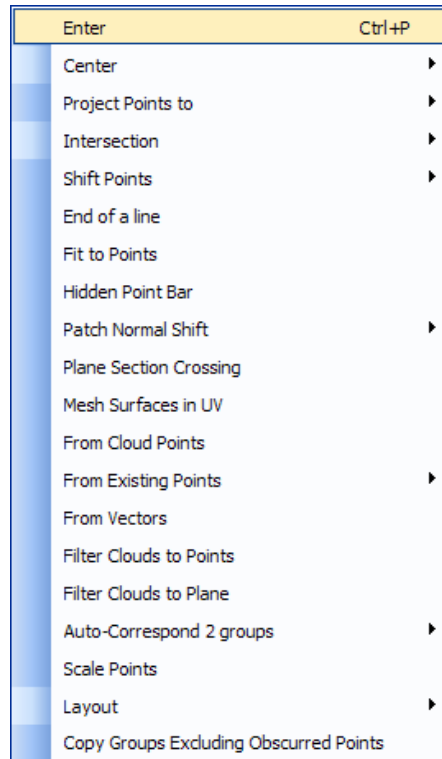
The mathematical formulation for a point is trivial; a point has only X, Y, and Z coordinate information. These coordinates are always reported to the user in the current Working Coordinate Frame, but may be presented in Cartesian, Polar, or Cylindrical coordinates.

SpatialAnalyzer organizes points into groups. All points must belong to a group and must have a unique name within that group. Both group names and point names are completely alphanumeric. It is possible, and in many cases desirable to have two points with the same name, as long as they belong to different groups. A group may contain any number of points. Points will inherit the Color properties of the group to which they belong. Forcing all points to belong to a group allows the user to quickly perform complex operations on a large number of points with few commands. For example, you can best fit a group of measured targets to a group of design points.

Constructing Points and Targets

There is only one way to add a target to a model. Whenever a measurement is taken and sufficient information is present, a target will be automatically added to the model. If further data is collected with regard to the target, its position will be appropriately modified.

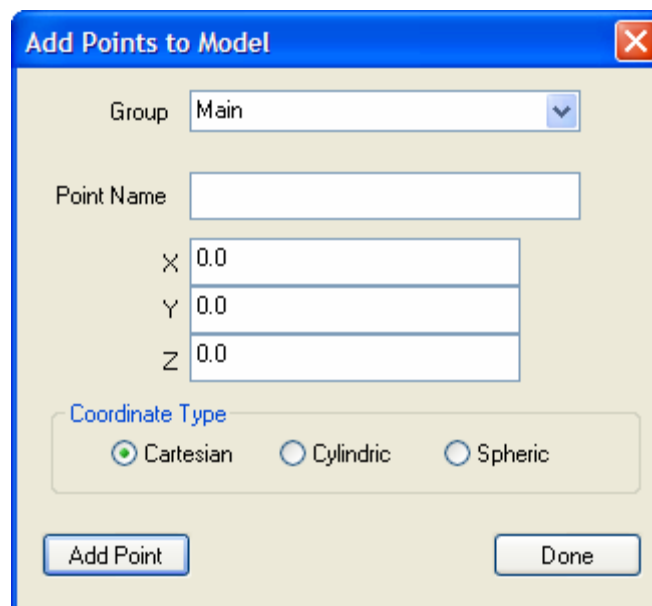
Conversely, there are several ways to add points to a model. You may create copies of measured points using the Edit, Copy Objects menu selection. Note that when you copy a target, the copy is a point and not a target. The copied coordinate was never measured.



Construct Points Menu

Entering Coordinates

As with other types of geometries, you may create a point by entering its coordinates. To do this select Construct >> Point >> Enter. You will be prompted for a group and point name and coordinates in the Current Working Frame with the dialog.



Add Points Dialog

Using this dialog a user may repetitively enter point coordinates without leaving the dialog. Select either the Cartesian, Cylindrical or Spherical coordinate type.

Center Point Methods

A series of functions that create points at the center of other objects can be a useful tool for metrology and analysis tasks. The descriptions below describe these functions.

Center of a Circle: You may create a point that is the center of a circle. To do this select the function from menu at Construct >> Point >> Center >> Circle. You are prompted to select the desired circle. Double click your left mouse button on the desired circle or select using the F2 function. Immediately, you will see a dialog showing you the coordinates of the center point in the Current Working Frame. Enter a point name for this new point.

Center of a Line: You may create a point that is the center of a line as well. To do this select Construct, Point, Center and Line. You will immediately be prompted to select the desired Line. Double click your left mouse button on the desired line or select using the F2 function. Immediately, you will see a dialog showing you the coordinates of the center point in the Current Working Frame. Enter a point name for this new point.

Center of a Sphere: You may create a point that is the center of a sphere. To use this function select Construct >> Point >> Center >> Sphere from the menus. You will immediately be prompted to select the desired sphere. Double click your left mouse button on the desired sphere or select using the F2 function. Immediately, you will see a dialog showing you the coordinates of the center point in the Current Working Frame. Enter a point name for this new point.

Center of a Cylinder: This function creates a series of points that are centered in the cylinder. There are two points at either end and another point at the midpoint. The points are labeled EndA, EndB and Center, respectively and are automatically put into a group with the same name as the cylinder.

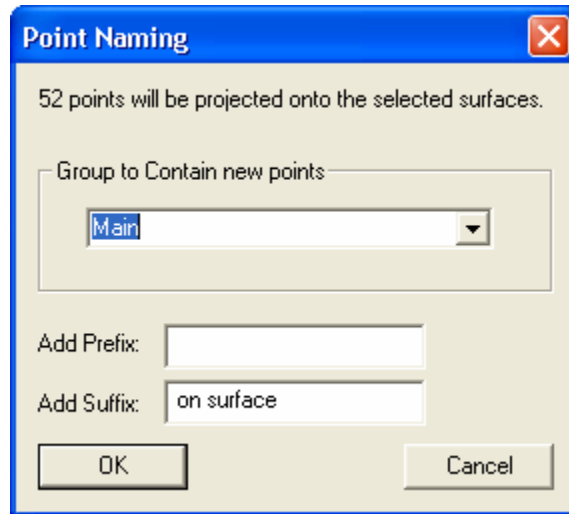
Point Projections Methods

A series of functions that create points by projecting them to other objects can be a useful tool for metrology and analysis tasks. The descriptions below describe these functions.

Project Points to Surfaces Closest Point and to Curves: The functions projecting points to surfaces and curves follow the same procedure. The manual will describe the process for projecting points to surfaces. Please note that the same process is applicable for projecting points to curves.

Points are created by projecting them onto selected surfaces. The function searches the surface for the point that has minimum distance between each selected point and surfaces. The process is select Construct, Point, Project Points to, Surfaces, Closest Point. Select the desired surfaces that you want to project points too. Select the surfaces using the graphical techniques or with the F2 selection tool. Use the enter key to end this phase of the selection process. Select the points to project and press enter. Next a

Point Naming dialog is presented to allow the point naming to be controlled. Select or enter a new group name. You can add a point name prefix and/or suffix to each projected point. Note the prefix and suffix are optional.



Point Naming Dialog

The next step allows you to enter a Probe Offset value for each point. Each created point will be projected along a line between the initial point and the point on the surface by the amount entered into this dialog. A value of 0.0 means the created point will lie on the surface.



Probe Offset Value Dialog

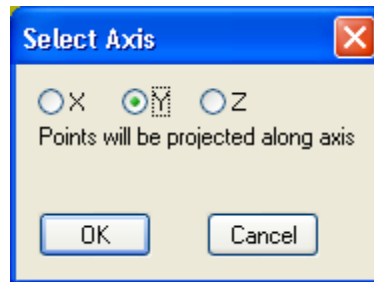
After the Probe Offset Value is entered the points are projected and created with the name and group attributes defined in the naming dialog.

Project Points to Surfaces Relative to Coordinate Axes Methods: The functions projecting points to surfaces relative to coordinate axes follow the same procedure. They differ in the type of axis used to define the direction and orientation of projection. The projection mechanisms are described in the sub-sections below.

The points are created by projecting them onto selected surfaces along a specific axial direction. If the projection intersects the surface more than once the function keeps the point closest to the initial point. Start the process by selecting Construct >> Point >> Project Points to >> Surfaces >> Relative to Coordinate Axes from the menu. Select the

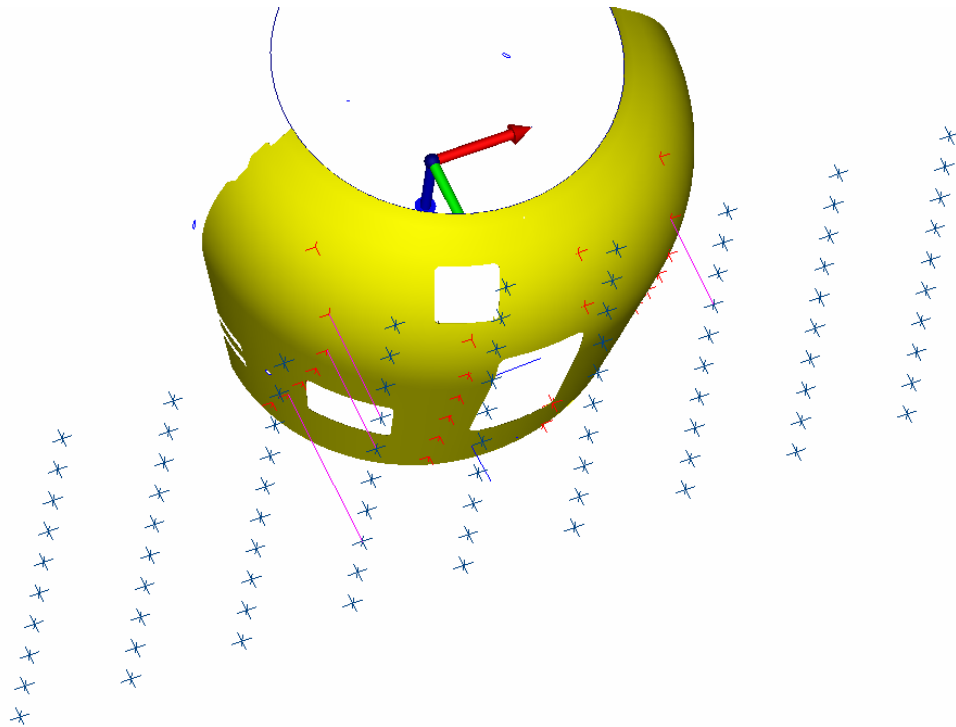
desired surfaces that you want to project points too. Use the enter key to end this phase of the selection process. Select the points to project and press enter.

Parallel to WCF Axis: The next step allows you to select the Axis that the points will be projected along. Each created point will be projected along a line parallel to the axis and passing through the selected surfaces. If the projection intersects the surfaces more than once, the point closest to the input point is created. If the projection does not intersect the surface then a point is not created.



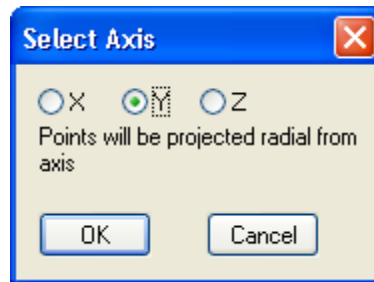
Dialog to select Parallel Projection Axis

The figure below show an example of points projected parallel to a current working frame axis. In this case the axis was the Y.



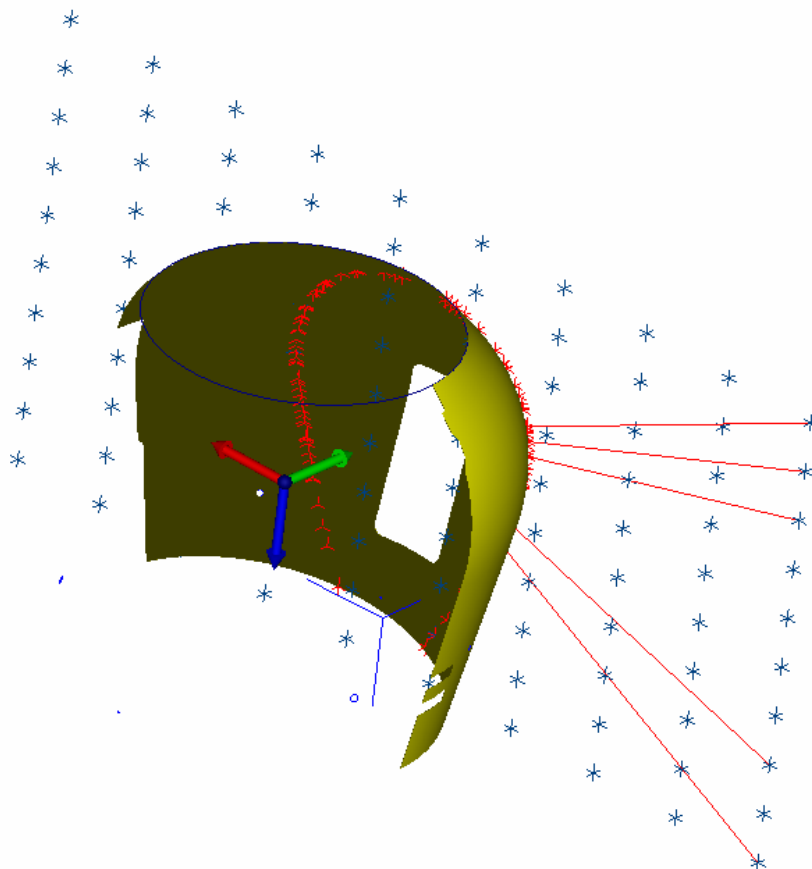
Points Projected onto Surfaces Parallel to Axis (e.g., Y-Axis)

Radial to WCF Axis: Select the axis that the points will be projected radially from. Each created point will be projected along a line extending radially from the selected axis and passing through the selected surfaces.



Dialog to select Radial Projection Axis

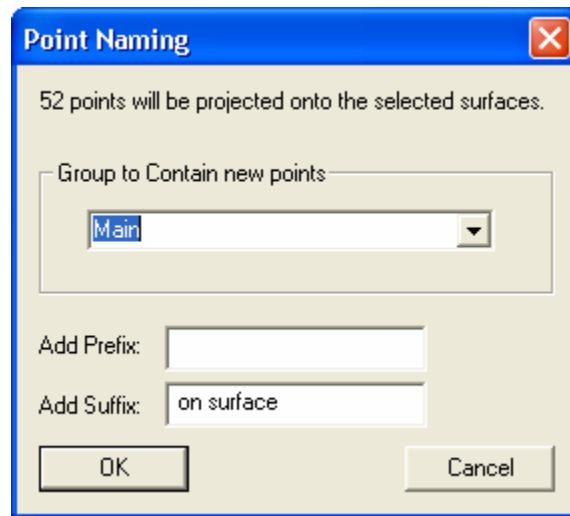
The figure below provides an example for this projection function. The created points lie at the intersection of a radial line that is normal to the selected axis (e.g., Y in this case) and the surface.



Points Projected to Surfaces Radial to WCF

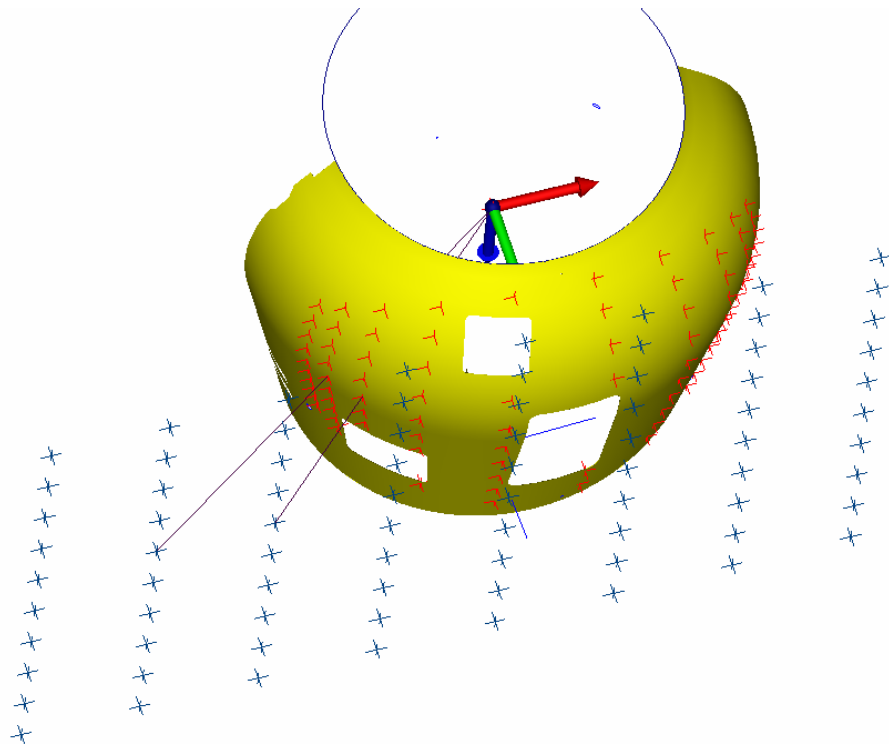
Polar from WCF Origin: A line is computed between each input point and the working frame origin. A point is created at the intersection between the line and surface(s). If the projection intersects the surfaces more than once, the point closest to the input point is created. If the projection does not intersect the surface then a point is not created.

Next a Point Naming dialog is presented to allow the point naming to be controlled. Select or enter a new group name. You can add a point name prefix and/or suffix to each projected point. Note the prefix and suffix are optional.



Point Naming Dialog

The figure below presents an example of the Polar projection function. The projected points lie on a line between the initial point and working frame origin. The point is created at the intersection of line and surface(s).



Points Projected along polar axes onto Surfaces

Projection of Points onto Objects

To create a point that is the projection of another point onto an Object, e.g., Planes, Circle, Lines, Spheres, Cylinders, Cones, etc. use the menu function Construct, Point, Projects Points to Objects. You are prompted to select the desired input points. Now you will be prompted to select the desired objects. A dialog showing the coordinates of the projected point in the Current Working Frame is present to allow you to control the point names. Note that for this projection method, the resulting point may not appear to lie on the object. The result may be off of the bounded graphical representation of the objects.

Intersection Point Construction Methods

The set of functions to create points at the intersection of objects follow a similar process.

Intersection of Three Planes: To create a point at the intersection of three planes select Construct, Point, Intersection, 3 Planes. You are initially prompted to select the three planes. Double click your left mouse button on the desired planes or select using the F2 function. A dialog showing you the coordinates of the point in the Current Working Frame is presented. Enter a point name and group for the computed point.

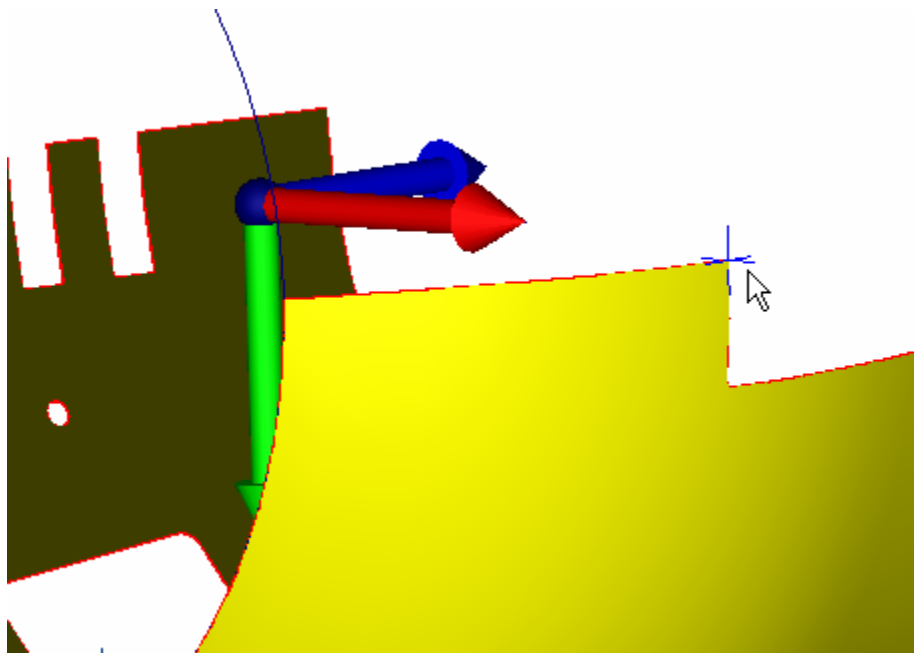
Intersection of Line and Plane: To create a point at the intersection of a line and plane select the menu option for Construct, Point, Intersection, Line and Plane. You are prompted to select the Line and then select the plane. A dialog showing the coordinates

of the intersection point in the Current Working Frame is presented. Enter a point name and group for the new point.

Intersection of 2 Lines (mutual perpendicular midpoint): To compute the point that lies at the intersection of 2 lines is done with the menu function Construct, Point, Intersection, 2 Lines. If the lines are not in the same plane or don't intersect the computed point will lie at the mid-point between the closest intersection of the lines. If the two lines are parallel the function will not create a point.

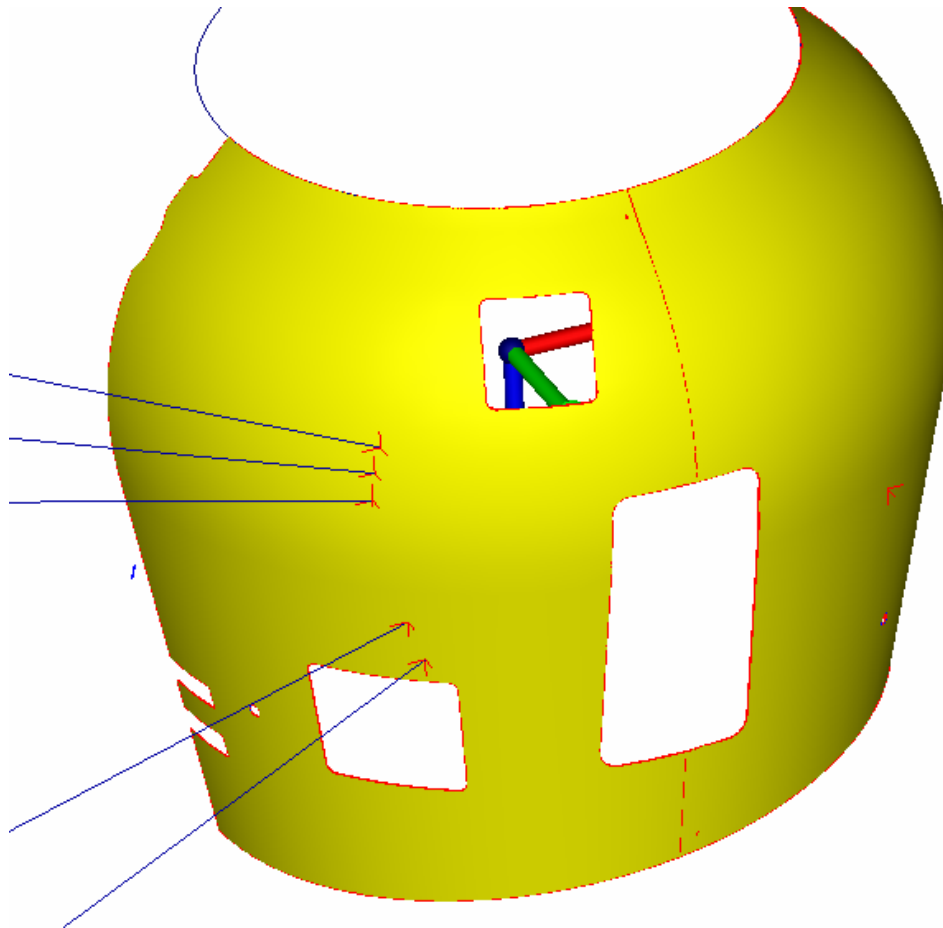
Intersection of 2 B-Splines: To compute the point that lies at the intersection of 2 B-Splines use the menu function Construct, Point, Intersection, 2 B-Splines. If the B-Splines don't intersect the computed point will lie at the mid-point between the two closest points on the respective B-Splines.

The figure below shows an example of a point created at the intersection of 2 B-Splines. In this case the point is at the corner of the part surface.



B-Spline Intersection Point Construction Method

Intersection at the Principle Object Axes Surface: To compute the intersection between objects and surfaces use the menu function Construct, Point, Intersection, Principle Object Axes Surface. All of the intersection points are created and put into a group that you define. An example of this function is shown below. In this example the lines were the principle objects and the points were created at the intersection of each of these objects and the selected surface(s).



Intersection Point at the Principle Object Axes Surface Method

B-Spline and Surfaces: To compute the intersection between B-Splines and surfaces use the menu function Construct, Point, Intersection, B-Splines and Surfaces. All of the intersection points are created and put into a group that you define. An example of this function is shown below. In this example the lines were the principle objects and the points were created at the intersection of each of these objects and the selected surface(s).

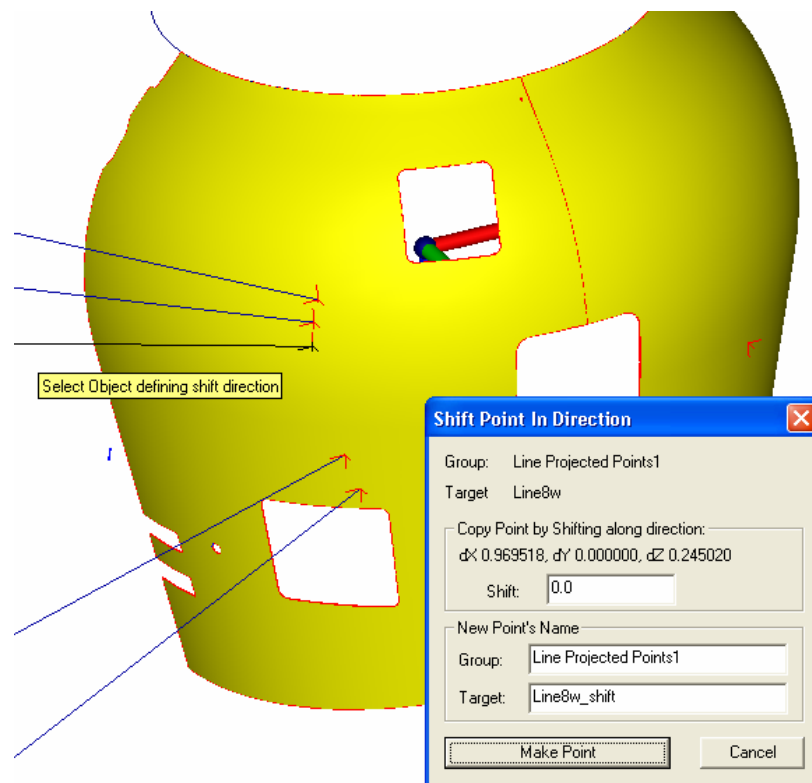
End of a Line

To create a point at the end of a line use the select Construct, Point, End of a Line function in the menu. You are prompted to select the line at close to the end that you want the point. Double click your left mouse button on the desired line or select using the F2 function. A dialog showing the coordinates of the end point in the Current Working Frame is presented to help you name the point and set its group name. Enter a point name for this new point and you're done.

Shift Point in a Direction

To shift a point in a particular direction the Construct >> Points >> Shift Point in a Direction function is helpful. Select the point to shift and then the object that defines the direction. Selecting a line will shift it along the line while a plane will shift it normal to

plane. The example below shows a point being shift down a line. After selecting the object (e.g., line) a dialog with inputs for the distance to shift and the point name and group is presented.



Shift a Point in a Direction Method

Fit to Points

To compute the centroid of a series of points use the Construct, Point, Fit to Points menu options. It prompts you to select the desired points. A dialog showing the computed coordinates of the mean point in the Current Working Frame is shown next. Enter a point name and group name for the computed point.

Hidden Point Bar

You may create a point using a hidden point bar. To do this select Construct >> Points >> Hidden point. You will be prompted to select Point A and Point B of the hidden point bar. You will be prompted to select the desired hidden point rod (in case you have several). After selecting the rod the new point will be added and you will be prompted for a point name.

Patch Normal Shift Construction Methods

The set of functions to create points by shifting along a patch normal follow a similar process. These functions work by computing a surface normal in a local area and then create points at an offset to the normal.

Single Point: This function starts by asking the user to select a point to offset. The next step asks for points that define the normal direction. After selecting these inputs a dialog to control the point name and the offset is presented. The figure below is an example of the input dialog.



Add Shifted Point along a Patch Normal dialog

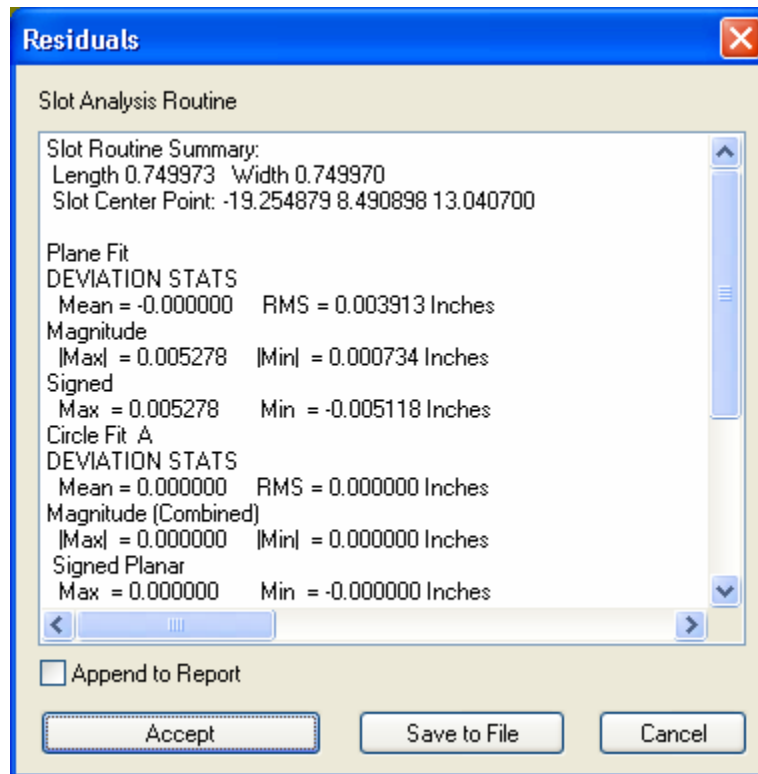
Enter the desired offset in the Target Offset field and any additional offset (e.g., material thickness) into the Additional Offset fields. The Group and Point fields left you name the point that will be created.

Hole Center: This function creates a point at the center of a hole at a specific offset from the surface. The function asks for two sets of points. The first set defines the circle perimeter and the second set defines the patch normal direction. The patch normal set of points can be the same set that defined the circle. The process computes a point at the center of the circle (from the first group of points) and at an offset along the plane normal (defined by the second set). The dialog to set the point and group name along with the offset amount is the same as the one used in the Single Point function (above).

When working with a hole in a surface as your starting point; one method to follow is to create the circle points is with the Construct >> Point >> Layout >> Object Vertices menu function. Select the B-Spline that defines the edge of the hole. If the B-Spline does not exist; create it with the Construct >> Curves >> B-Splines >> from Surfaces function.

Slot: Computing offset points from a slot involve defining the two circles for the slot and then the patch points. The circles are solved for by selecting points at their perimeter. The process asks first for points at the perimeter of CircleA and then for the points for CircleB. The next step asks for the points to define the normal direction. A plane is solved for from these points and then the planes normal direction is used as the direction to offset the created offset slot points. A dialog allowing the user to set the point name and group along with the offset amount is presented. The created points include the centers of each slot circle offset by the amount defined in the dialog.

The computation presents a Residuals dialog to the user for review before the points are created. The residuals dialog shows a Slot Routine Summary with the slot length, width, and center point. Plane and two circle fit statistics are also posted in the Residuals dialog for review. If the computed characteristics don't match expected results the user can cancel the operation. If the results are within expected bounds then they are accepted with the Accept button.



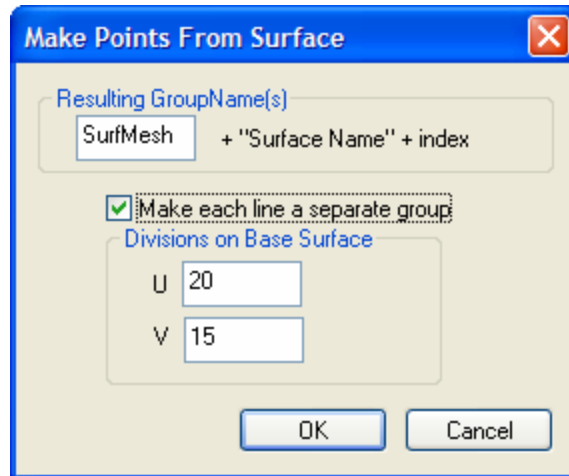
Slot Analysis Residuals and Summary dialog

Plane Section Crossing

Given a group of points and a plane the Plane Section Crossing function first computes a best-fit line from the group of points then solves for the point at the intersection of the line and the plane. The point is automatically named and put into a new group.

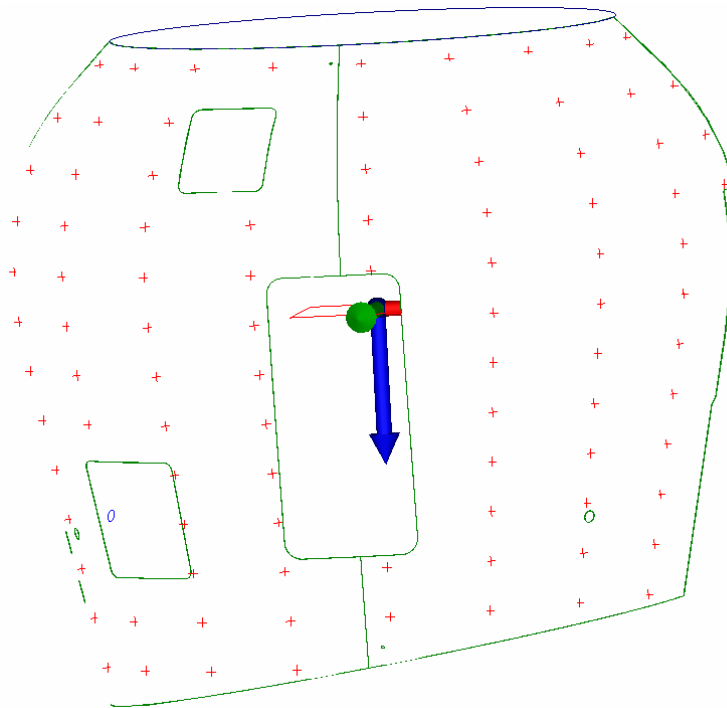
From a Surface Mesh

This option allows you to create points that lie on the vertices of a surface mesh. This function is useful for simulating data collection of a complex part. After selecting the surface(s) to create points on a dialog that defines the group naming and mesh density is presented (shown below).



Surface Mesh Point Creation dialog

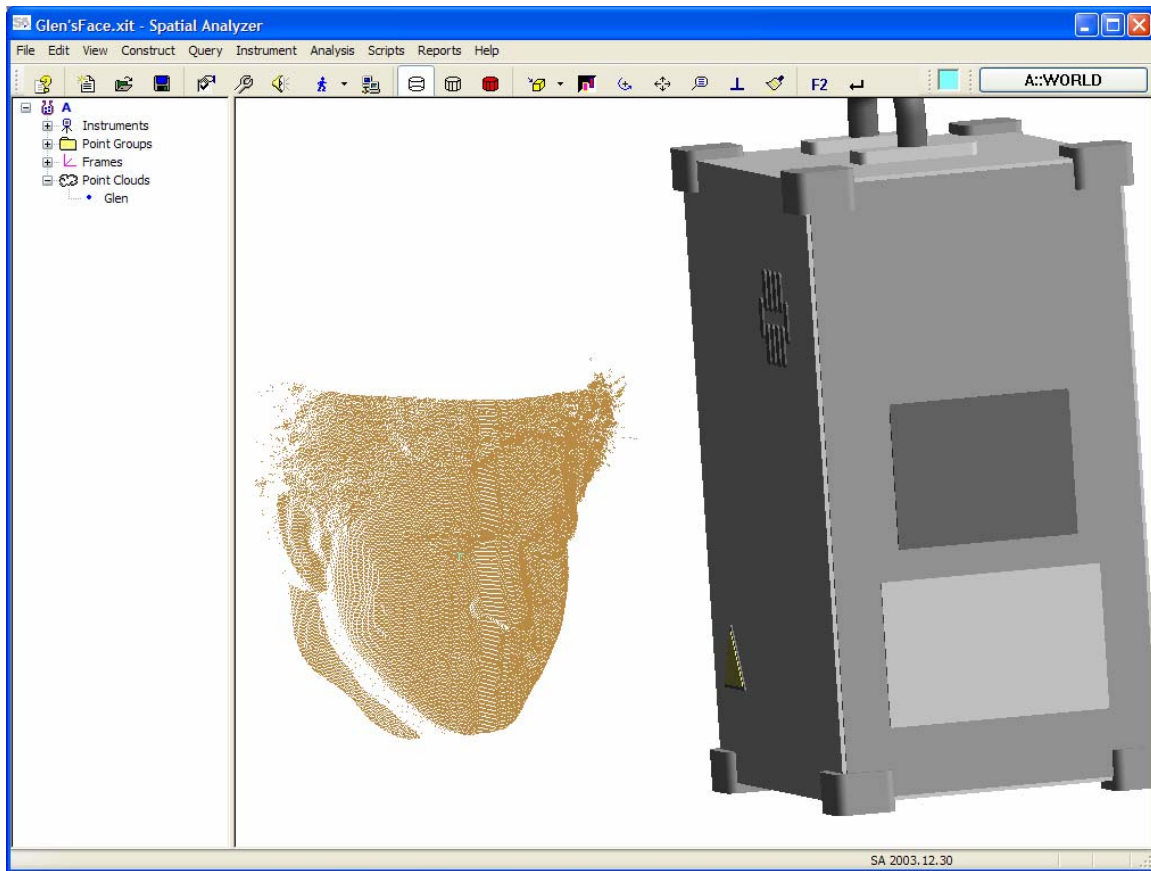
The points are created in a U-V based grid on the surface. By selecting the Make each line a separate group the points will be separated into groups along the U direction. The figure below shows an example of a grid network.



Surface Mesh Points

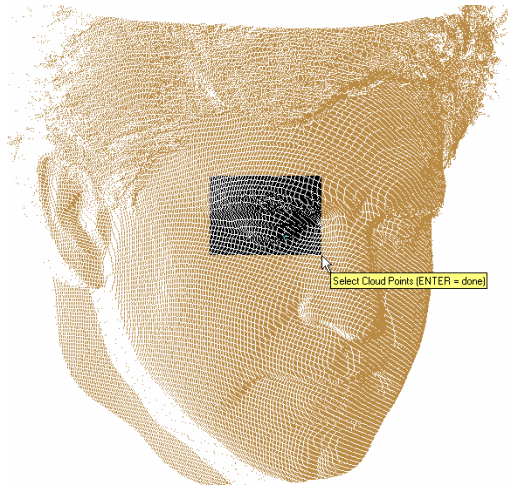
From Cloud Points

In 3D scanning applications there can be millions of points, to help you manage data sets of this type SA uses a special object type called point clouds. These objects make the process of manage scanner clouds of points much easier.

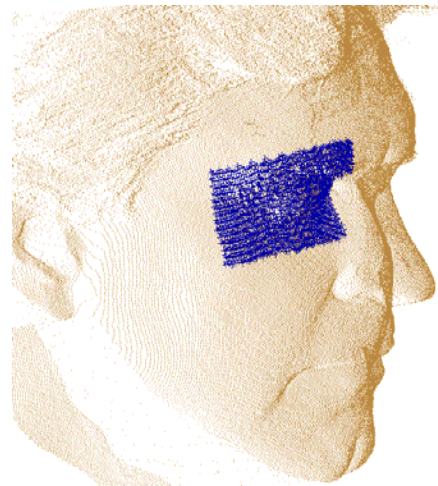


Point Cloud from Scanner in SA

Cloud points need to be converted to points in groups for a number of processes in SA (e.g., fitting to surfaces). The From Cloud Point function is one means of accomplishing that task. The process allows you to select cloud points to convert. The figure below shows a selected area.



Selected Cloud Points to convert to Points



Points in a group from Cloud Points

After selecting the area(s) of interest hit the Enter key. A dialog allowing you to name the point group and the point name prefix is the last step. The figures above show an example of the From Cloud Points function.

From Existing Points

The functions under the From Existing Points submenu are helpful with measurement jobs with a large number of groups. This group of functions lets you operate on points from different groups i.e., without regards to their group boundaries. For example 10 points from GroupA and 6 from GroupB can be moved in one operation without affecting other points in either group.

From Existing Points >> Copy: The copy function supports selection from any/all groups and then puts them into a new common group. When collecting data, sets of points are frequently put into separate groups. Using the copy From Existing Points function is a quick means of re-grouping a series of point sets. Subsequent operations on the points will then only need to be applied to this new grouping of the points.

The copy process has you select points from any of the groups. All of the selection methods are available including graphically double left-clicking points, trapping with the shift button down and dragging with left mouse button, F2 selection tool, and selecting on the tree view by double clicking on the points. After selecting the points to copy; hit the enter key to end the selection process. A dialog allows you to name to “New” group. After naming the group copies of the selected points appear on the tree view under the “New” group.

As discussed previously copied targets do not copy their measurements so the copies of targets are points without measurements.

From Existing Points >> Move: Moving a set of points from a series of groups into a new group can be accomplished with the Move From Existing Points function. The process follows the same one used in the copy function. The results are different because the

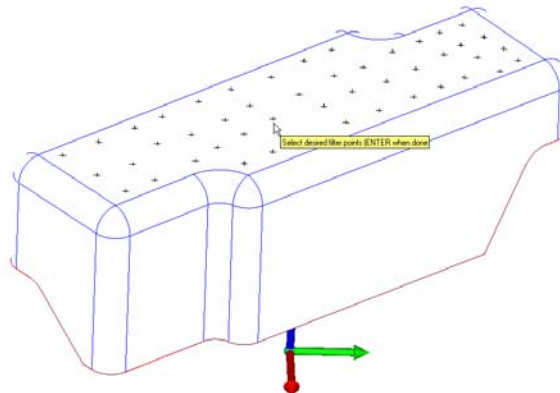
points are removed from the source group(s). This function also does not preserve measurements during the move, i.e., targets are converted to points without measurements in the “New” group.

From Existing Points >> Subset with Greatest Spacing: Sampling a set of points down to those points with the greatest point to point spacing is supported with the Subset with Greatest Spacing function. The first step is to select the points to filter. Hit the enter key when your selection is complete. A dialog then asks you how many points from the selected points are needed in the new group. After computing the filtered group of points a dialog asking you to name the “Spaced Points” is presented. The points in the subset group are copies from source group(s).

Filter Clouds to Points

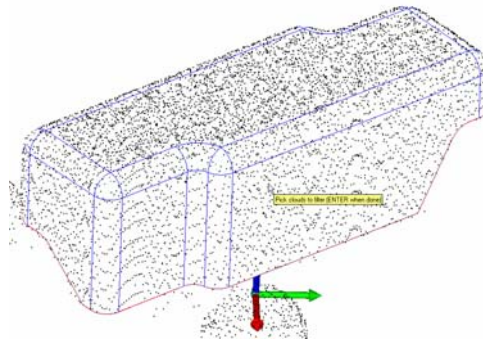
Frequently point cloud data needs to be filtered down to points that relate to specific features (e.g., plane, circle/hole, and surface.) One means of accomplishing feature base point extraction is the Filter Clouds to Points function.

The first step has you select a set of points that will be used as the filter points for the filtering process. For extracting a plane from point clouds select a series of point's through-out on the plane. The figure below shows a series of points on top of the CAD model of the part.



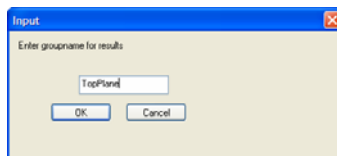
Filter Seed Points to extract Top Plane points from cloud(s) of points

Next select the point cloud(s) to filter points from. The figure below shows a point cloud of 314,000 points. Using the filter points selected in the first step the cloud of 314,000 points will be filtered down to a point group that defines the top plane.

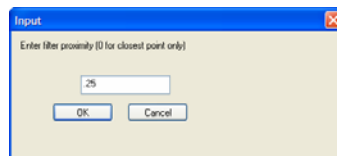


Cloud(s) of points to be filtered

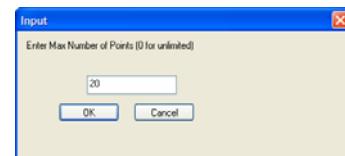
The next three inputs are for the name of the filtered point group, the proximity distance around each filter point, and the maximum number of points to extract around each filter point. The figures below provide an example for these inputs.



**Group name dialog for
Filtered Points**

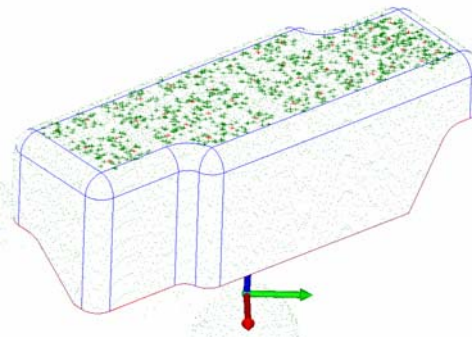


Proximity distance dialog



**Max number of points to
filter from within
proximity distance**

Cloud points that are within the radius defined by the proximity distance around each filter point are automatically selected and then converted to regular points. The figure below shows the results.



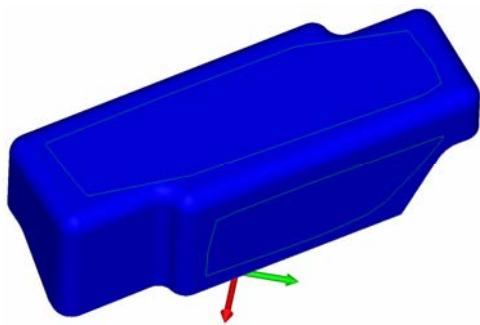
Filtered points from the cloud

Options:

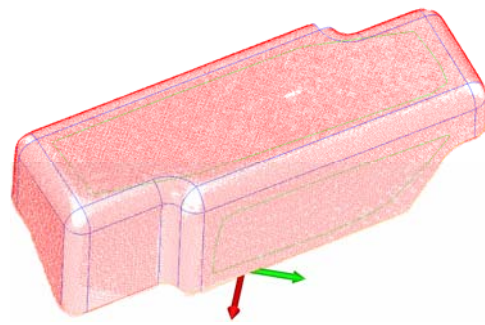
- If you only want the closest points around each filter point then set the proximity quantity to zero. The routine will then search the clouds for only the closest points.
- If you want the 5 closest points then set the proximity to zero and the Max Number of Points quantity to 5.
- If you want all of the points within a specific radius set the proximity to the radius quantity and leave the Max Number of Points set to 0.

Filter Clouds to Plane

Extracting planar points from a cloud is a common process when working with 3D scanner data. The Filter Clouds to Plane function is one method to extract planar points from a cloud.

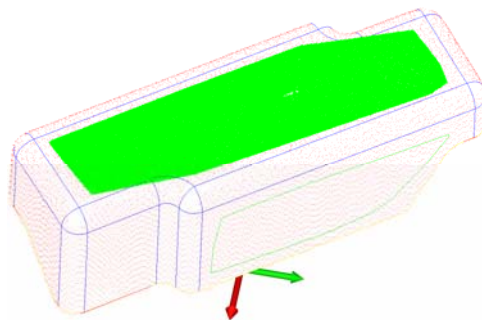


Model with plane used to Filter Cloud Points



Cloud of (~230,000) points to filter

This function asks for a filter plane and then the clouds to extract the planar points from. A proximity quantity is used to specify the maximum distance cloud points can be from the plane.



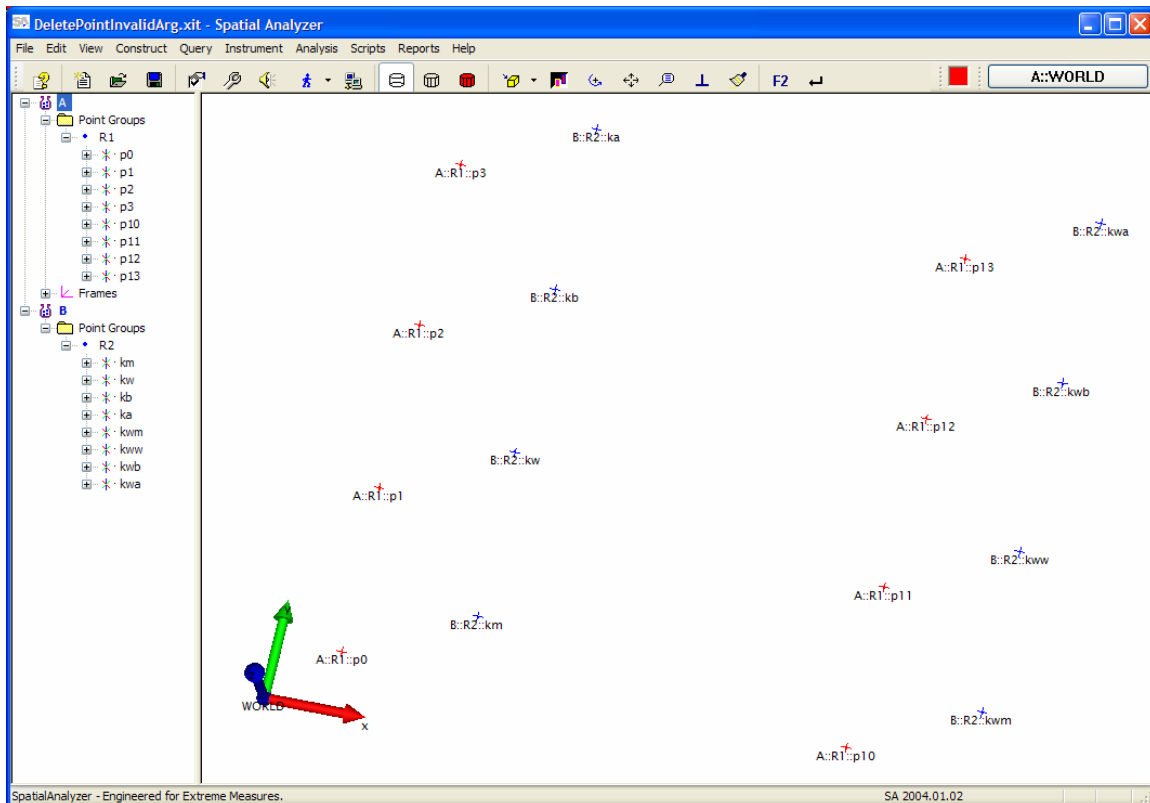
Filtered (~69,000) points to plane boundaries and proximity tolerance

The figure above shows the filter plane and then the cloud of points from which the planar points will be filtered from.

Auto-Correspond Two Groups

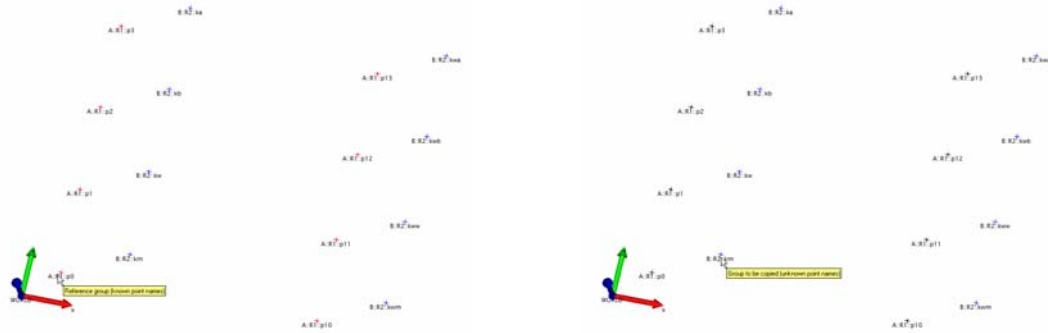
Getting points within two groups to have consistent names is an important step for transforming and reporting tasks. You can use the Auto-Correspond Two Groups function to rename the points within a group to match each respective point in a reference group. The function relies on unique inter-point distances to identify point name correspondence.

The figure below shows two sets of points that clearly correspond but do not have matching point names. The first step in the process is to choose the reference point group. In this case the group R1 in collection A is used as the reference group.



Point groups with names that do not correlate

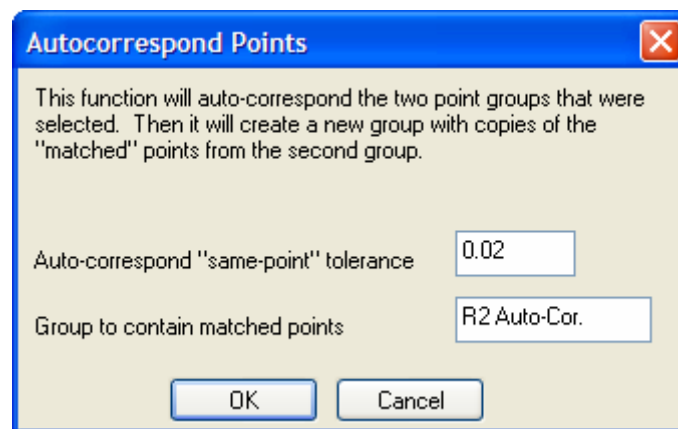
The figures below show the selection process for selecting the reference group and then the group that will be copied and then have its points renamed to match the references.



**Select the group that will serve as set of
Reference point names**

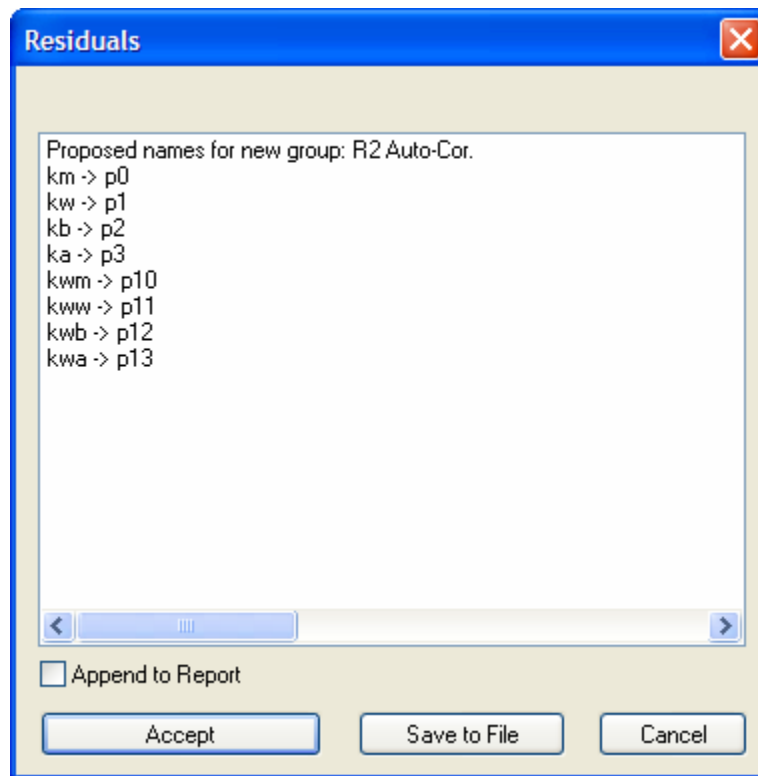
**Select the group to be copied and
renamed**

After selecting the two groups an input dialog that asks for the Auto-correspond same-point tolerance and the new group name is presented. The tolerance is used to isolate unique inter-point distances. Selection of the smallest tolerance that will uniquely set point to point distances is suggested. For most applications using roughly twice an instruments precision is a good initial estimate for the tolerance.



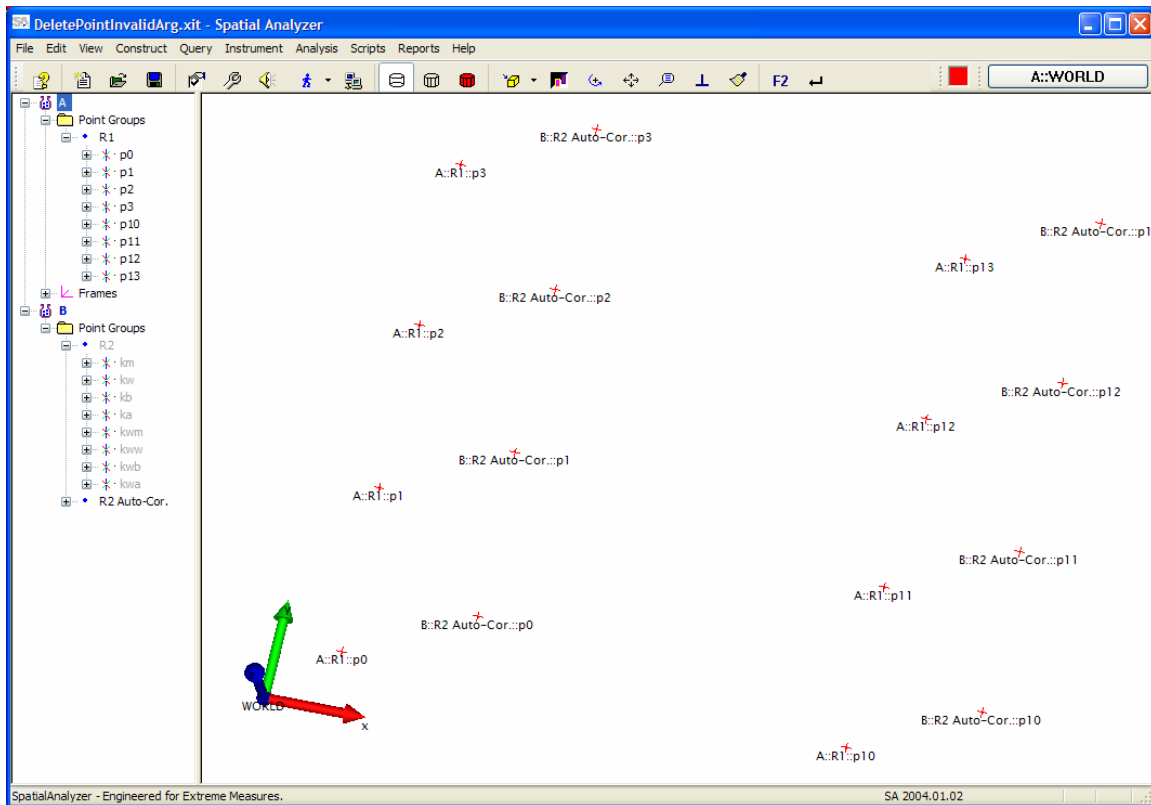
Auto-Correlation Input Parameter Dialog

A residuals dialog with the proposed renaming map shows each point. For those that a match was found a corresponding point is shown. If a match was found the original point name is retained in the new group. The figure below shows the proposed names for the points in the new group.



Auto-Correlation Residuals dialog showing proposed Renaming Map

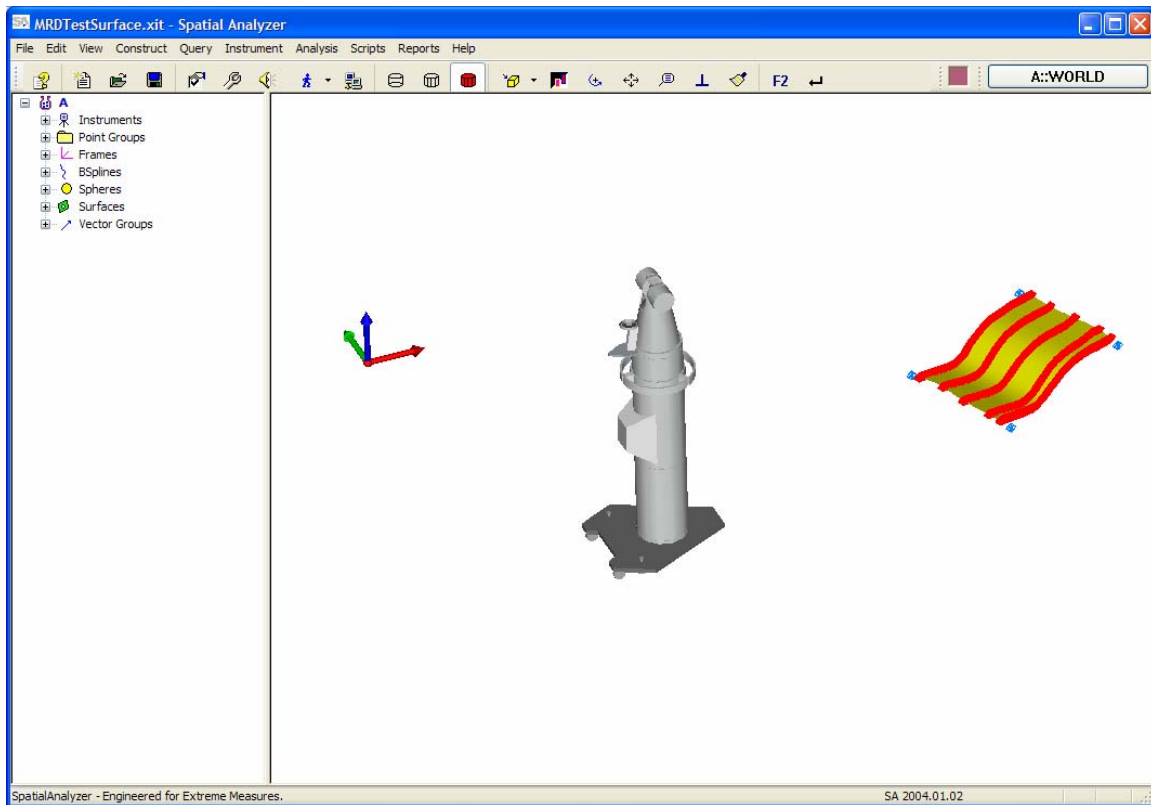
The figure below shows the results after accepting the renamed group. The R2 Auto-Cor group contains copies of the points in the R2 group however the point names now correspond to those in the R1 reference group.



Results from Auto-Correlation of Point Groups

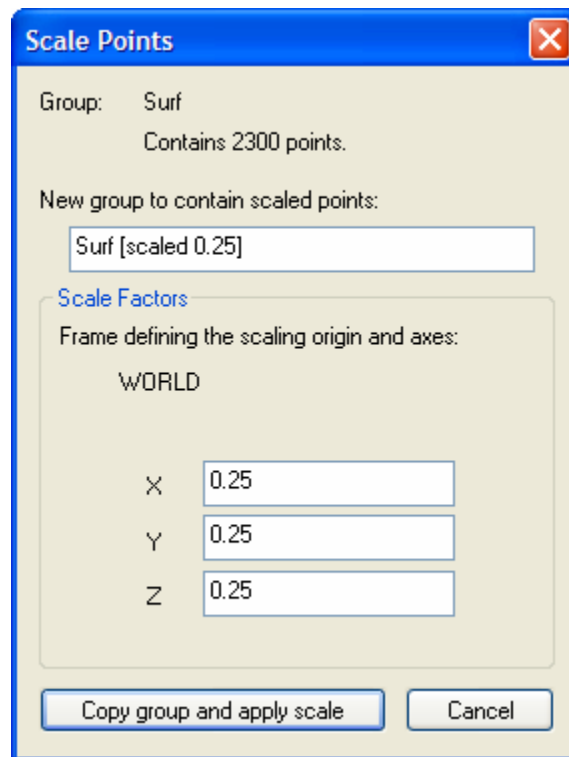
Scale Points

The Scale Points function is able to scale a group of points either consistently in all three axes or differently in each X, Y, and/or Z direction. As an example if the points for the surface show in the figure below need to be scaled down by a factor of 4, you can use the Scale Points function and a scale factor of 0.25.



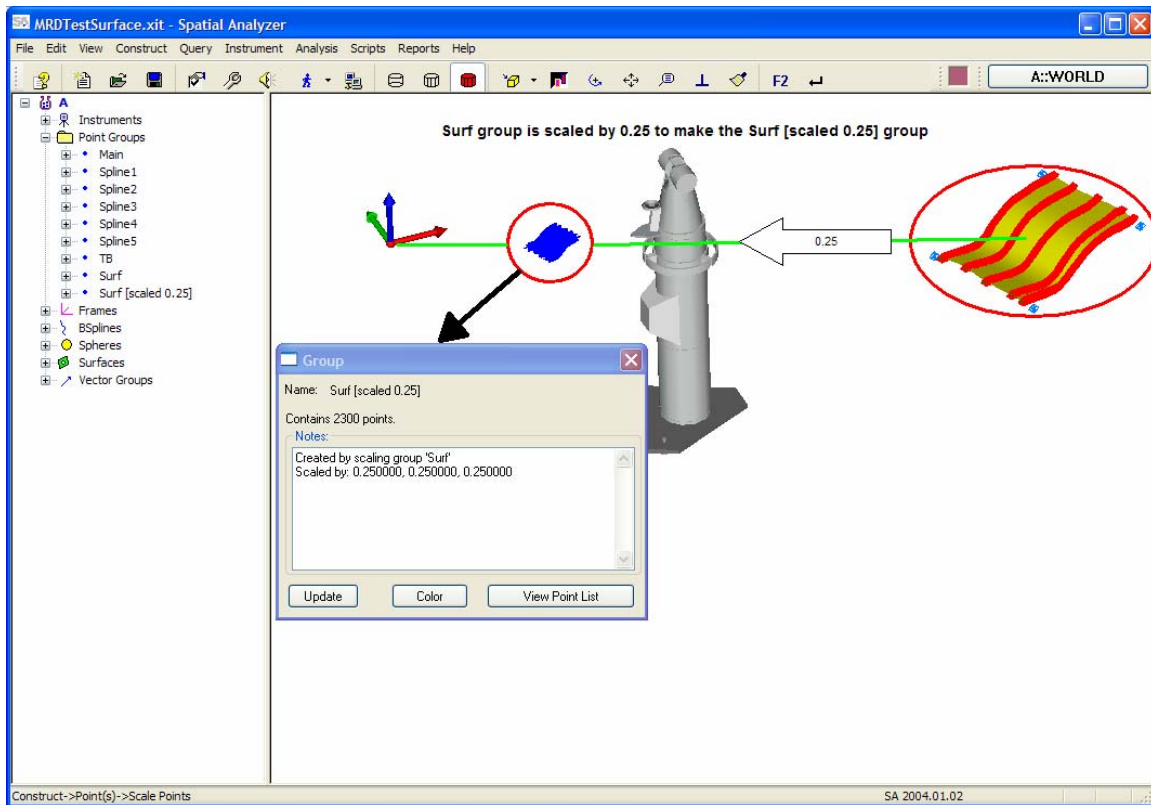
Point group from a model that needs to be scale down by a factor of 4

The first step in Scale Point's process is selecting the frame to scale relative too. The next step is selecting the point group too scale. With the frame and point group known the function now needs to know by how much you need to scale the X, Y, and Z components. A dialog asking for those parameters along with a field for the new group name is presented. For this example a scale factor of 0.25 is used in the X, Y, and Z component directions. Note the component values for each direction can be different. That feature can be particularly useful when compensating for non-isotropic material expansion. The figure below shows the dialog for the example case.



Scale Points Input Dialog

After selecting the 'Copy group and apply scale' button the points are copied and adjusted relative to the selected frame. For the example the results are those shown in the figure below.

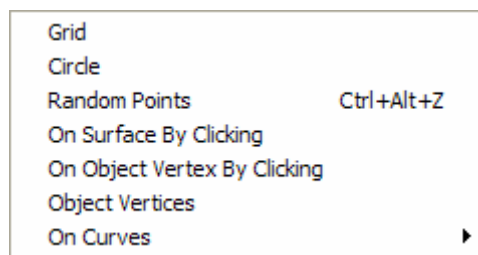


Results from Scale Points Function

Once the group is created the details of the process used to create it is saved in the group's Notes field.

Layout

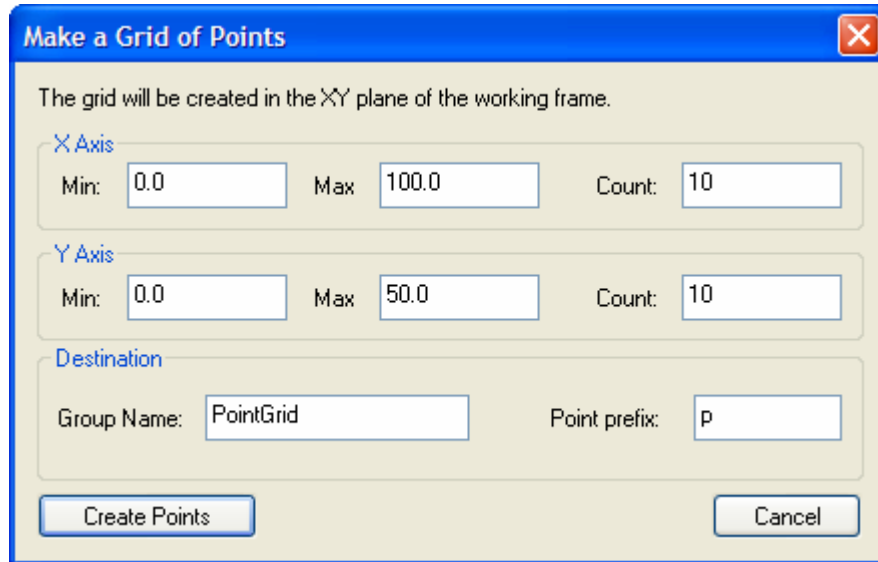
A series of point layout functions are grouped under the Layout submenu. These functions enable you create points in unique ways ... most of these layout processes are the results of special requests from metrology system users.



Layout Point Function Submenu

Grid: This option allows you to create a 2D grid of points in a group. This function is useful for simulating data collected at regularly spaced intervals.

The grid is always created in the XY plane of the working frame. Once created the group of point can be moved with one of the Move Object functions. An alternate technique for positioning the grid in 3D space is to first create a frame with its X-Y plane origin at the location for the initial point of the grid. Then before using the Points >> Layout >> Grid function make the frame the Working frame. The input dialog for the Grid function is shown below.



The dialog box is titled "Make a Grid of Points". It contains the following fields and controls:

- Instruction: "The grid will be created in the XY plane of the working frame."
- X Axis**
 - Min: 0.0
 - Max: 100.0
 - Count: 10
- Y Axis**
 - Min: 0.0
 - Max: 50.0
 - Count: 10
- Destination**
 - Group Name: PointGrid
 - Point prefix: p
- Buttons: "Create Points" and "Cancel"

Make a Grid of Points Dialog

The default grid is created with 100 points, 10 rows in the X and 10 in the Y direction. The axis offsets can be set with the Min and Max fields and then the number of points set in the Count Field. If you want a regular spacing of say 10 inches over a span of 100 inches in the X direction, specify that 11 points (not 10) be created. The distance between points is always distance divided by the number of points minus one.

Circle: Creating a circle or arc of points is a function frequently used when simulating measurement processes. The points on a circle or arc are created in the X-Y plane of the current Working Frame. The default input dialog for creating a circle of points is shown below.

Layout Points on Circle [X]

This function will layout a points in a circle in the XY plane of the working frame.

Radius:

Starting angle (deg., relative to X axis):

Number of points on circle:

☒ Evenly space points around entire circle
☐ Space points by deg.

Point Naming

Group:

First Point:

☐ Create center point

OK Cancel

Default Input Dialog for Laying out Points on a Circle

Inputs for the circle radius, starting angle, and the number of points on the circle are the initial fields to specify. The starting angle is based relative to the X axis. Therefore if you specific starting at an angle of 10 degrees and then space 20 points at an angular interval of 2 degrees the function will create 20 points with the first one at 10 degrees, the second at 12 degrees, then 14 and so on until 38 degrees. The results of this example are shown below.

Layout Points on Circle [X]

This function will layout a points in a circle in the XY plane of the working frame.

Radius:

Starting angle (deg., relative to X axis):

Number of points on circle:

☐ Evenly space points around entire circle
☒ Space points by deg.

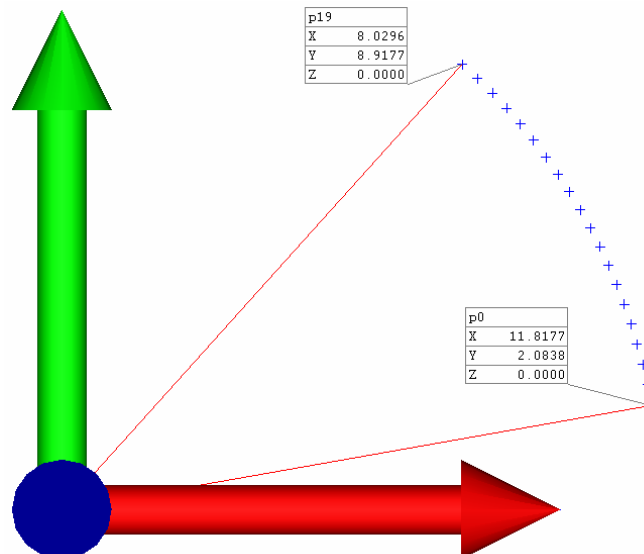
Point Naming

Group:

First Point:

☐ Create center point

OK Cancel

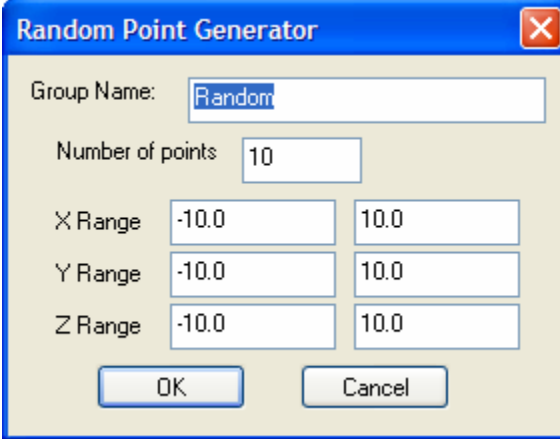


Layout Points on a Circle Dialog

Arc of Points

To specify “point spacing” from a start angle to another angle set the number of points on the circle to the number of spaces plus one. Referring back to the example above; to get the last point at 40 degrees from the start we need to set the number of points on the circle to 21 instead of 20.

Random Points: Creating a set of random 3D points in SA is accomplished with the Random Points function under the Points >> Layout submenu. The 3D space X, Y, and Z ranges are defined relative to the current Working Frame. The default input dialog is shown below.

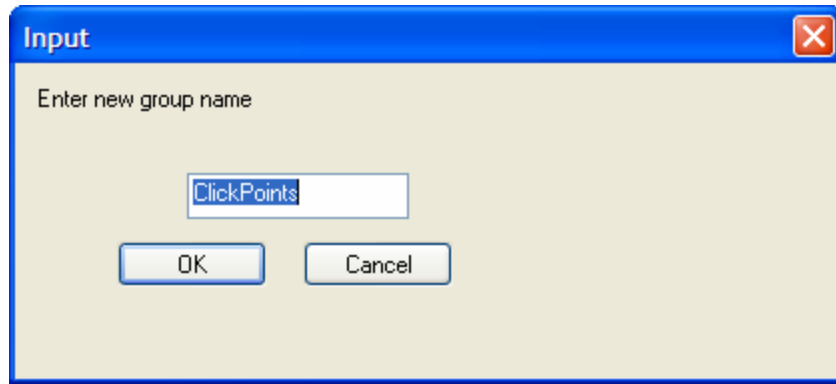
The image shows a dialog box titled "Random Point Generator" with a blue title bar and a red close button. Inside the dialog, there is a "Group Name:" label followed by a text box containing the word "Random". Below this is a "Number of points" label followed by a text box containing the number "10". Underneath are three rows of range inputs: "X Range" with two text boxes containing "-10.0" and "10.0", "Y Range" with two text boxes containing "-10.0" and "10.0", and "Z Range" with two text boxes containing "-10.0" and "10.0". At the bottom of the dialog are two buttons: "OK" and "Cancel".

Create Random 3D Points Dialog

Set the group name and the total number of random 3D points to create. The next set of inputs allows you to specify the spacing over which to randomly select points. The X, Y, and Z component are each randomly chosen so the space does not have to be a cube, or even a 3D volume.

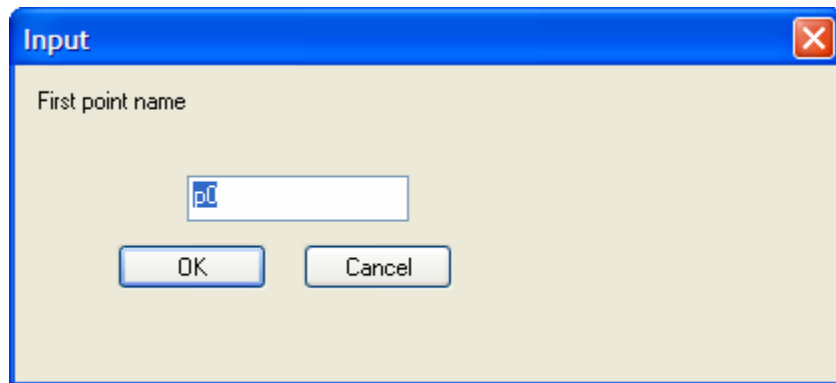
To create random points on a plane set one of the 3 components range values to the same number (e.g., Z Range = -10 and -10.) This results in a plane of randomly spaced points all with Z components of -10. A line of randomly spaced points is created by setting two of the three component ranges to the same number.

On Surface by Clicking: This option allows you click on a surface to create a point. SA projects the clicked location onto the surface and then creates a point. This function is setup by first entering the name of the group to put the points in (shown below.)



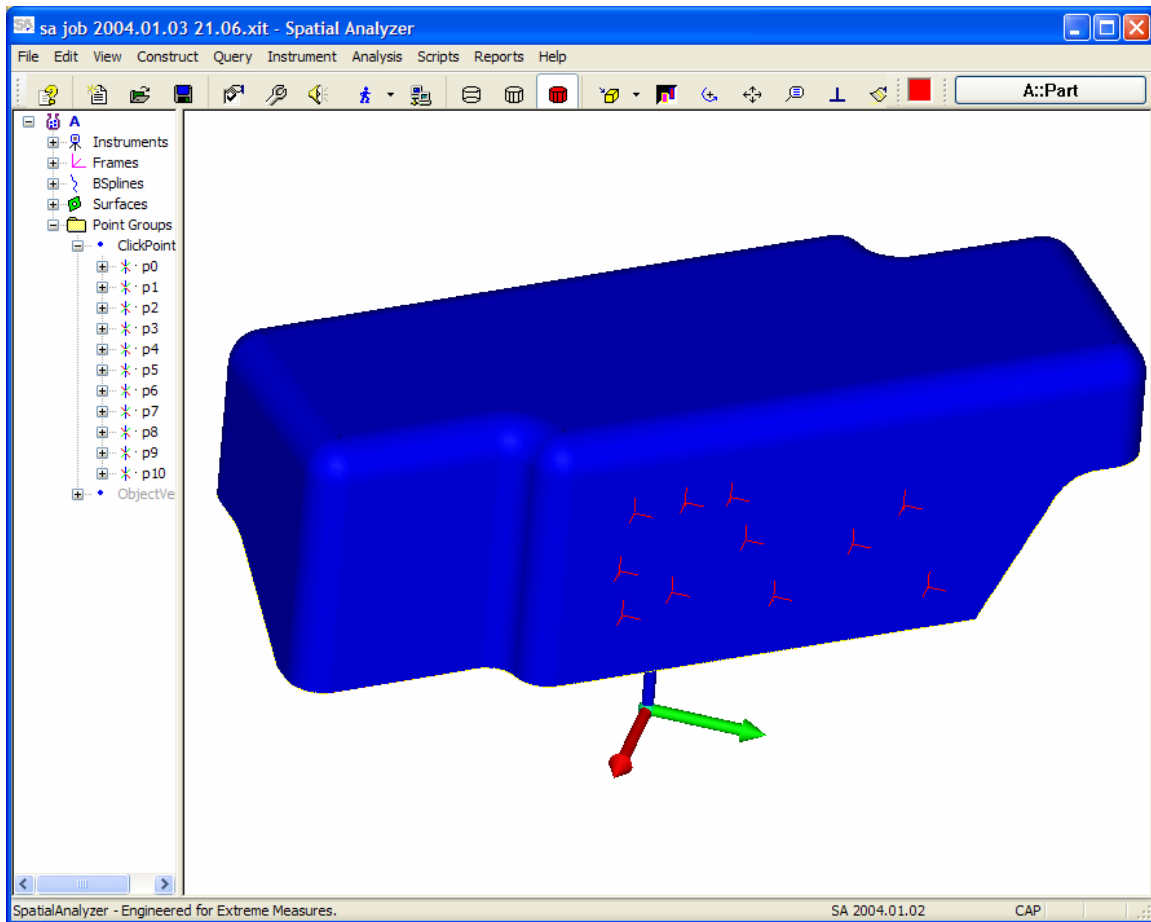
Group Name Dialog

The next step is to identify the name of the first point. Subsequent points are automatically named based on the first point's name. The point name input dialog is shown below.



Point Naming Dialog

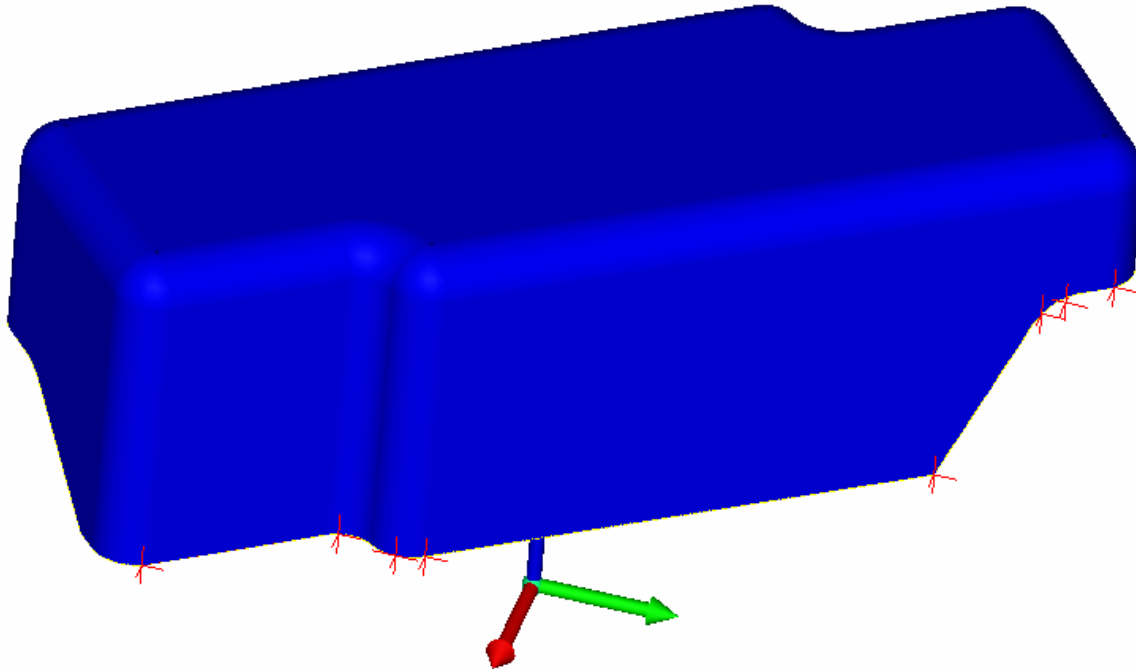
With the group and point name fixed the next step is to click on the surface. Each time you do a point is projected onto the surface and put in the group. The figure below shows an example.



Results from Construct >> Points >> Layout >> On Surface by Clicking

To end the process hit the enter key.

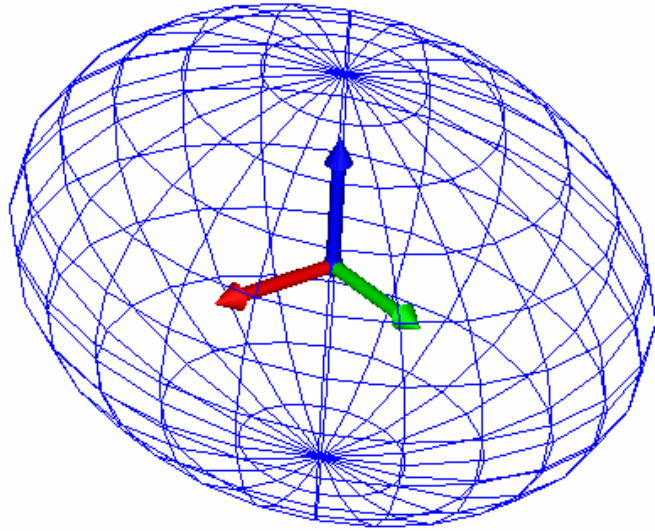
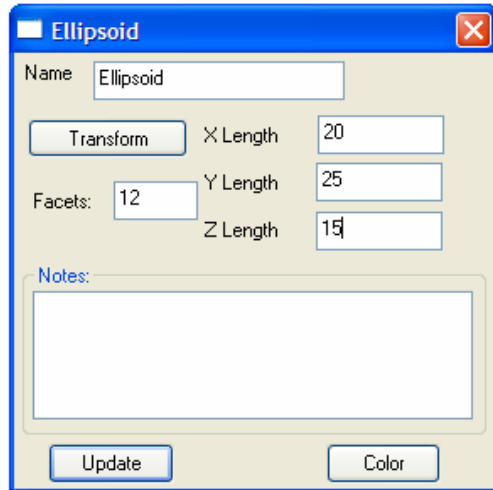
On Object Vertex by Clicking: This option allows you to create a group of points that lie on the vertices of an object. This function is useful for simulating data collection of a complex part. The process follows the same one used for clicking on surfaces. The difference for this function is the projection function finds the closest object vertex and creates a point. An example of the results is shown below.



Results from Construct >> Points >> Layout >> On Object Vertices by Clicking

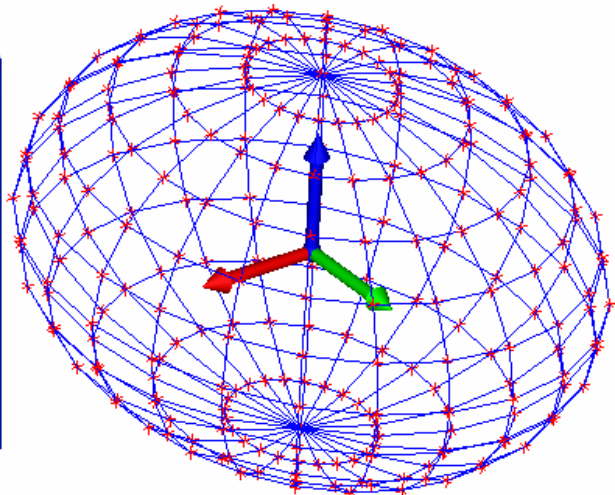
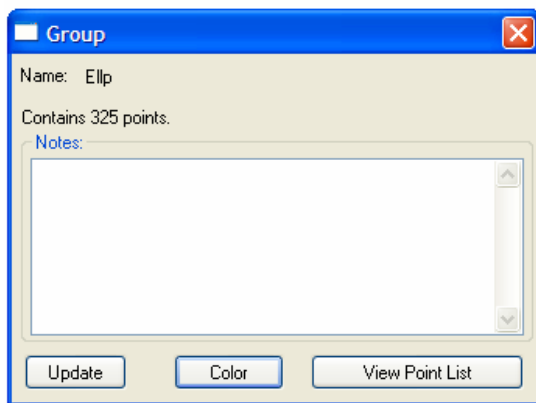
On Object Vertices: SA brings a number of functions to help layout points for a job. The function Construct >> Points >> Layout >> On Object Vertices is able to automatically create a whole series of points in critical areas. These points are then typically used as drive points for measurement processes or in simulations efforts where the measurements of these points are critical to a job's success.

With this function you select an object plus a point group name and points at each vertex are created. Each geometric object in SA supports constructing points on their vertices, making this a general function for creating points to simulate geometry. The example below uses an Ellipsoid object created in SA with the Construct >> Ellipsoid >> Enter function.



Results from Construct >> Points >> Layout >> On Object Vertices by Clicking

The figure below shows the results of the Construct >> Points >> Layout >> On Object Vertices function. 325 points were created at each object vertex. A change to the facet property will yield a different set of points when you use this function.



Results from Construct >> Points >> Layout >> On Object Vertices by Clicking

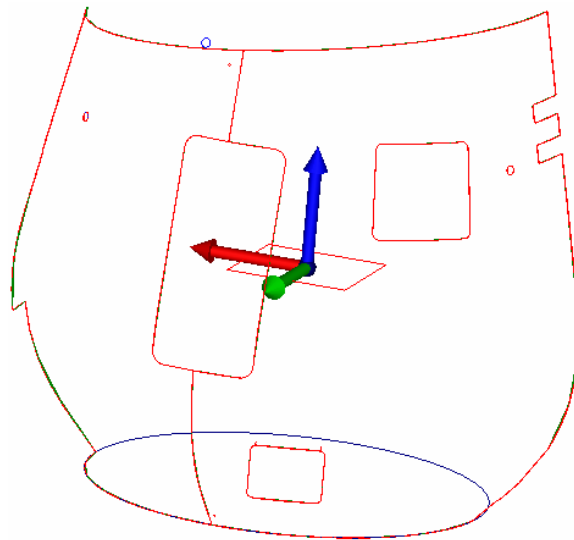
On Curves: This option helps you to create points that lie on curves at locations you choose, at specific intervals along the curve, and will create a certain number of points along a curve. Each function offers a slightly different mechanism to create points on curves and together they bring solutions for your metrology applications needs. This is another function that is useful for simulating data collection of a complex part.

The figure below shows the Layout >> On Curves submenu from Construct >> Points

n Spaced Points
Spaced at distance
By Clicking

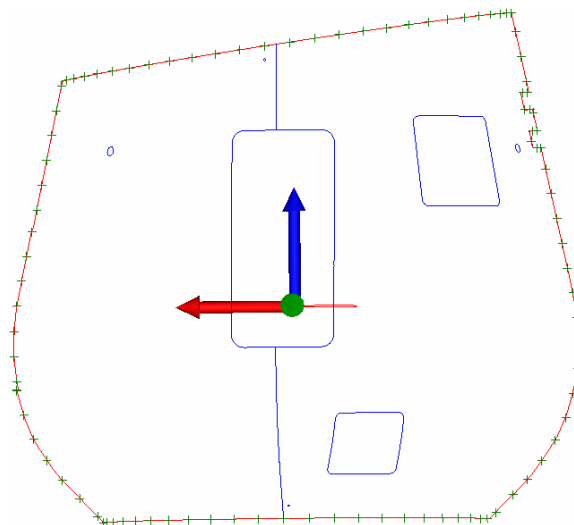
On Curves Point Layout Submenu

For this set of functions the model shown in the figure below helps illustrate the output point group for each function.



On Curves Point Layout Example

On Curves >> n Spaced Points: Putting a specific number of points on a curve at regularly spaced intervals is accomplished with the On Curves >> n Spaced Points function. Inputs for the function include selecting the curve, point group name and the number of points to put on the curve. The figure below shows the results when constructing 100 points on a curve.

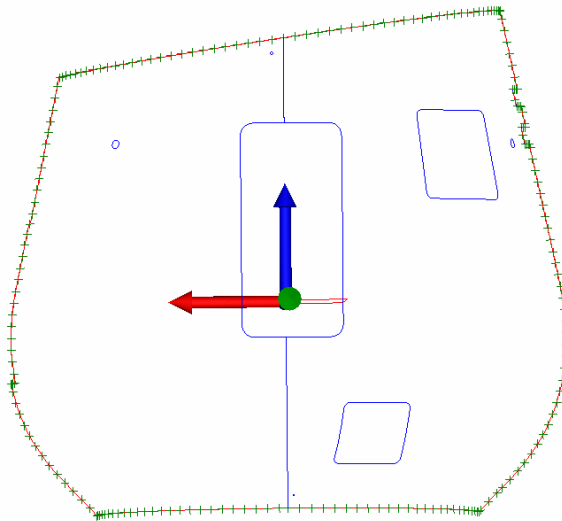


On Curves >> 100 Spaced Points

With this function the distance between points along the curve is equal and evenly divided between the 100 points (e.g., 99 spaces.)

On Curves >> Spaced at a distance: This option allows you to create a group of points where each point is at a set distance along the curve. This function is useful for simulating data collection of a complex part. Inputs include selecting the curve, defining the new group name, and setting the distance along the curve. Job units are used for the distance units along the curve.

The figure below shows the results produced with 1 inch point to point spacing.

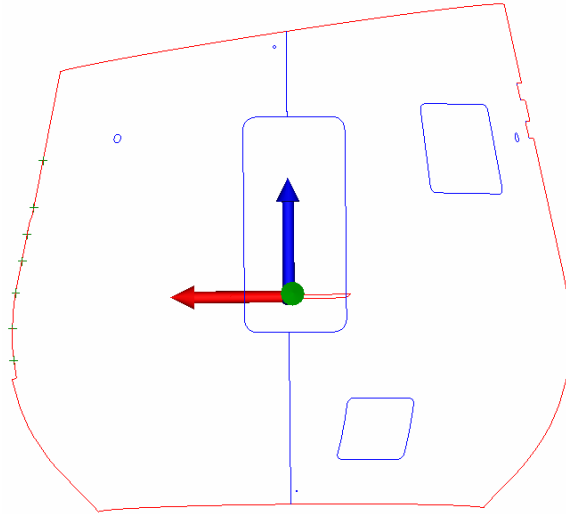


On Curves >> Spaced at 1-inch

This example produced 211 points for the 1 inch spacing. The points are on the curve and the distance between the points is 1 inch when measured along the curve. Using this point count we can say that the curve is at least 210 inches long but not longer than 211 inches.

On Curves >> By Clicking: Constructing points on a curve by clicking is a function that is helpful. After selecting the curve(s) to put the points on you are asked to name the group and first point. With these inputs now you can click anywhere close to the curves and the closest point to the curves is constructed and put into the group.

The figure below shows the results for several points.



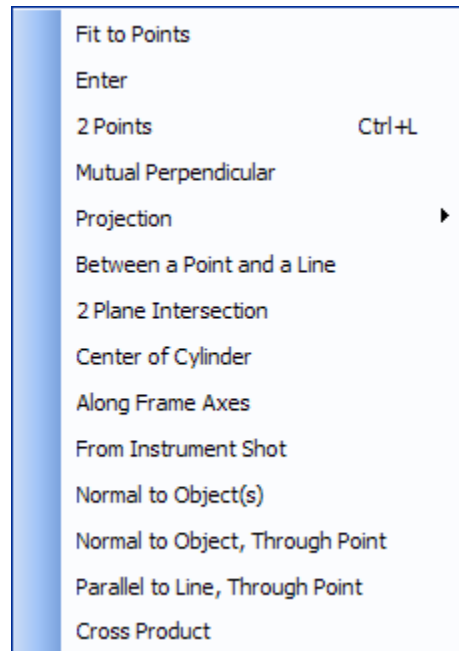
On Curves >> Spaced at 1-inch

Additionally, there are several Point import functions available within the Spatial Analyzer. An IGES file may be imported from another CAD system, or, a point data file, like a GSI V-STARS file, may be imported.

Lines

Lines are defined by a set of two points, End A and End B. One can map the description of a line as a single point with a direction and a distance as well. Lines do have direction and this direction may be indicated using the optional arrow designation.

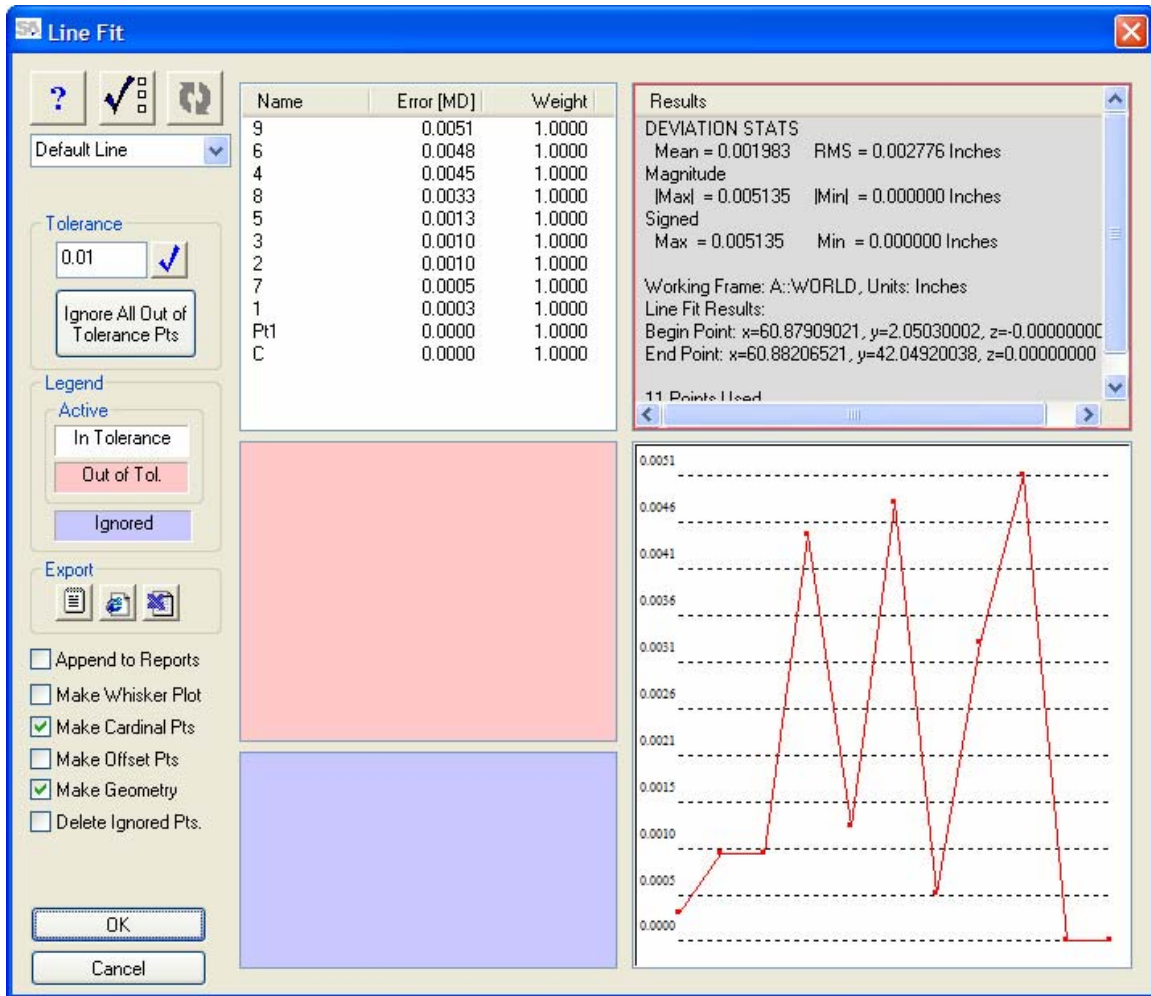
Graphically, the Spatial Analyzer treats all lines as line-segments. You may, however, use the directional information (vector) of the line for various construction options. Computed intersections will always use the mathematical infinite line to form an intersection. You may construct lines many ways.



Construct Line Menu Options

Fit to Points

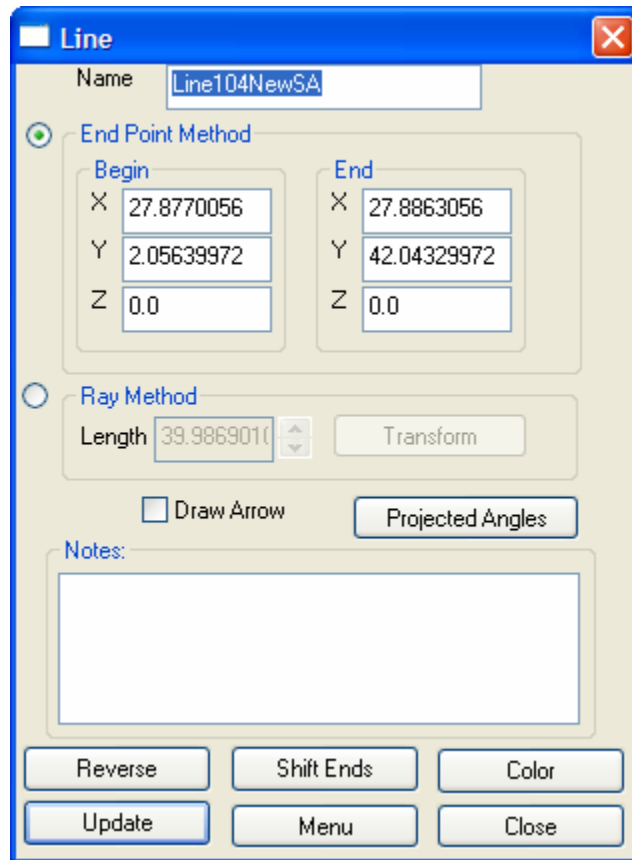
After selecting the points to fit a line to, **SpatialAnalyzer** will determine the line parameters that best-fit the selected points. The results of the fit are displayed, and you will have the chance to modify the results in the Line Fit dialog (shown below).



Line Fit Results Dialog

Enter

This will allow you to enter the end-points of the line or designate a direction and a length. These values are entered using the Line Property dialog.



The image shows a 'Line' property dialog box. At the top, there's a title bar with a close button. Below it, a 'Name' field contains 'Line104NewSA'. There are two radio buttons: 'End Point Method' (selected) and 'Ray Method'. Under 'End Point Method', there are two columns: 'Begin' and 'End'. Each column has three input fields for X, Y, and Z coordinates. The 'Begin' values are X: 27.8770056, Y: 2.05639972, Z: 0.0. The 'End' values are X: 27.8863056, Y: 42.04329972, Z: 0.0. Under 'Ray Method', there is a 'Length' field with the value 39.9869011 and a 'Transform' button. Below these, there is a 'Draw Arrow' checkbox (unchecked) and a 'Projected Angles' button. At the bottom, there is a 'Notes' text area. At the very bottom, there are six buttons: 'Reverse', 'Shift Ends', 'Color', 'Update', 'Menu', and 'Close'.

Line Property Dialog

2 Points Ctrl+L

You can construct a line between two points in the model using this option. You will be prompted to select each point and then the line will be created. The property dialog will appear next allowing you to assign a name to the line and change and other properties as needed.

Mutual Perpendicular

This function will prompt you to select two lines. It will then determine the line that is the mutual perpendicular to the selected lines and create it.

Projection >> Line to >> Plane // Circle

After selecting either a plane or a circle as the destination, you will be prompted to select the line you wish to project. A new line will be created on the projection surface.

Between a Point and a Line

This function creates a line between a user-selected point and a line.

2 Plane Intersection

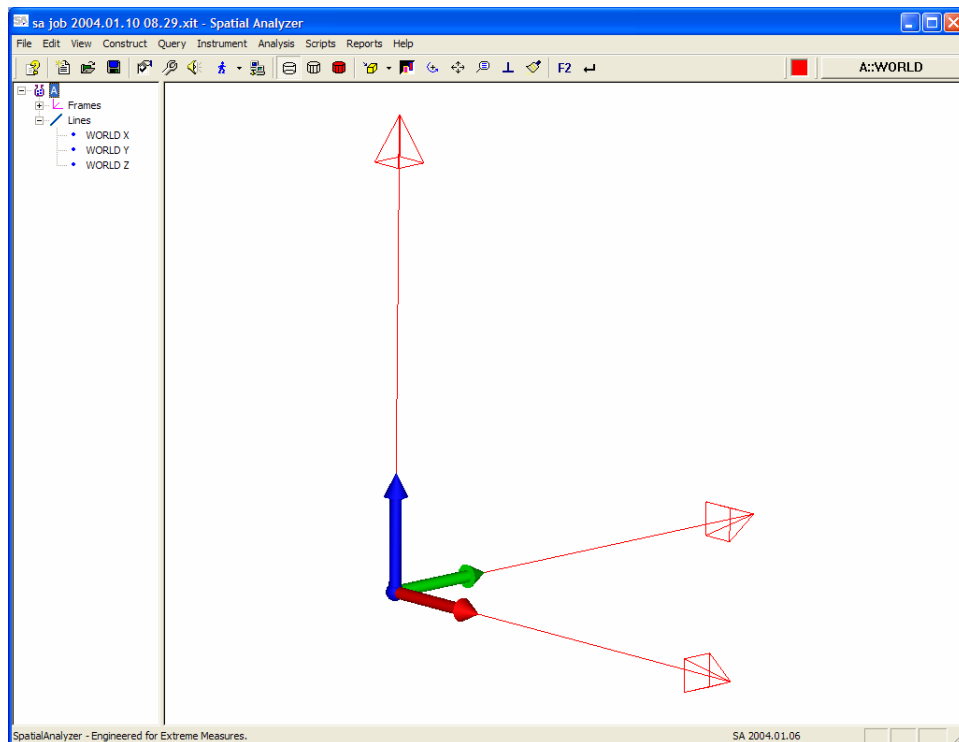
A line is created at the intersection of two planes. If the planes are parallel, the function reports that the line can not be created.

Center of Cylinder

This function is provided to allow you to easily access the axis of a cylinder in the form of a line for other analysis.

Along Frame Axes

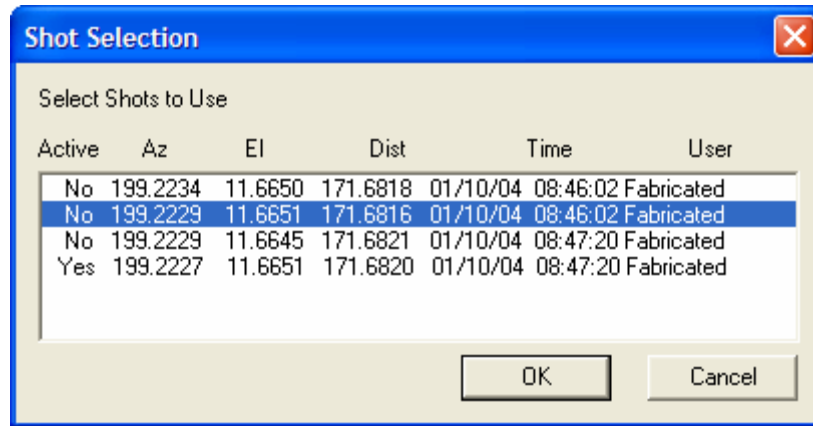
This is another function provided to make it easier to create a line for further analysis. The lines are automatically named with frame name with the axial direction as a suffix (e.g., World X, World Y, and World Z.) The figure below shows an example.



Lines Created along Frame Axes

From Instrument Shot

Creating lines for further analysis from measurements can be a useful tool. This function asks you to first select the instrument and then the target with measurements that you need to make into lines. Since a target could have been measured many times by a station a selection dialog is presented that allows you to select which shot lines to create. An example shot selection dialog is shown below.

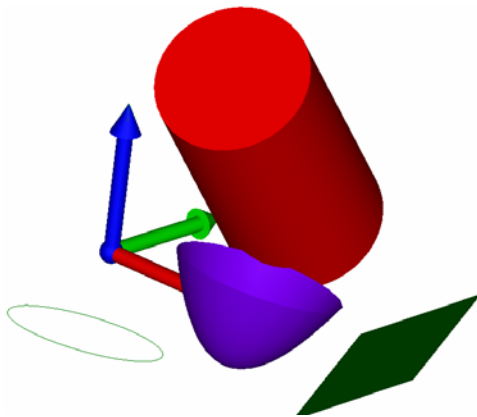


Shot Selection dialog

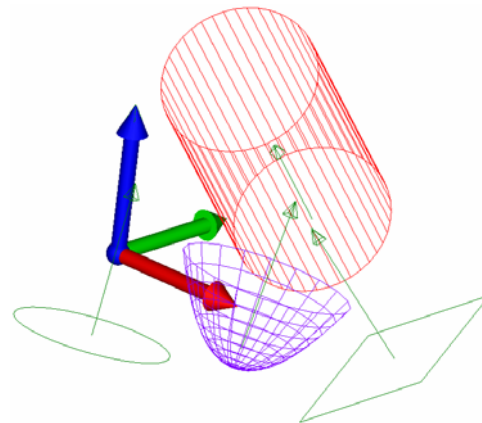
Normal to Object(s)

Creating lines on objects that have inherent normal directions is straightforward with the Normal to Objects function. Objects that have an inherent normal direction include plane, line, circle, cylinder, parabola, cone, and frame (the z-direction). A sphere is an example of an object that does not have an inherent normal direction. It is defined by a point and a radius.

Select the objects that you need to create a line on its normal. Hit the enter key to complete the process. An example of the original objects and the lines created on their normal directions is shown below.



Objects with normal directions



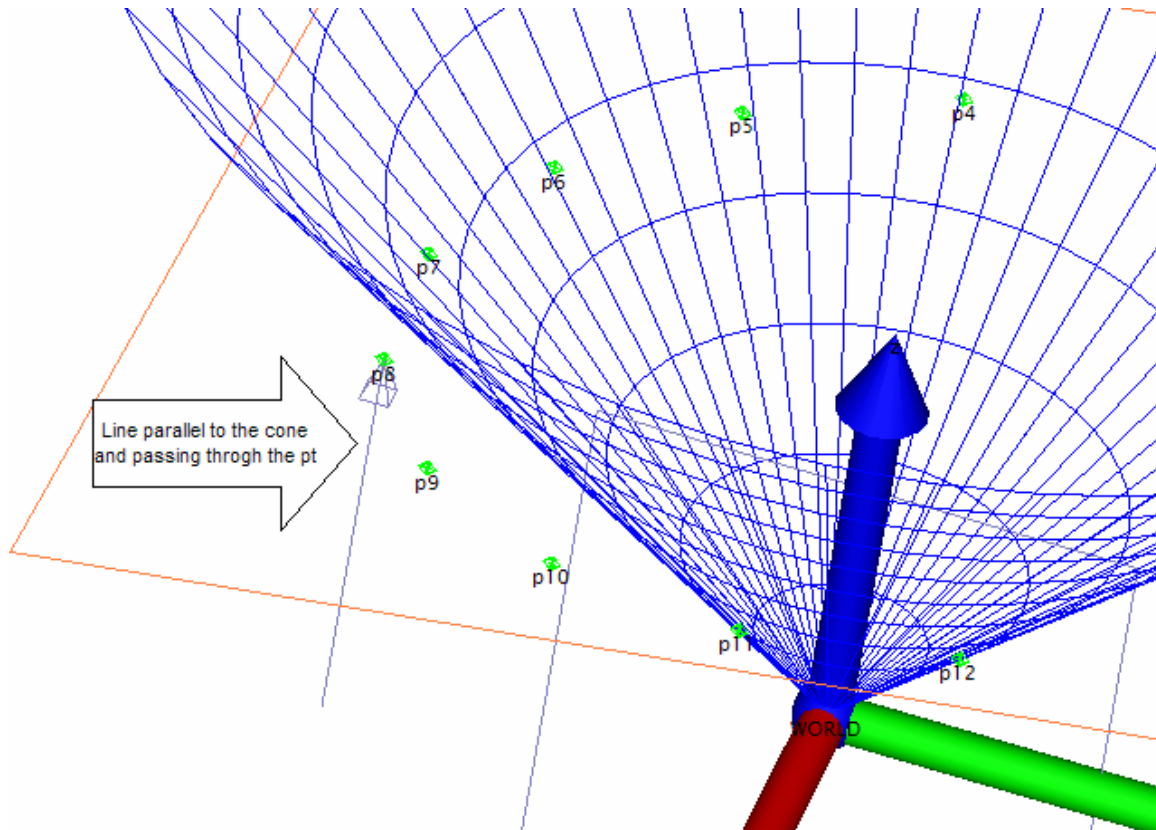
Lines created on object normal directions

Normal to Objects, through Point

Lines that are normal to an object and pass through a point are useful for many analysis tasks. The function Normal to Objects, through a Point creates lines with these properties. The created line will be parallel to the object's normal direction and pass

through a selected point. Objects that have an inherent normal direction include plane, line, circle, cylinder, parabola, cone, and frame (the z-direction).

Selecting a cone and then a point yields a line parallel to the cone's axis that passes through the point. The figure below shows an example.



Create a line parallel to object's normal and through point

Parallel to Line, Through Point

Creating a line that is parallel to an existing line that passes through a point is the output of the Parallel to Line, Through Point function. The inputs for the function are first select a line and then a point. A parallel line passing through the point is created. Graphically the line uses the selected point as one of its ends.

Circles

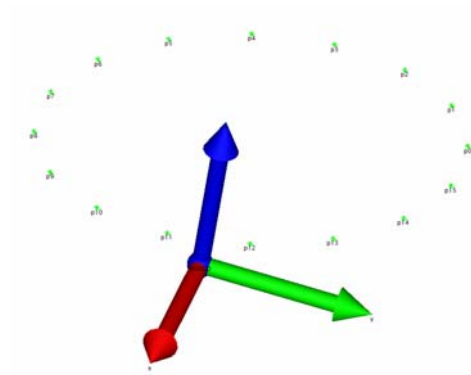
Circles are stored as a center point, a normal vector direction, and a diameter. There is also a parameter for the number of facets used to draw the curved profile. This parameter does not affect the mathematical description of the circle itself, but does affect the visual appearance.

Circles may be constructed with three different methods.

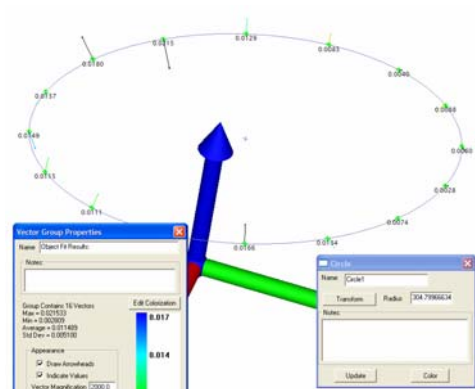
Fit to points

Best-fit circles fitted based on a series of data points. When using this option, you will be prompted to select the points that are to be used in the computation. You may individually double click on all of the points, or you may press F2 to select the points from a list. After all of the points have been selected, press , and the fit algorithm will compute the best-fit circle.

The figures below show an example of a set of points and the circle computed from them.

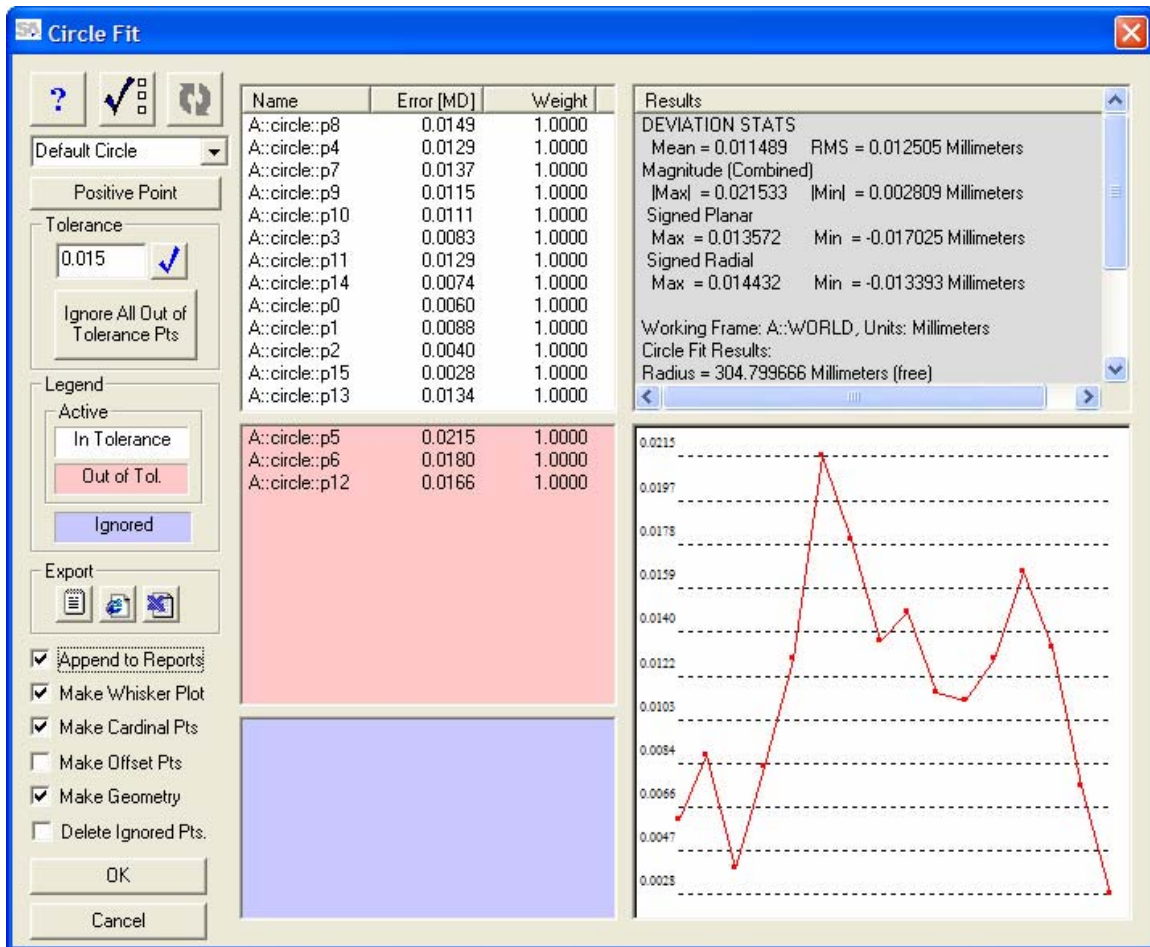


Points to create a best fit circle



Best-fit circle with vector group and center point

After computing the circle a Circle fit dialog showing the residual errors, summary, and graphical views of the residuals is presented. The figure below shows an example for the fit dialog.

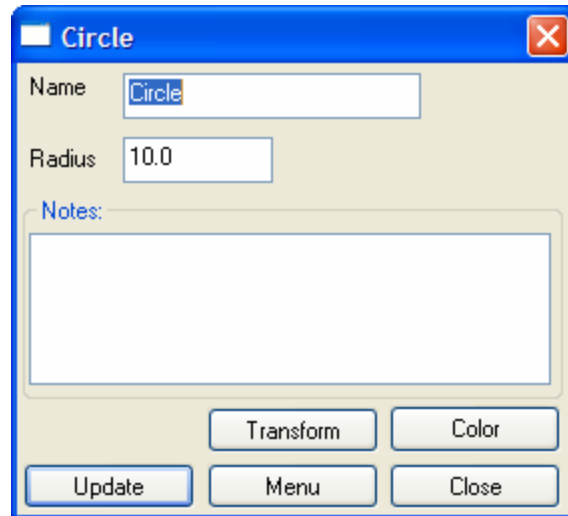


Best-fit circle with vector group and center point

Note also that many of the Plane geometry functions may also be performed on circles. This is because a circle does in fact contain a plane. You may, for example, project a point or line onto a circle in the same manner that you project them onto a plane.

Enter

This is the manual entry method for circle construction. Here the user simply selects the Construct, Circle, Enter menu option and is presented with the circle properties dialog. Using this dialog the operator may enter the circle radius, change the circle's transform, and edit the Color properties of the circle.



Circle Properties Dialog

Maximum Material Condition (from scan patch)

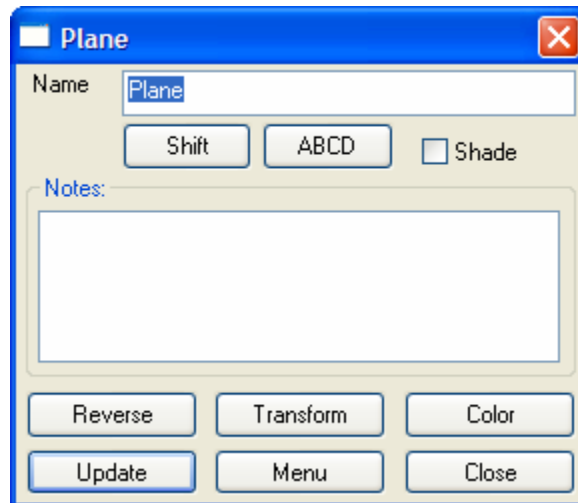
Planes

Planes are stored using the plane coefficients A, B, C, and D. These coefficients store the mathematical representation of the plane given by:

$$Ax + By + Cz + D = 0$$

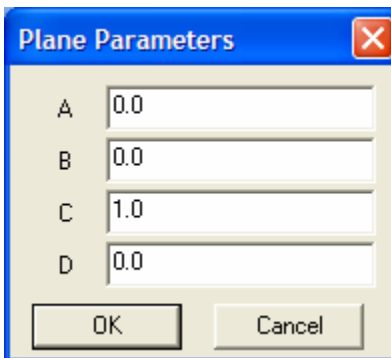
In addition to the mathematical description, there are also physical points created to define the extents of the plane. It is important to note that these extents are merely used to draw a representation of the infinite plane. All projection operations and intersection operations will use the mathematical representation of the infinite plane.

The plane properties dialog (shown below) has controls for naming the plane and setting the shading attribute. Transforming or moving the planes position and orientation is accessible with the transform button.



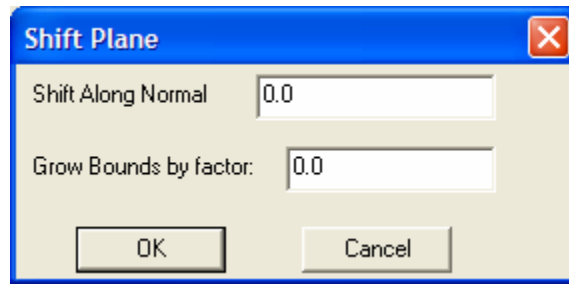
Plane Properties Dialog

Pressing the button marked ABCD produces the plane's parameter dialog. A plane is defined as a direction at a distance from the working frame. The A, B, C parameters are the normalize vector components forming the direction. D is the distance from the origin along the vector. For the example shown below the plane is in the x-y plane and passes through the origin.



Plane Properties Dialog

Shifting a plane along the normal is a common mechanism in a number of metrology applications. The Shift button on the properties dialog makes this task straight-forward. Enter the distance to shift the plane into the Shift Along Normal field. The sine of the value determines the sense of the along the normal.

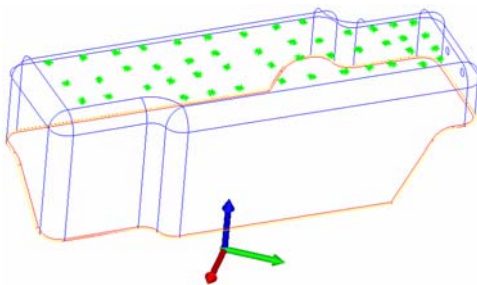


Plane Properties Dialog

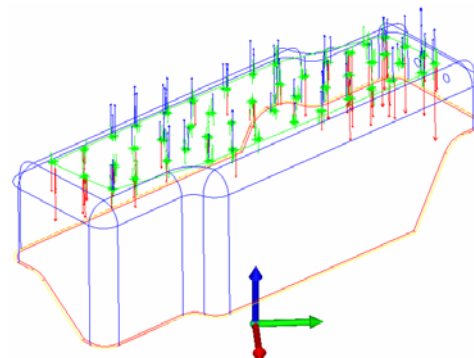
The Grow Bounds by factor allows you to grow or shrink the graphical presentation of the plane. Enter a value that is less than one to shrink the graphical re-presentation. Entering a value greater than one for the Grow Bounds by factor field grows the graphics presentation of the plane.

Fit to Points

Planes are most often created by determining the coefficients A, B, C, and D that best-fit a series of points. To do this, select the plane-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press , and the best-fit parameters will be determined and the errors will be displayed in a Fit Dialog. The two figures below show the points on a planar surface and then after the plane is solve; a vector group showing the residual errors against the computed best-fit plane.

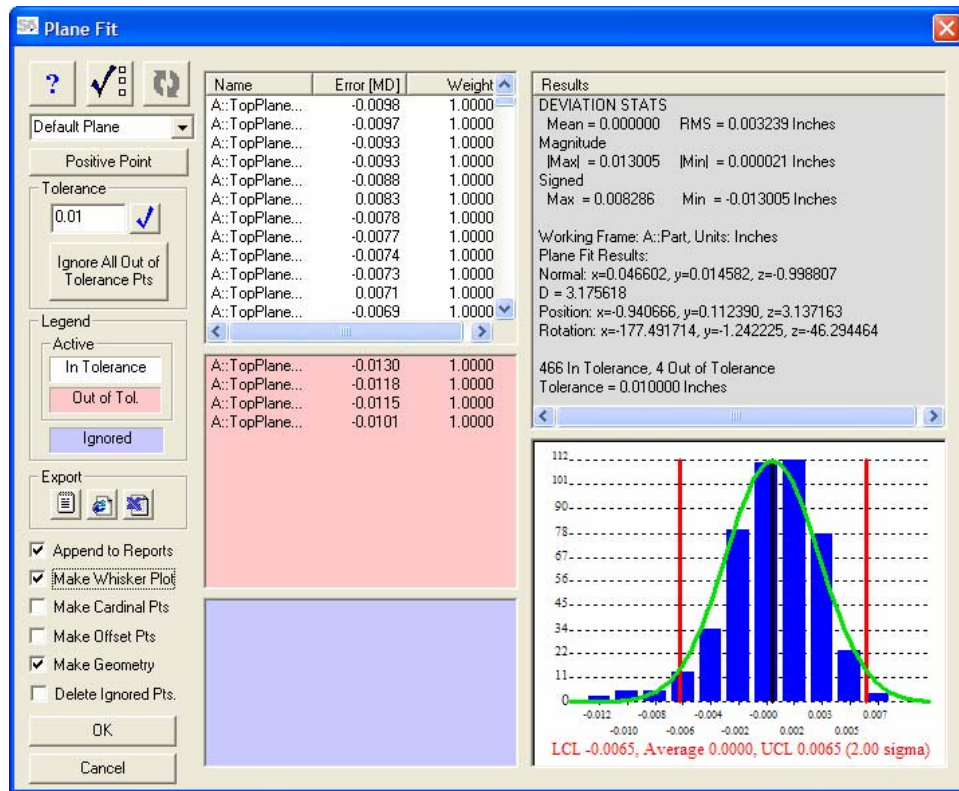


Points to fit Plane



Plane + Vector Group

The plane fit dialog provides a number of analysis views of the fit and the details of computed plane. You are able sort and modify which points are included in the fit ... resolve and then define what output analysis and reports you want created from this dialog. The figure below is an example showing the fit for the plane points. The graphic is a histogram of the residual errors. In this case the errors have a nice Gaussian profile.



Plane Fit Dialog

Enter

This is the manual entry method for plane construction. Here the user simply selects the Construct, Plane, Enter menu option and is presented with the plane properties dialog. Using this dialog the operator may enter the plane coefficients, change the plane's transform, edit the Color properties of the plane, and if desired represent the plane as a shaded object.

Normal to Object, through Point

A plane is created that is normal to a selected object and passes through a selected line. Objects that have an inherent normal direction include plane, line, circle, cylinder, parabola, cone, and frame (the z-direction).

Normal to Line, through end

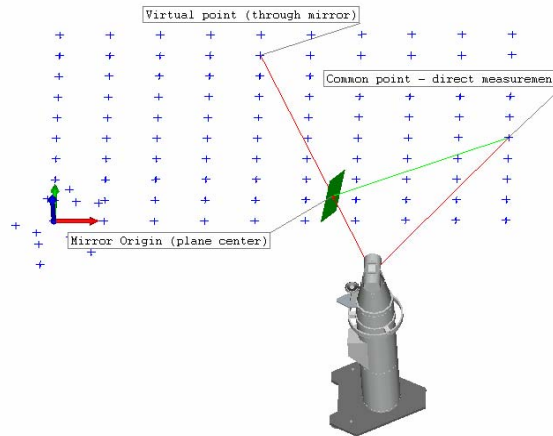
A plane is created that is normal to the line and passes through a selected end of line.

Mirror from 2 measured points

The mirror plane function is used to determine the location of a mirror by measuring a point both directly and through the mirror. SA also includes uncertainty analysis for the location and orientation of the mirror.

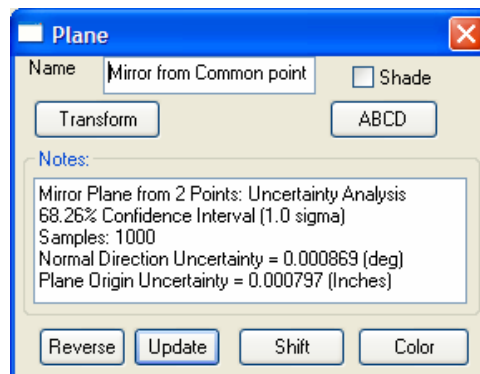
The process follows this outline:

- Pick the instrument.
- Pick the point measured directly (not through mirror)
- Pick the virtual point that is really the same point, but measured through the mirror.
- SA creates a plane representing the mirror location. You can make a frame on this plane if needed.



Layout for Mirror from 2 Measured Points

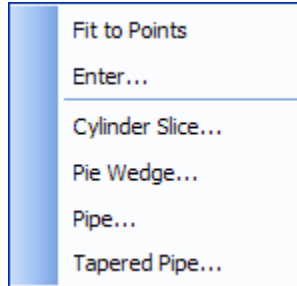
If the points used in the construction have uncertainty fields (Analysis>>Coordinate Uncertainty>>Create Point Uncertainty Fields), SA will automatically compute the uncertainty of the mirror plane and add the results to the notes field for the constructed plane:



Plane Properties Results Mirror from 2 Measured Points function

Cylinders

A cylinder is defined using two points and a diameter. There is also a facets parameter, but as with circles, this parameter only affects the graphical representation of the geometry.



Cylinder Construction Options

Enter

In the Spatial Analyzer, the cylinder properties window allows the user to enter and view the parameters in two ways. One method displays the A and B coordinates as well as the diameter. The other method uses a transformation matrix to place the base of the cylinder, a length along the Z axis of that transformation, and a diameter. The interface will allow you to toggle between these two description options.

Cylinder

Name:

☒ Transformation Method

Length

☐ End-Point Method

	Begin	End
X	<input type="text" value="0.0"/>	<input type="text" value="0.0"/>
Y	<input type="text" value="0.0"/>	<input type="text" value="0.0"/>
Z	<input type="text" value="0.0"/>	<input type="text" value="20.0"/>

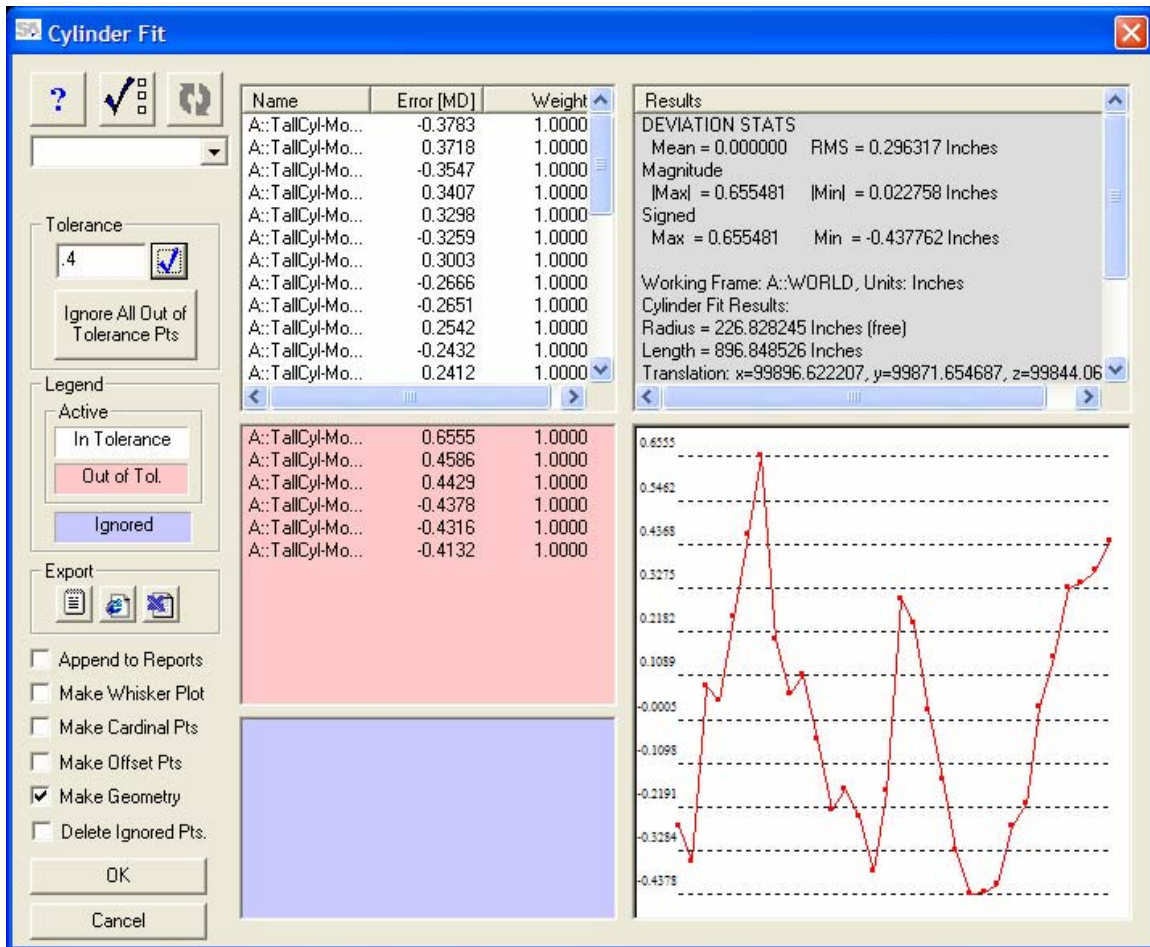
Diameter Facets

Notes:

Cylinder Properties Dialog

Fit to Points

One may also create Cylinders by using a best-fit algorithm. This algorithm will take a series of points and determine the cylinder that best-fits the given point field. The interface for this option is similar to that for circles and planes. Once the points are selected, you will be prompted for an initial guess at the cylinder's diameter. This helps the algorithm to find the appropriate solution. Next, the fit is performed, the errors are displayed, and the cylinder will appear.



Cylinder Fit Dialog

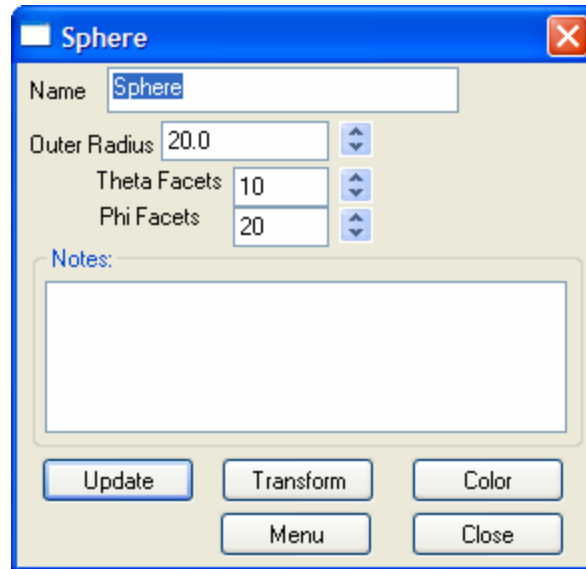
There is also a "re-fit to data" option on the Construct menu (Construct >> Re-Fit Geometry >> Use existing geometry as a starting point). This option allows you to perform the fit operation again using the current cylinder as the starting guess. This is to help the algorithm find the appropriate solution when dealing with point data that represents "squat" cylinder. When you press the re-fit button, you will be prompted to select the points for the fit, and the operation will be performed.

Sphere

Spheres are stored as a center point and a radius. There is also a facets parameter, but as with circles, this parameter only affects the graphical representation of the geometry.

Enter

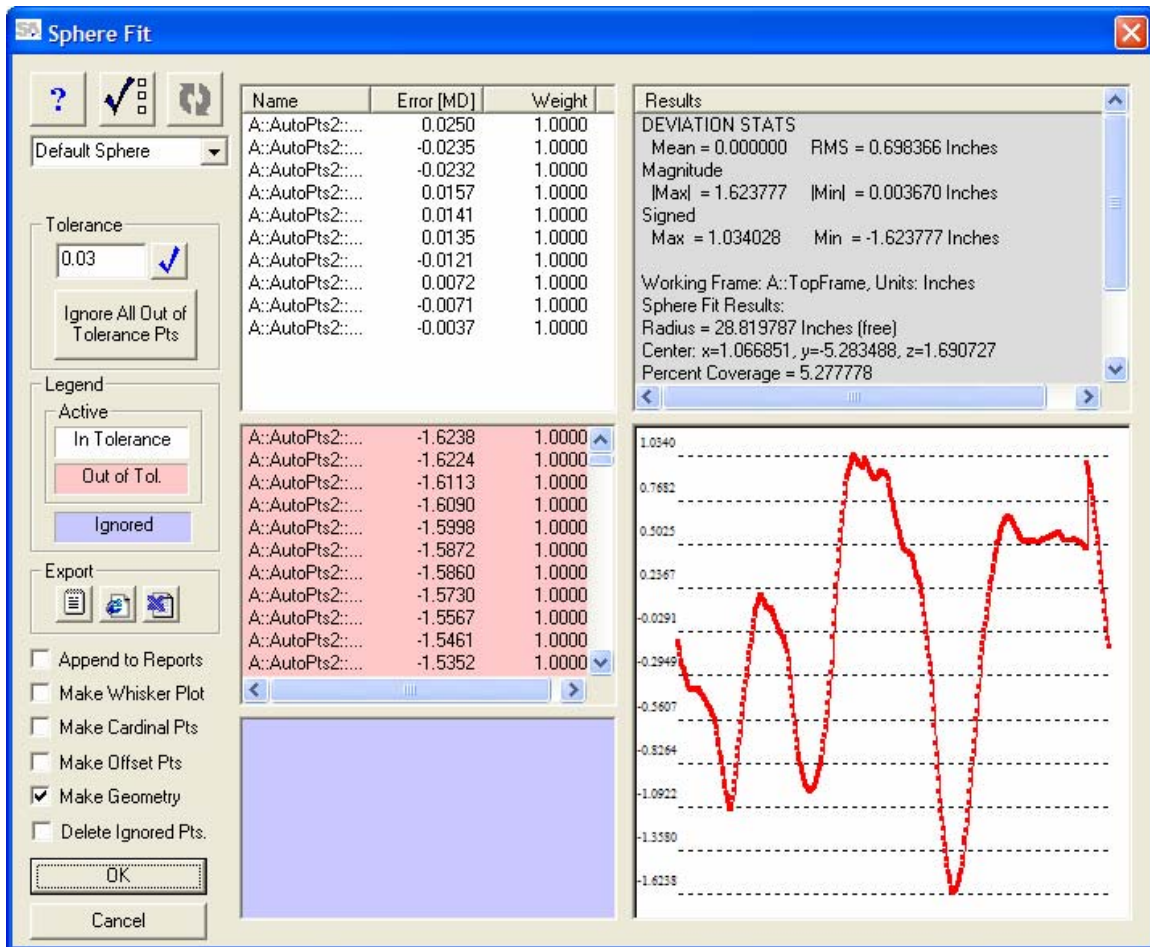
This is the manual entry method for Sphere construction. Here the user simply selects the Construct, Sphere, Enter menu option and is presented with the sphere properties dialog. Using this dialog the operator may enter the sphere center coordinates, change the sphere's radius, and edit the Color properties of the sphere.



Sphere Properties Dialog

Fit to Points

Spheres are most often created by determining the center point and radius that best-fit a series of points. To do this, select the sphere-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press enter and the best-fit parameters will be determined and the errors will be displayed in the fit dialog. An example Sphere fit dialog is shown in the figure below.



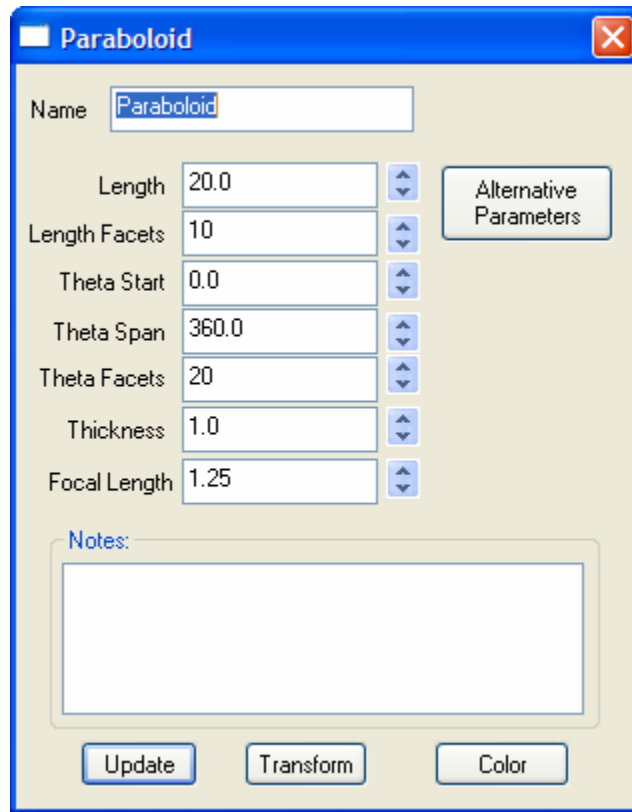
Sphere Fit Dialog

Paraboloid

Paraboloids are stored as a transform, a physical length, and a focal length. There are also several other presentation parameters, but these parameters only affect the graphical representation of the geometry.

Enter

This is the manual entry method for cone construction. Here the user simply selects the Construct, Paraboloid, Enter menu option and is presented with the paraboloid properties dialog. Using this dialog the operator may enter the paraboloid transform, change the paraboloid's physical length and focal length, and edit the Color properties of the paraboloid.



The image shows a software dialog box titled "Paraboloid". It contains several input fields for defining a paraboloid's properties. The "Name" field is set to "Paraboloid". The "Length" field is 20.0, "Length Facets" is 10, "Theta Start" is 0.0, "Theta Span" is 360.0, "Theta Facets" is 20, "Thickness" is 1.0, and "Focal Length" is 1.25. Each of these numerical fields has up and down arrow buttons next to it. To the right of these fields is a button labeled "Alternative Parameters". Below the input fields is a "Notes:" label followed by a large empty text area. At the bottom of the dialog are three buttons: "Update", "Transform", and "Color".

Paraboloid Property Dialog

Fit to Points

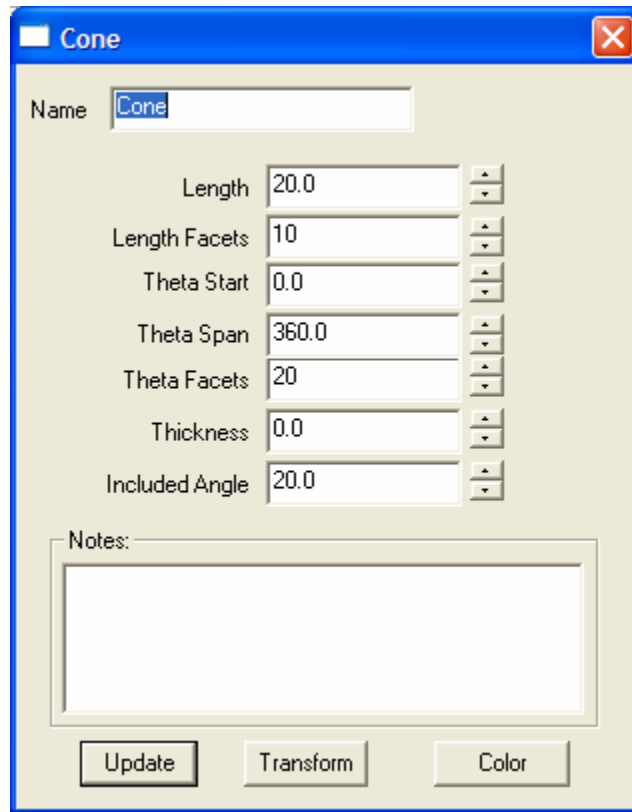
Paraboloids are most often created by determining the transform and focal length that best-fit a series of points. To do this, select the paraboloid-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press enter and the best-fit parameters will be determined and the errors will be displayed in the fit dialog.

Cone

Cones are stored as a transform, a length, and an included angle. There are also several other presentation parameters, but these parameters only affect the graphical representation of the geometry.

Enter

This is the manual entry method for cone construction. Here the user simply selects the Construct, Cone, Enter menu option and is presented with the cone properties. Using this dialog the operator may enter the cone transform, change the cone's length and included angle, and edit the Color properties of the cone.



The image shows a software dialog box titled "Cone". It has a blue title bar with a close button (X) in the top right corner. The main area is light beige. At the top, there is a "Name" label followed by a text box containing the word "Cone". Below this, there are seven rows of numerical input fields, each with a label to its left and a small up/down arrow button to its right. The labels and values are: "Length" (20.0), "Length Facets" (10), "Theta Start" (0.0), "Theta Span" (360.0), "Theta Facets" (20), "Thickness" (0.0), and "Included Angle" (20.0). At the bottom of the dialog, there is a "Notes:" label followed by a large empty text area. Below the notes area are three buttons: "Update", "Transform", and "Color".

Cone Property Dialog

Fit to Points

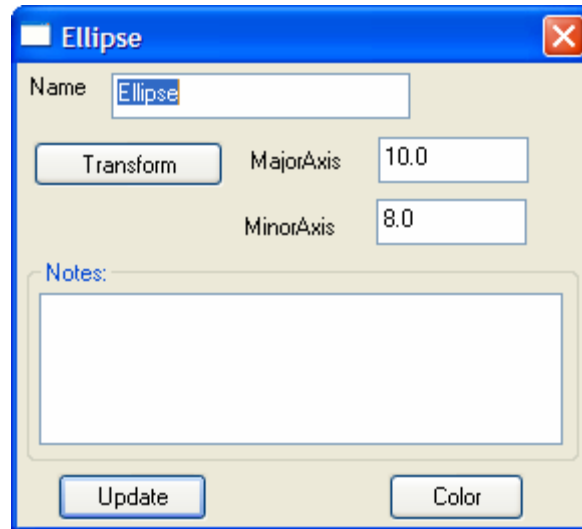
Cones are most often created by determining the transform, length, and included angle that best-fit a series of points. To do this, select the cone-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press enter and the best-fit parameters will be determined and the errors will be displayed.

Ellipse

Ellipses are stored as a transform, a major axis, and a minor axis. There is also a facets parameter, but as with circles, this parameter only affects the graphical representation of the geometry.

Enter

This is the manual entry method for ellipse construction. Here the user simply selects the Construct, Ellipse, Enter menu option and is presented with the ellipse properties dialog. Using this dialog the operator may enter the ellipse transform, change the ellipse's major and minor axis values, and edit the Color properties of the ellipse.

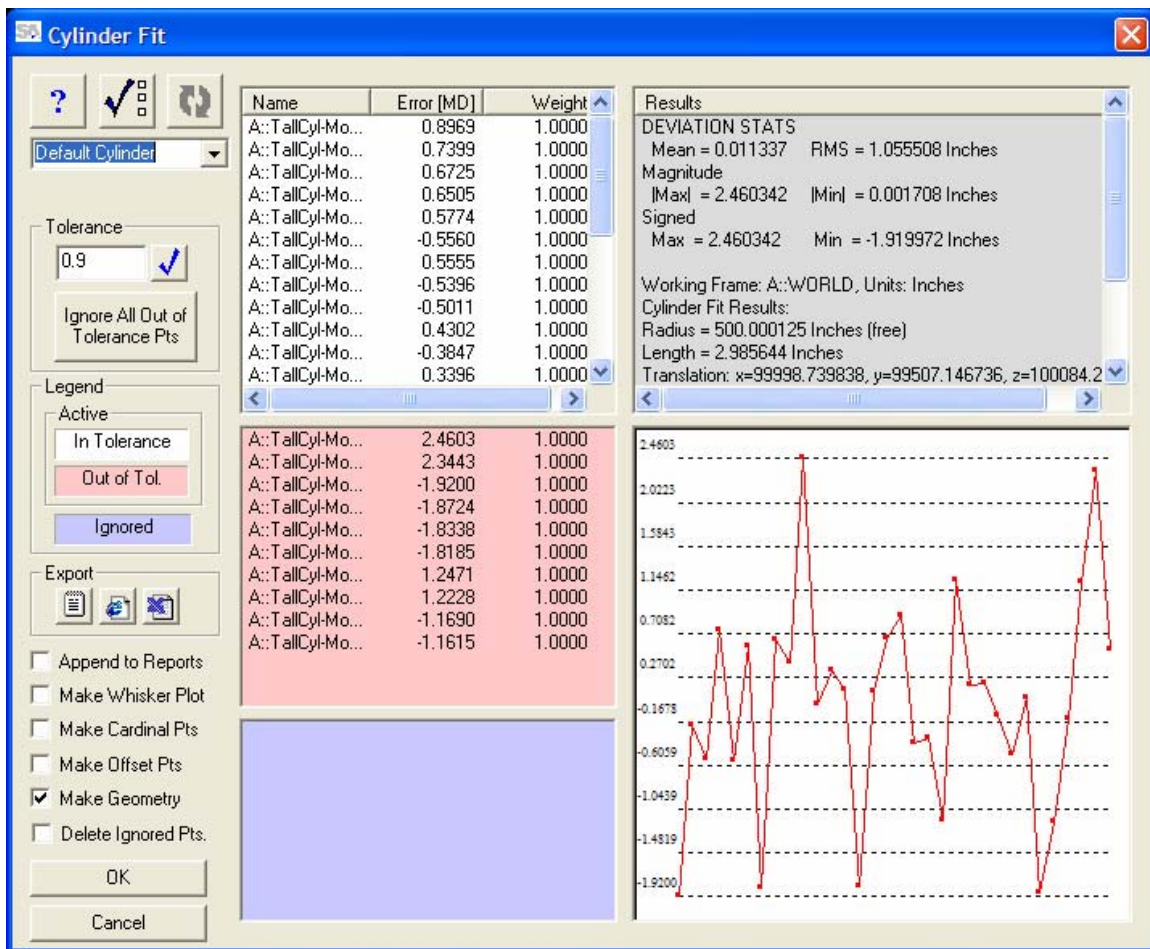


Ellipse Properties Dialog

Re-Fit Geometry, Use Existing Geometry as Starting Point

When using numerical methods to solve for geometries from data sets a number of interesting out-comes can be produced. SA provides an alternate method for solving for geometric objects from points. The 'Re-Fit Geometry, Use Existing Geometry as Starting Point' function can use an object as an initial guess and then associate a dataset of points to use in a numeric optimization of the object.

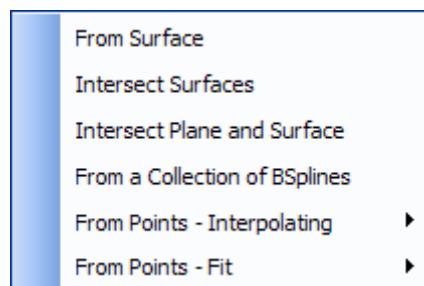
The function asks you first to select the object that will serve as the initial guess and then the points to use in the solution. The results are then displayed in a fit dialog. An example for re-fitting a cylinder is shown below.



Re-Fit Cylinder Fit Dialog

Curves

You can use these options to construct curves using either a linear or a cubic model. You will be asked to select the points for the curve, and the curve will be constructed.



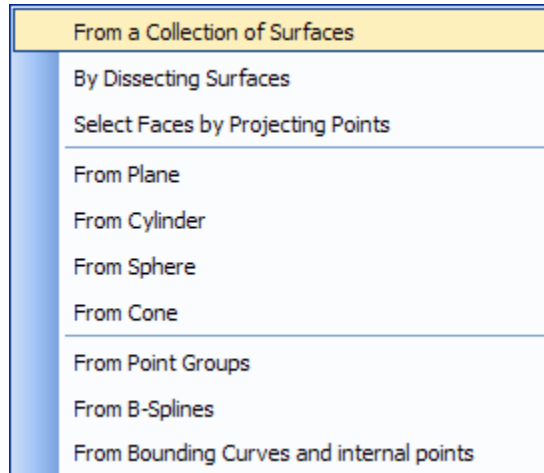
Construct Curve Submenu

From Surfaces

Curves are created on the bounding edges of the selected surface. Each curve is segmented into the constituent of the surfaces.

Surfaces

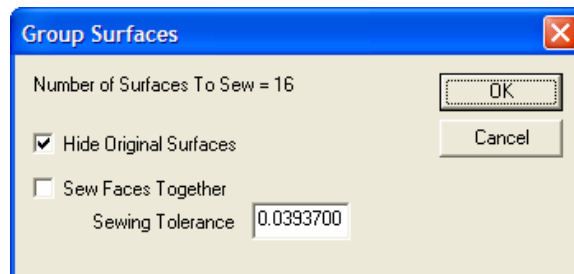
SpatialAnalyzer contains surfacing functionality. You can import nominal surface information from IGES files, compare it to measurements, and produce colorized error plots. In addition, you can best-fit a series of points to a nominal surface and perform a transformation.



Construct Surface Submenu

From a Collection of Surfaces

Surfaces can be merged into a single object for use in analysis functions with this function. Select the surfaces to collect into the common object and press enter. The next step asks for input on whether the original surfaces should be hidden and if the surface faces should be sewn together. An example of the dialog is shown below.



Group Surfaces dialog

After accepting the inputs in the dialog the surfaces are merged together and put on the tree view as a single object. The original surfaces will be hidden if the default option is selected.

By Dissecting Surfaces

If several surfaces have been merged together they can be divided into individual surfaces. Select the surfaces to dissect and hit the enter key. The component surfaces are then created from the parent and put on the tree view.

From Plane

Creating a planar surface is supported with this function. Select a plane and a surface object is created from it and put on the tree view.

From Point Groups and From B-Splines

You can construct a Non-Uniform Rational B-Spline Surface using groups of measured points. In this case, the coordinates must be grouped in scan-line order. In addition, each successive group of points must proceed in the same direction (i.e. left to right for example).

The process of fitting a NURB surface to measured points is quite complex. You must select the order of the surface, and the number of control points you wish to use in both the U and V parametric axes.

The algorithm will then construct a bi-parametric cubic spline surface using the selected points. Using this surface, the initial values for the control points will be guessed by evenly sampling the cubic surface in the U and V direction.

This option allows you to build the surface using only point data. This data must be ordered in terms of scan lines. Within this option, there are several methods for making the point selection:

- Enter Group Names – This option requires you to type in the names of the groups that you wish to fit the surface to.
- Select a Point from Each Group – You will be allowed to select a single point from each group, and the program will use the entire group for the scan line. Remember to pick the groups in the proper order.
- Pick Window Corners – In the case of a laser scanning device, the scan lines can automatically be imported in properly ordered groups. An example is the RAS file format for the CLR measurement device. In this case, the groups are also sequential in the database. As a result, you can pick two points (one in each corner), and the program can determine both the groups to use and the subset of points within the group to use for the scan lines.

From B-Splines

Bi-parametric cubic spline surfaces are composed of a series of parametric cubic curves running in the u direction. By sampling these curves with a uniform parametric sampling, it is possible to construct another set of curves traversing the v direction. These curve definitions produce a surface definition.

If you are to construct a bi-parametric cubic spline surface, you must sample all of the contour lines in the same direction with no overlap. In addition, place each individual line scan in a separate group.

In this case, you select a series of parametric curves in the proper order, and the program will sample them and construct the bi-parametric surface. This requires creating the curves before creating the surface. To do this, use the Curve construction option discussed previously.

From Bounding Curves and Internal Points

From Plane and Bounding Closed Curve

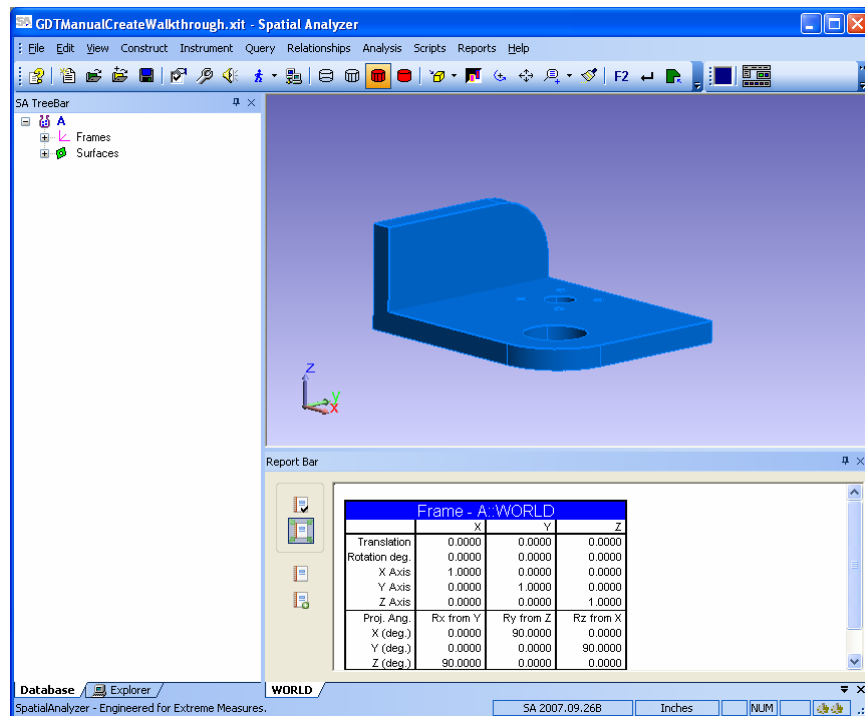
GD&T MODULE

Overview

This feature provides the user with the capability of creating GD&T annotations for datums and feature checks, removing the need to have the annotations already present in the CAD part file. Previously, it was necessary to have a CAD part file (specifically CATIA V5) which already had GD&T annotations in order to work with the GD&T features in SA. With this new functionality, users can import a CAD part and decorate it themselves with GD&T datum and feature check annotations. This guide walks through a sample CAD part showing how to decorate that part with GD&T annotations and then use the newly created annotations.

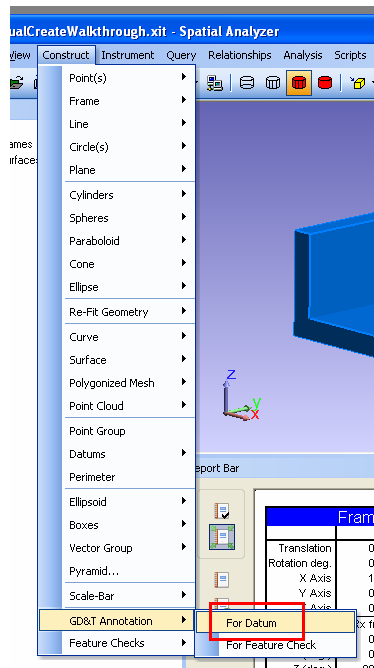
Getting Started

Start Spatial Analyzer, and open the undecorated sample part job file (GDTManualCreateWalkthrough.xit). You must be running SA v2007.09.27 or newer to continue. Switch to solid shading with edges display and you should see the below.

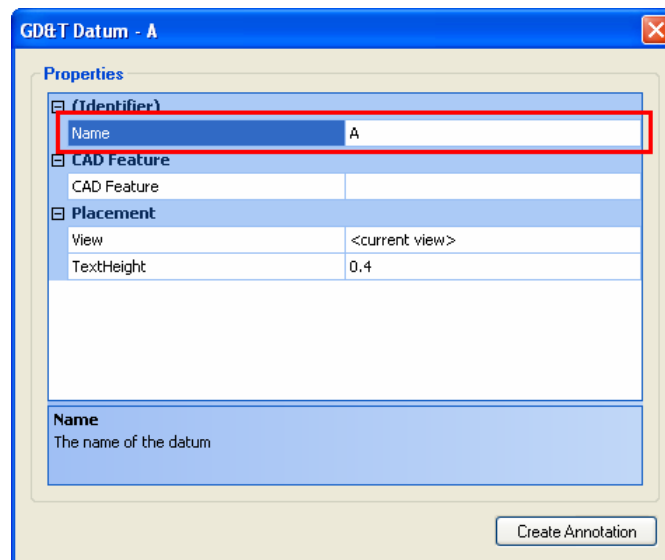


Creating GD&T Datum Annotations

Select "Construct>>GD&T Annotation>>For Datum" from the menu.

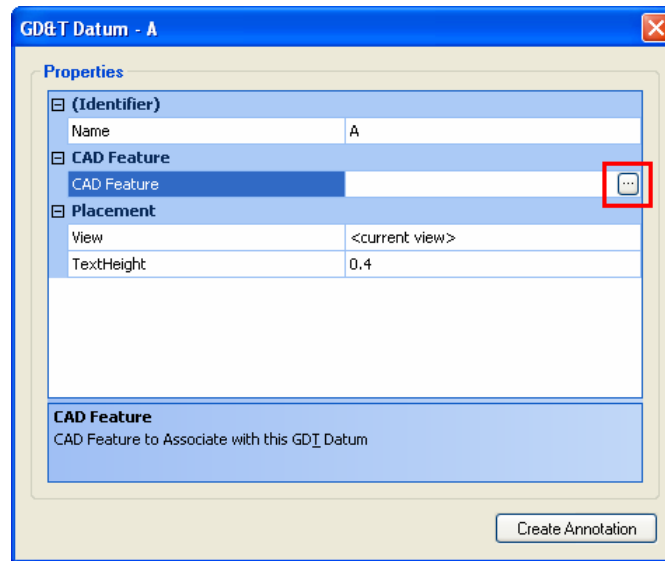


This will open the Datum Annotation Editor shown below. Fill in the Name as shown below. This will be Datum A.

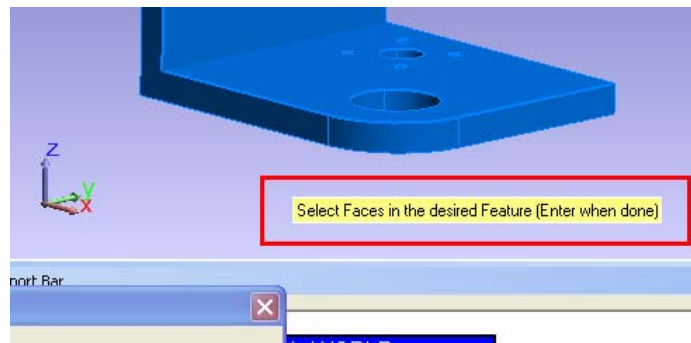


It is now necessary to choose a feature in the part with which this datum will be associated. A feature may contain multiple faces (in the case of a cylinder with the 2 halves of the shape). The feature type you select will determine the datum type which eventually gets created. For this example, we are interested in creating a planar datum, so we will select a single planar face.

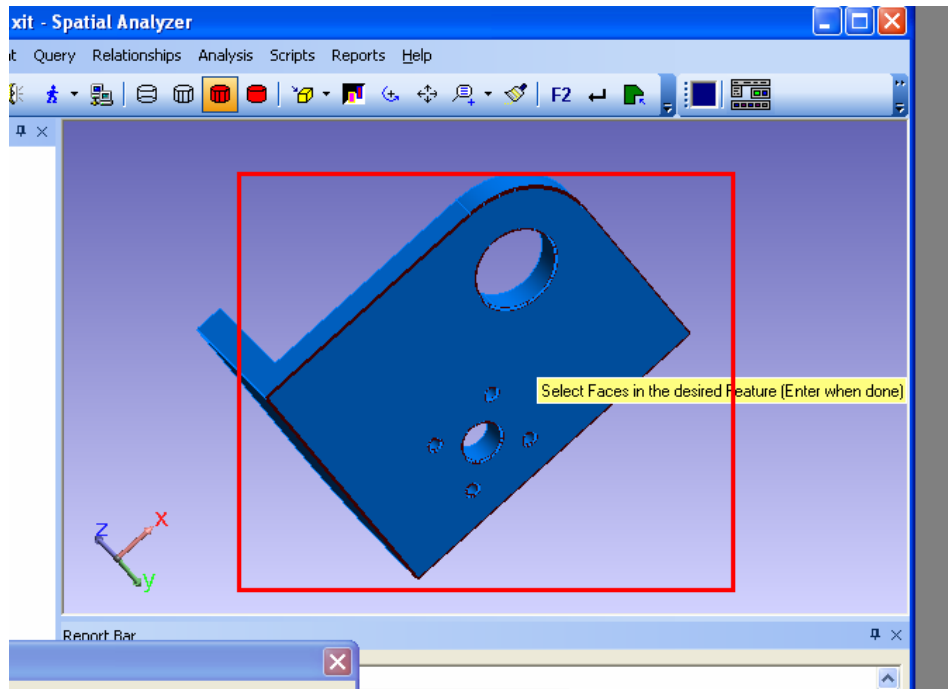
To select the feature for this datum annotation, select the “CAD Feature” field and you should see a button with “...” appear. Click this button.



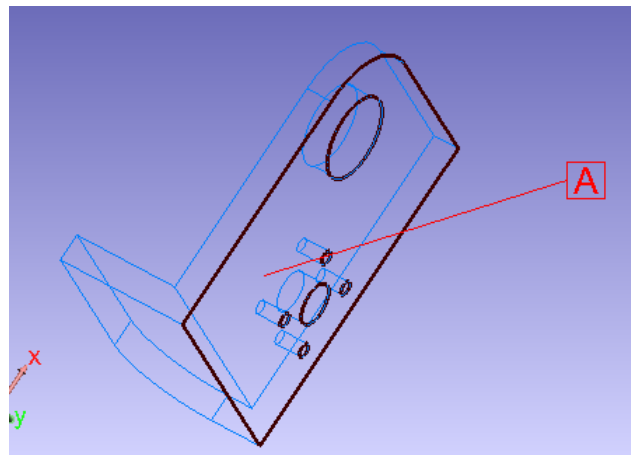
This will initiate an SA face selection mode as shown below. In this mode, you will need to double-click on the desired faces and when done hit the ENTER key.



Rotate the part around so you can see the bottom face and double-click on it. The face will highlight to confirm your selection. Since we're only selecting a single planar face for this datum, hit ENTER to complete the selection.

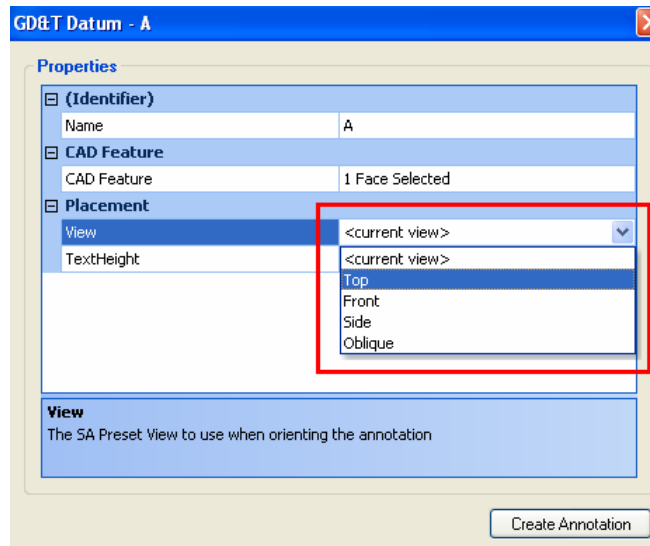


Switch to wireframe mode and you'll be able to see a rendering of the datum annotation. You can click and drag it around. While dragging it, the associated faces will highlight as a reminder of which faces the annotation goes with.



At this point, the annotation will render parallel to the screen. This allows you to rotate the view around while still maintaining a clear view of the annotation. Before creating the annotation, you'll need to pick one of the SA preset views to orient the annotation.

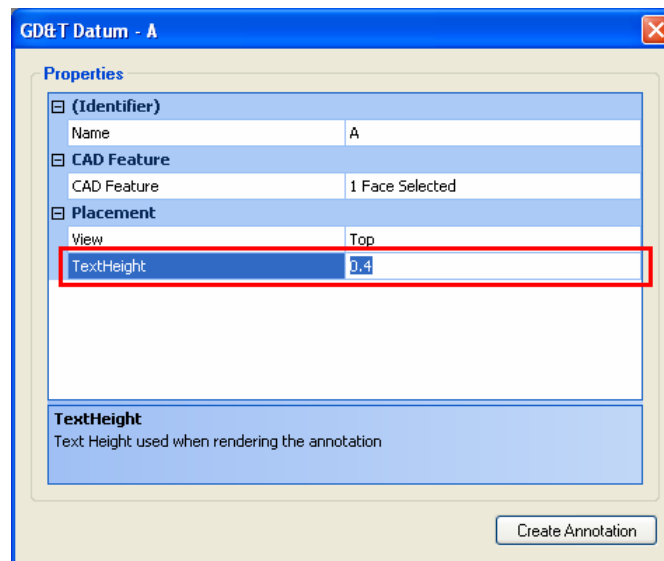
In the Datum Editor, use the View field to pick one of the SA Preset views.



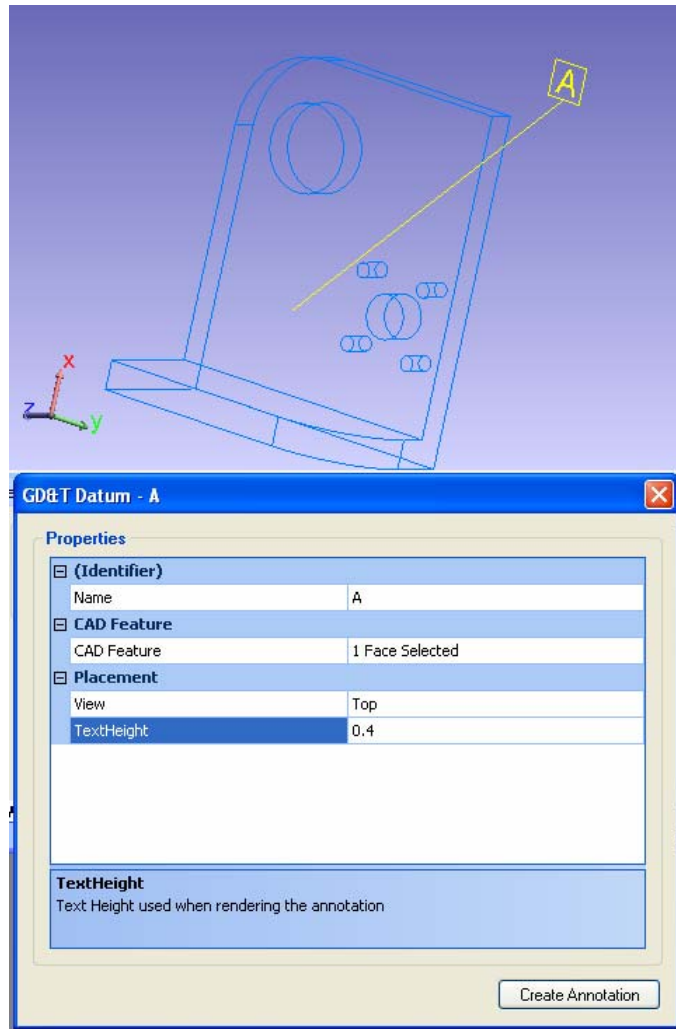
The Preset View choice controls the rotation of the annotation. When picking a Preset View the annotation is rotated to be parallel to that view and with the screen when that view is selected.

The position of the annotation is controlled by dragging it around with the mouse. When you drag an annotation it moves in its plane (determined by the selected Preset View).

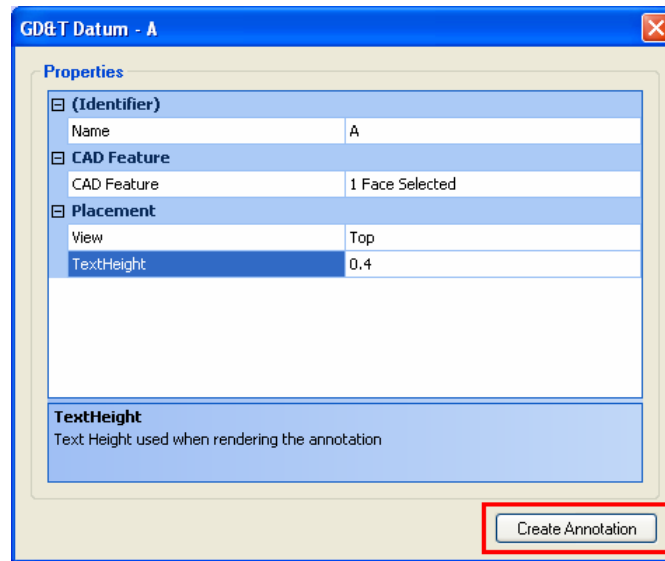
In addition to orienting and positioning the annotation, you can also customize the text size of the annotation using the TextHeight field as shown below.



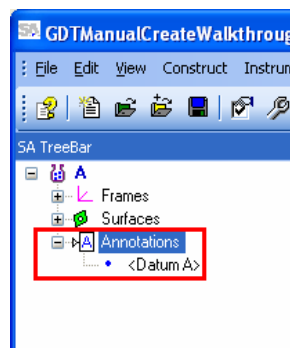
When finished the annotation should look similar to the below (assuming top view was chosen).



To complete the process, click the “Create Annotation” button.

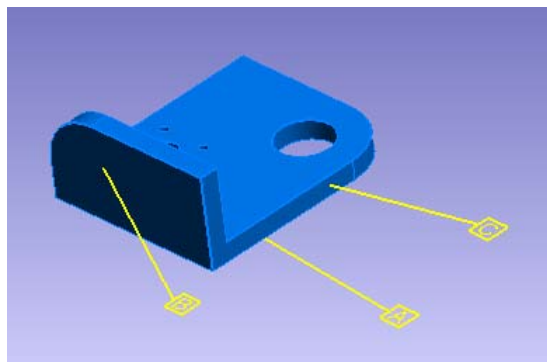


A datum annotation should be added to the job and the editor window should close.



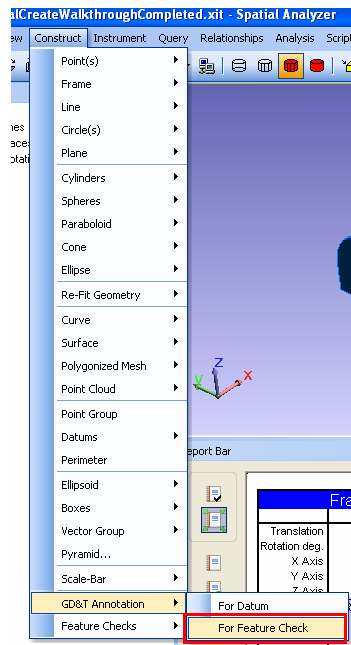
Repeat this process and create two more planar datums B and C. Datum B should be associated with the large side face and Datum C should be associated with the thin back edge face.

When finished, you should have something which looks similar to the below.



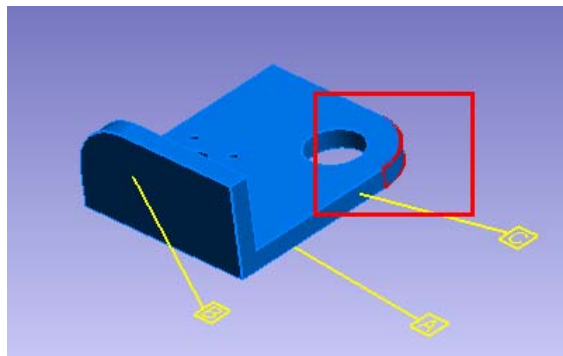
Creating a GD&T Feature Check Annotation

Select “Construct>>GD&T Annotation>>For Feature Check” from the menu.

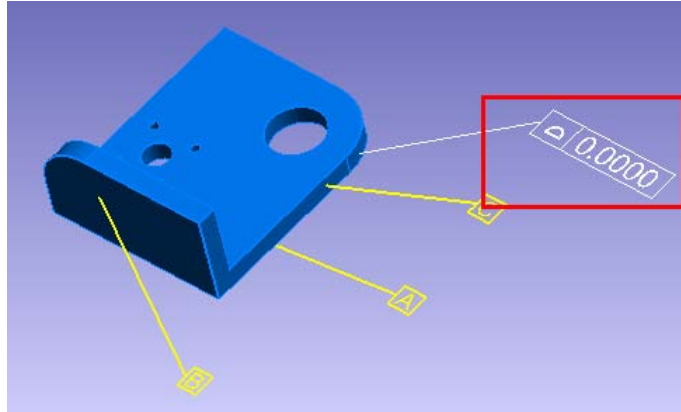


This will open the GD&T Feature Check Annotation Editor. Select “Surface Profile” using the drop-down in the Type field. Enter “SampleSurfProfileCheck” in the Name field.

Feature Checks can be associated with multiple features (for example, a true position which is associated with 4 hole features (each having 2 faces)). For now, enter 1 in the “# CAD Features” field and a new field, “CAD Feature #1” will be added. Use this field just as with the datum annotation creation to pick the faces which make up the desired feature in the CAD part. For this walkthrough, select the corner surface face adjacent to Datum C as shown in the below screenshot.



Once the feature is selected, the annotation will render in the 3D view (it also renders at the top of the window in 2D). You can now drag it around, choose the preset view for orientation, and modify the text height. Pick the top Preset view again so it ends up in the same plane as the datum annotations.



Now we'll need to specify datums for the feature check. Enter 1 in the “# Primary Datums”, “# Secondary Datums”, and “# Tertiary Datums” fields. As you do this, additional fields will be added allowing you to choose the individual datums. Choose “A” as the primary datum, “B” as the secondary datum, and “C” as the tertiary datum.

Now Enter a tolerance value. When done, the properties should look similar to the below (the tolerance value is scrolled off the bottom in the below screenshot but should just be whatever value is desired).

GD&T Annotation - SampleSurfProfileCheck

Annotation

⏏ 3.1416 A B C

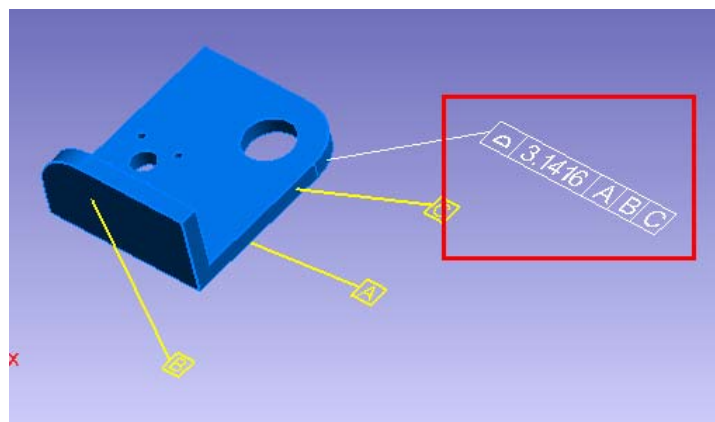
Properties

Type	Surface Profile
Features	
# CAD Features	1
CAD Feature #1	1 Face Selected
Placement	
View	Top
TextHeight	12.000
Datums	
# Primary Datums	1
# Secondary Datums	1
# Tertiary Datums	1
Primary	
Datum 1	
Datum	A
MaterialModifier	NoMCM
Secondary	
Datum 1	
Datum	B
MaterialModifier	NoMCM
Tertiary	
Datum 1	
Datum	C
MaterialModifier	NoMCM
Tolerance	
Tolerance	3.14159

Tolerance
Tolerance value

? Create Annotation

When finished the annotation in the 3D view should match the properties entered in the editor.



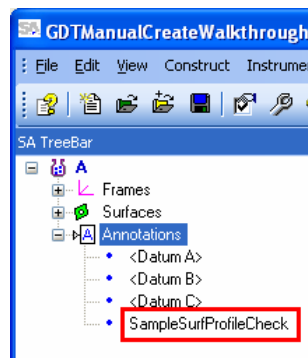
Click the “Create Annotation” button.

Properties	
Type	Surface Profile
Features	
# CAD Features	1
CAD Feature #1	1 Face Selected
Placement	
View	Top
TextHeight	12.000
Datums	
# Primary Datums	1
# Secondary Datums	1
# Tertiary Datums	1
Primary	
Datum 1	
Datum	A
MaterialModifier	NoMCM
Secondary	
Datum 1	
Datum	B
MaterialModifier	NoMCM
Tertiary	
Datum 1	
Datum	C
MaterialModifier	NoMCM
Tolerance	
Tolerance	3.14159

Tolerance
Tolerance value

Buttons: ? (Help), Create Annotation

An annotation should be added to the job and the editor window should close.

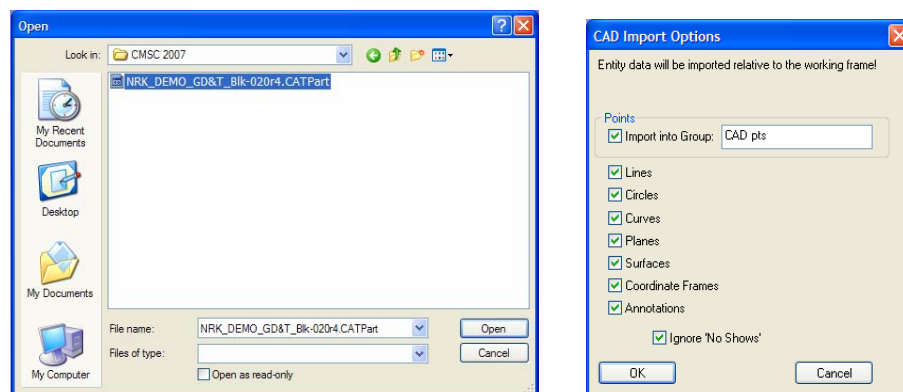
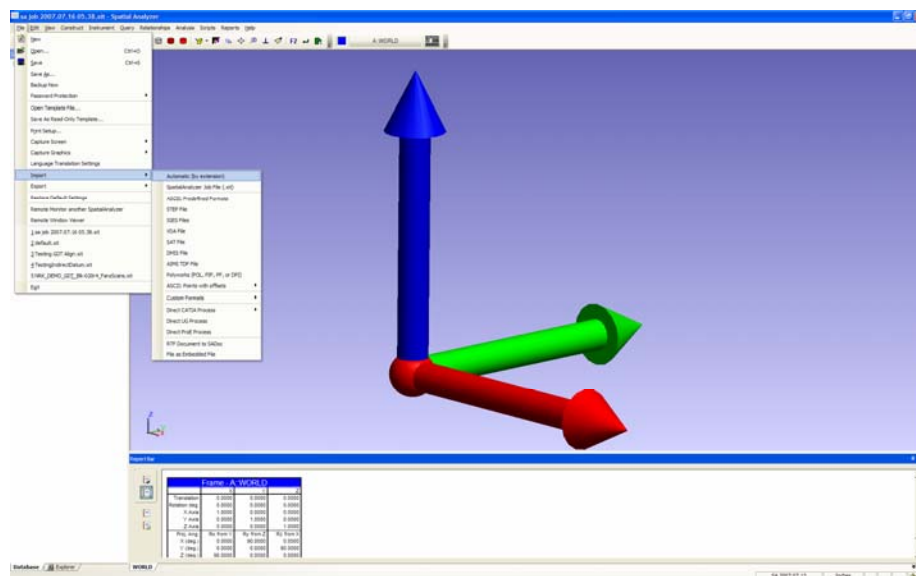


Now that the datum and feature check annotations are created, they can be used just as though they had been imported with the CAD part.

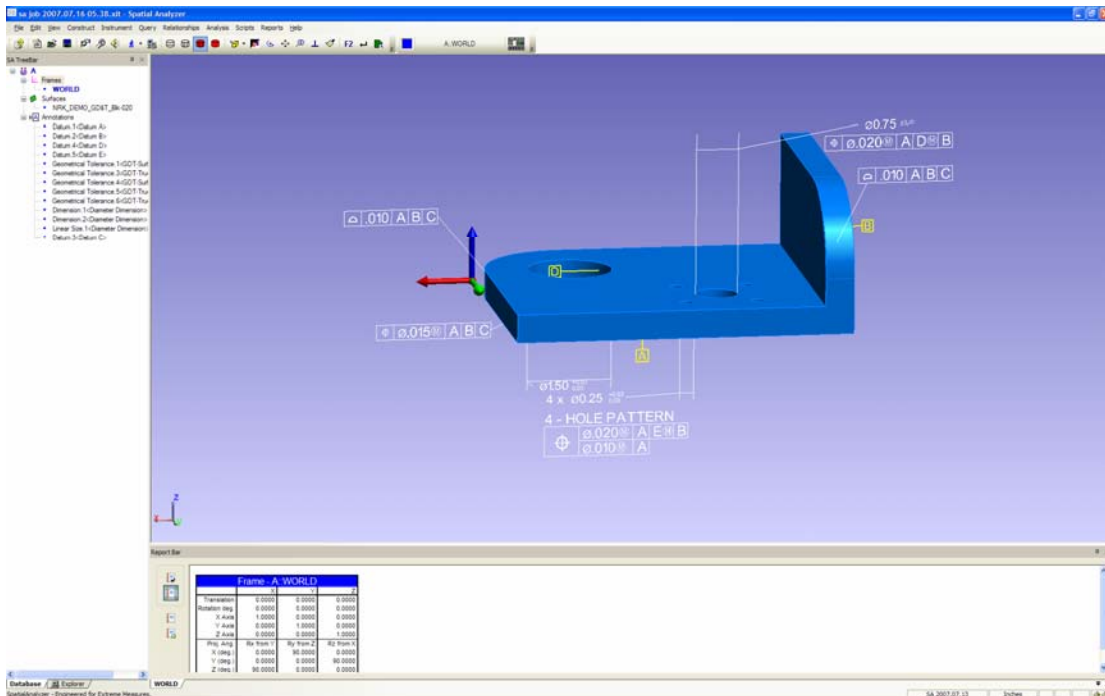
GD&T Module

Geometrical Dimensioning & Tolerancing (GD&T) is available as an add-on to Spatial Analyzer's (SA) CATIA V5 import option as of release 2007.07.21, while availability to the rest of the SA community will follow later this year. SA's feature checks, which derive from tolerance frame annotations, function in a similar manner as SA's relationships. Measured data is associated with the feature check and reported via the Quick Reporting and Report Bar mechanism.

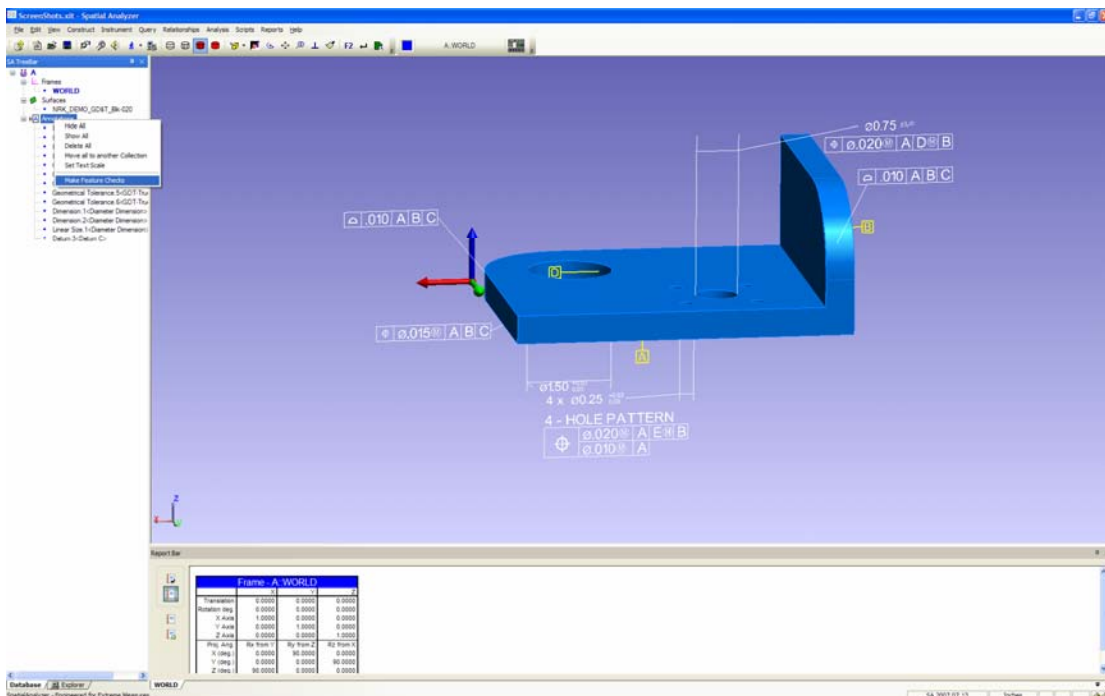
Let's review a sample session. We begin by importing the CATIA model.



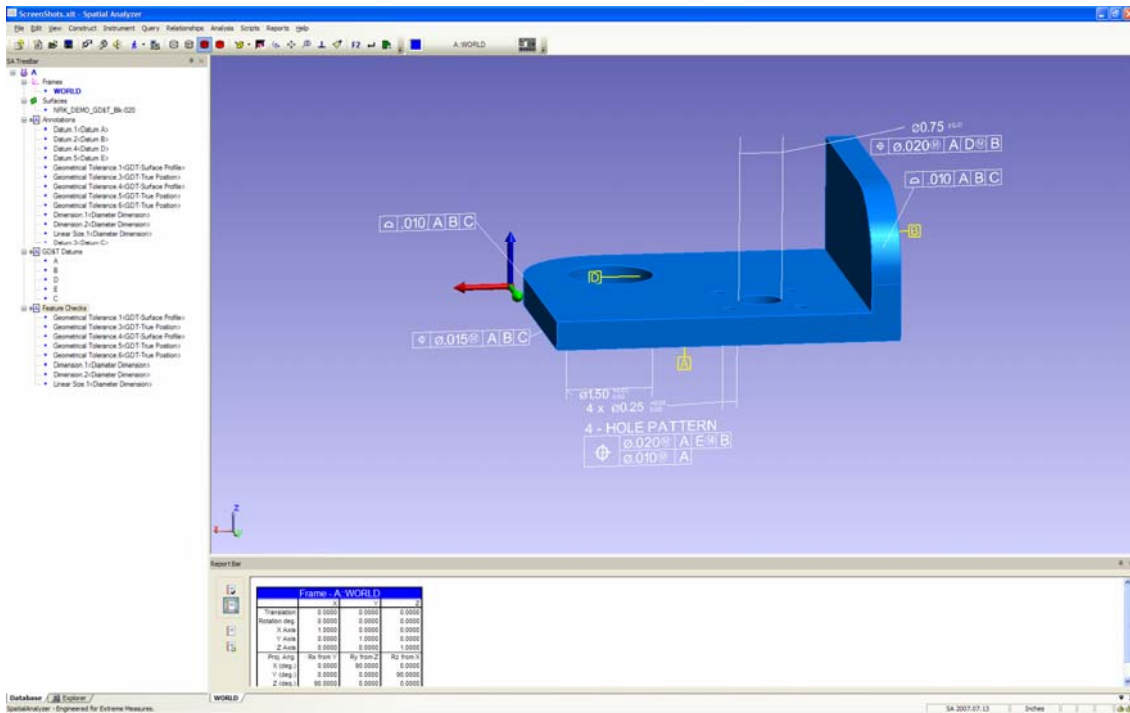
Once the file has loaded, annotations will appear both in the graphic window and in the SA Treebar.



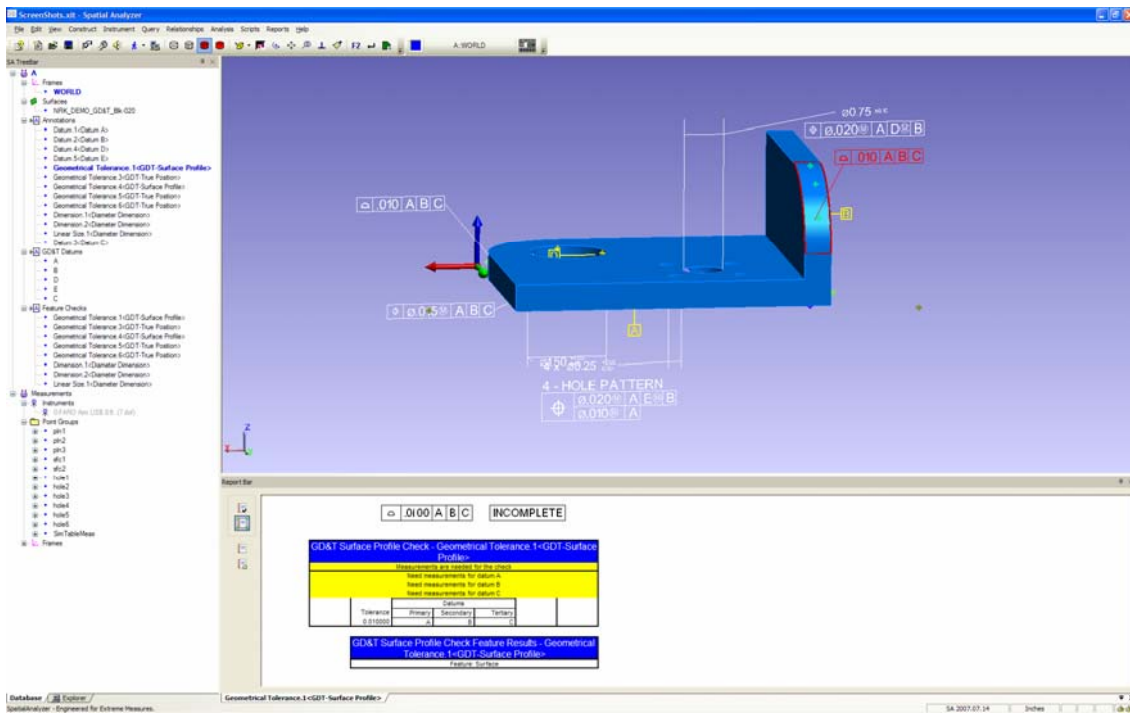
Expand the Annotations section heading to see the individual entries. Feature checks can be created one at a time by right clicking the individual entry, or all at once by right clicking the section heading.



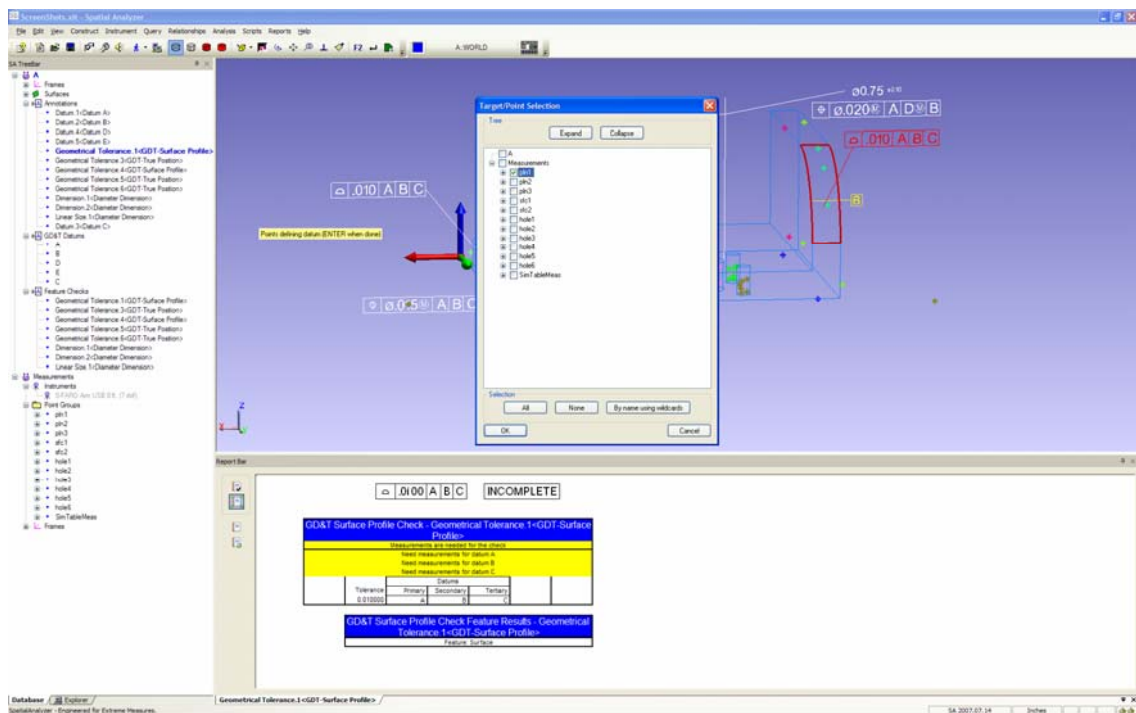
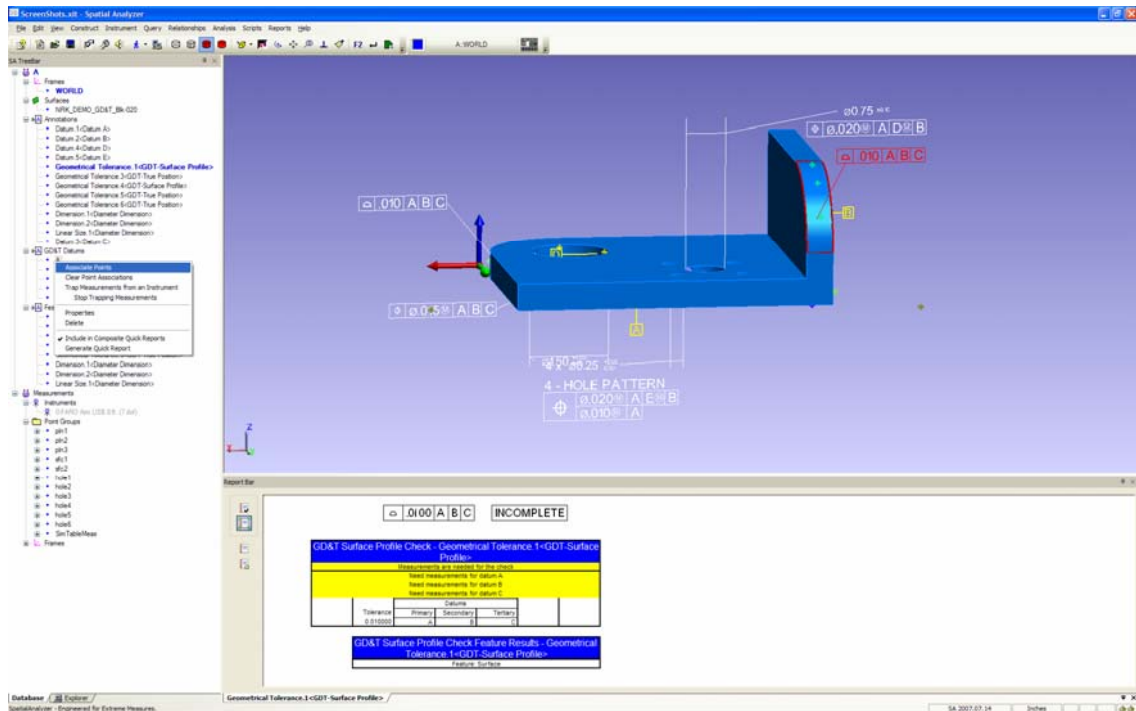
Notice that creating the feature check automatically creates the necessary GD&T Datums.



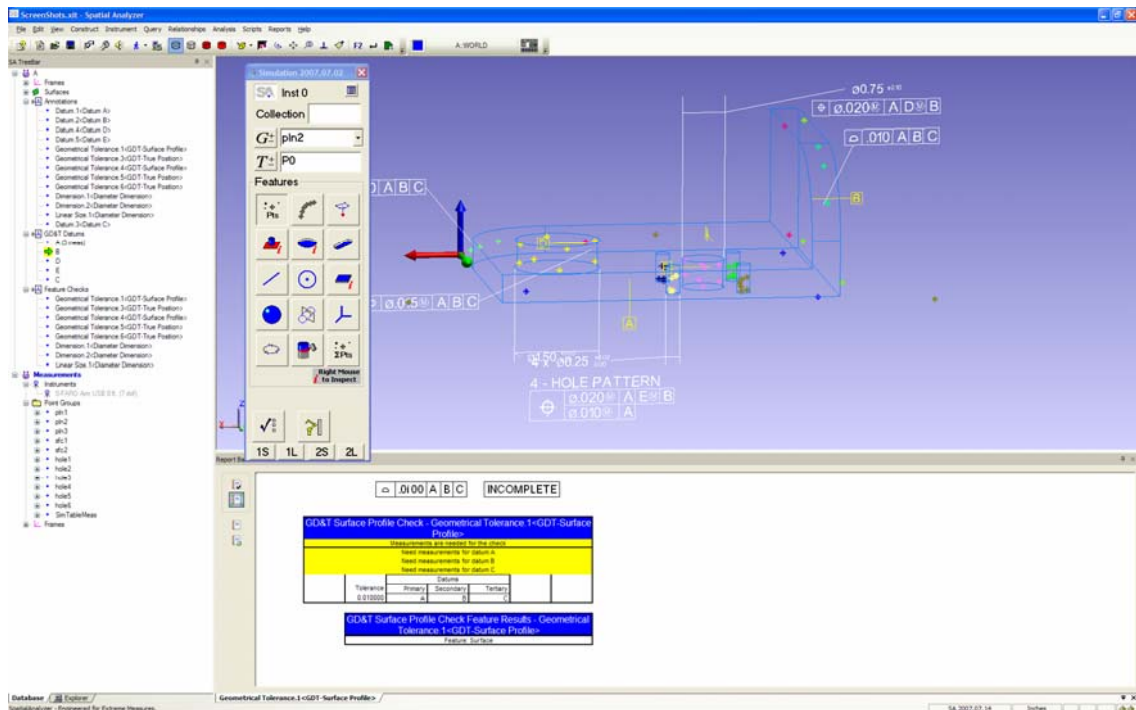
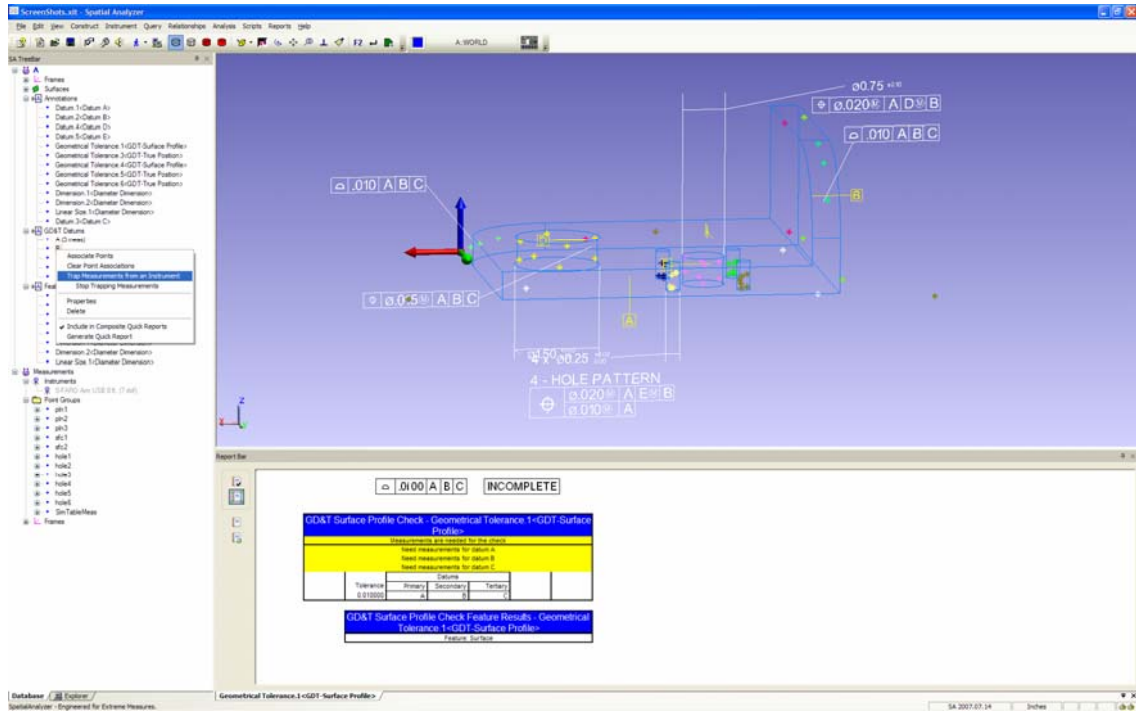
Clicking on a Feature Check causes the Report Bar to display the current report for the feature check, as it does for any other Treebar object. If insufficient measurements exist for evaluation of the feature check, it is noted at the top of the report. Highlighting the GD&T frame in the graphics window is accomplished by right clicking the annotation.



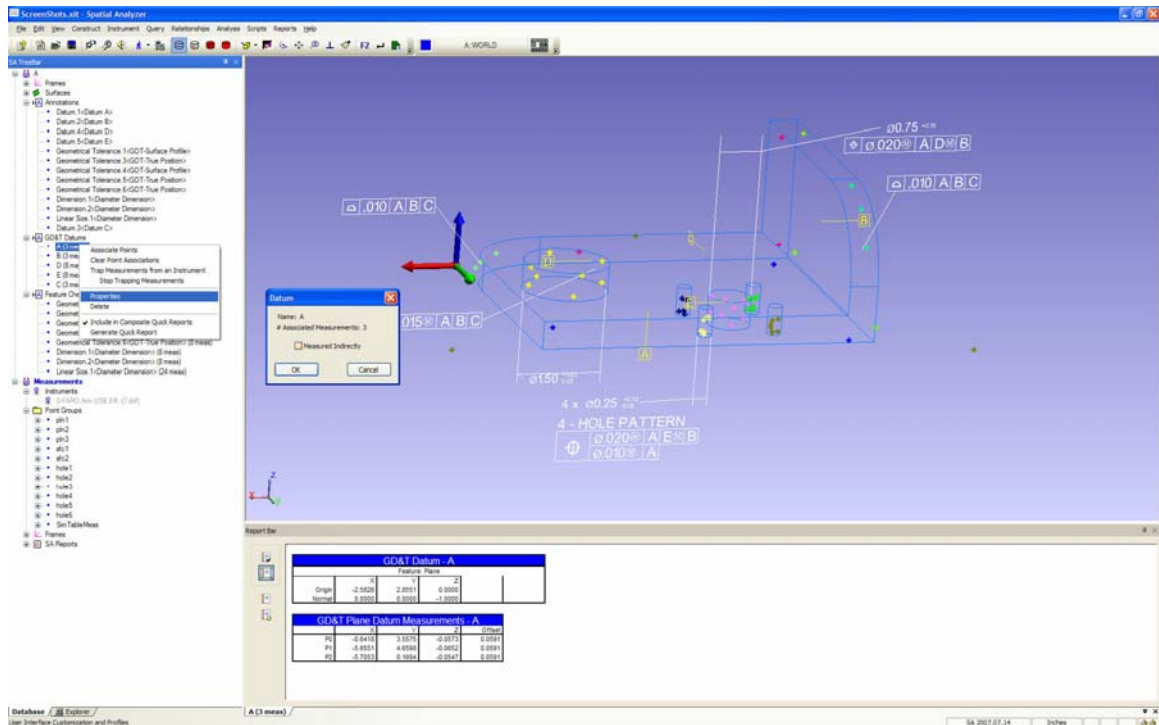
Measurements are linked to datums and feature checks by right clicking and selecting either Trap Measurements or Associate Points. If measurements have already been collected, choose associate points. The number of associated points is shown adjacent to the datum or feature check.



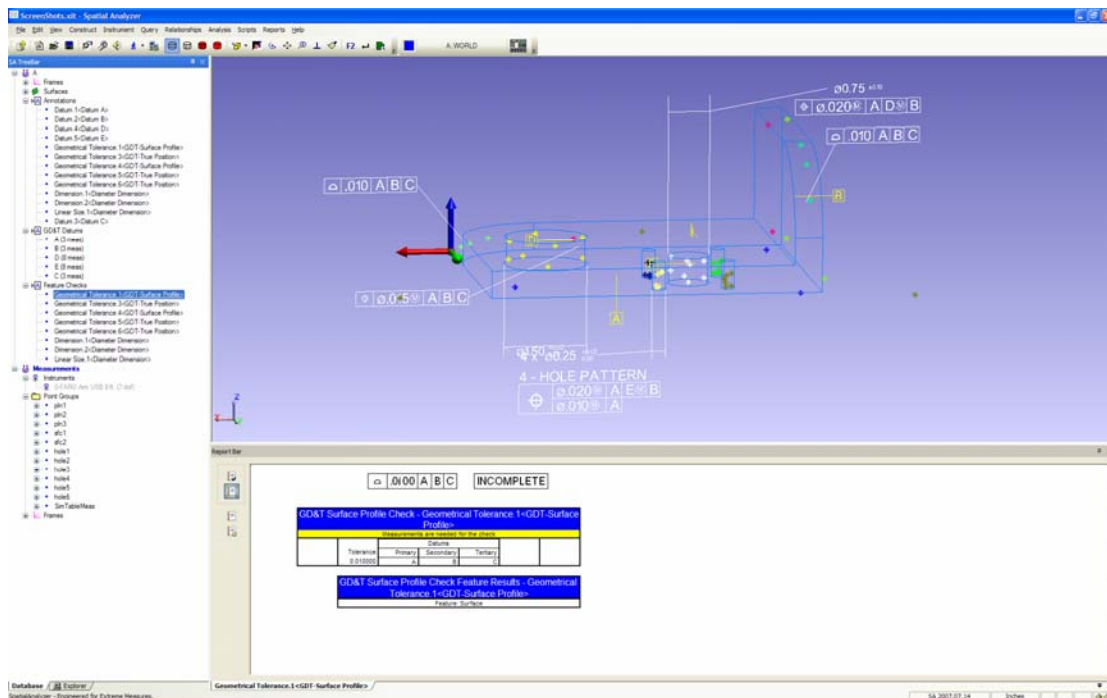
If measurements haven't been collected, choose Trap Measurements. All subsequent measurements will automatically be associated with the datum or feature check. Although the collection::group::target entered in the instrument interface will not impact correctly associating the measurements, appropriate naming conventions will facilitate subsequent analysis.



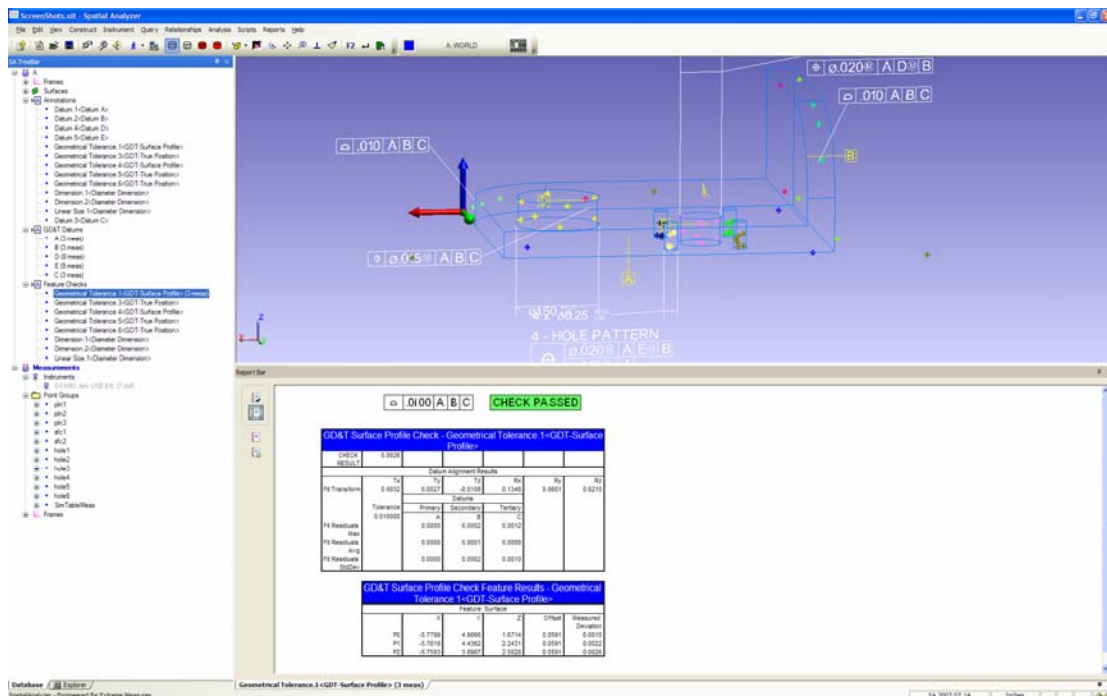
Recall that in SA, measurements relative to a surface must always be made on the side corresponding with the positive normal vector. Similar considerations exist in GD&T. Rather than considering the positive normal direction, point on positive side, or Right Hand Rule, an option exists to indicate that the measurements have been taken indirectly. The default is that all measurements are taken directly. As an example, consider a prismatic part with a datum resting on a surface plate. Frequently, the datum is indirectly measured by measuring (or assuming nominal values for) the surface plate. Indicate this in SA by selecting the Measured Indirectly property.



An example of a surface profile feature check is shown below. The note informs us that insufficient feature measurements exist. Since there isn't a note about the datums, we observe that they have been adequately defined.



Once sufficient measurements exist, the feature check may be evaluated. Passing or Failed status is shown immediately adjacent to the feature frame. Details follow, with the virtual transform required by the datum definition, followed by individual residuals.



Since each GD&T feature check defines its own orientation, it is often convenient to perform an alignment. (Perhaps solely for a graphical illustration of where the

The screenshot shows the SolidWorks CAD environment. The main window displays a 3D model of a mechanical part with various geometric features and tolerances. The model is blue and set against a light blue background. A coordinate system (X, Y, Z) is visible. The left sidebar shows the Feature Tree with various features like Datum A, Datum B, Datum C, and several GD&T Surface Profiles. The bottom right corner displays a 'CHECK PASSED' message and a table of results for the GD&T Surface Profile Check.

Feature Tree (Left Sidebar):

- Part
 - Surfaces
 - Annotations
 - Datum 1-Datum A
 - Datum 2-Datum B
 - Datum 3-Datum C
 - Datum 5-Datum E
 - Geometrical Tolerance 1-GDT Surface Profile
 - Geometrical Tolerance 3-GDT True Position
 - Geometrical Tolerance 4-GDT Surface Profile
 - Geometrical Tolerance 5-GDT True Position
 - Geometrical Tolerance 6-GDT True Position
 - Dimension 1-Diameter Dimension
 - Dimension 2-Diameter Dimension
 - Linear Slot 1-Diameter Dimension
 - Datum 3-Datum C
 - Feature Check
 - Geometrical Tolerance 1-GDT Surface Profile (2 meas)
 - Geometrical Tolerance 3-GDT True Position
 - Geometrical Tolerance 4-GDT Surface Profile
 - Geometrical Tolerance 5-GDT True Position
 - Geometrical Tolerance 6-GDT True Position
 - Dimension 1-Diameter Dimension
 - Dimension 2-Diameter Dimension
 - Linear Slot 1-Diameter Dimension
 - Measurements
 - Feature Check
 - Geometrical Tolerance 1-GDT Surface Profile (2 meas)
 - Geometrical Tolerance 3-GDT True Position
 - Geometrical Tolerance 4-GDT Surface Profile
 - Geometrical Tolerance 5-GDT True Position
 - Geometrical Tolerance 6-GDT True Position
 - Dimension 1-Diameter Dimension
 - Dimension 2-Diameter Dimension
 - Linear Slot 1-Diameter Dimension

Model Annotations:

- Feature 1: $\phi 0.020 \text{ M} \text{ A D B}$
- Feature 2: $\phi 0.10 \text{ A B C}$
- Feature 3: $\phi 0.075 \text{ A B C}$
- Feature 4: $\phi 0.015 \text{ A B C}$
- Feature 5: $\phi 0.020 \text{ M} \text{ A E B}$
- Feature 6: $\phi 0.010 \text{ M} \text{ A}$

Bottom Right Corner:

4 - HOLE PATTERN

$\phi 0.020 \text{ M} \text{ A E B}$

$\phi 0.010 \text{ M} \text{ A}$

Report Bar:

$\phi 0.00 \text{ A B C}$ **CHECK PASSED**

GD&T Surface Profile Check - Geometrical Tolerance 1-GDT Surface Profile

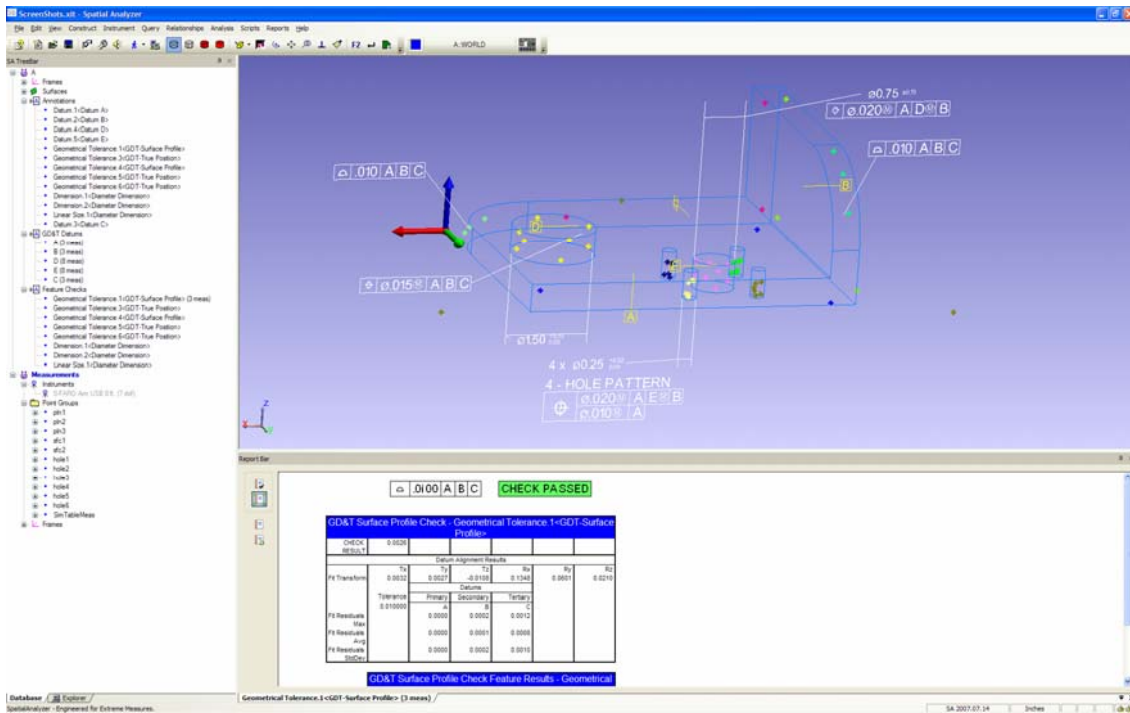
CHECK RESULTS	0.000	Datum Alignment Results					
	Ta	Tb	Tc	Ra	Rb	Rc	Rd
F1 Transform	0.0002	0.0002	-0.0003	0.1345	0.0001	0.0001	0.0001
F2 Results Max	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Database | Options |

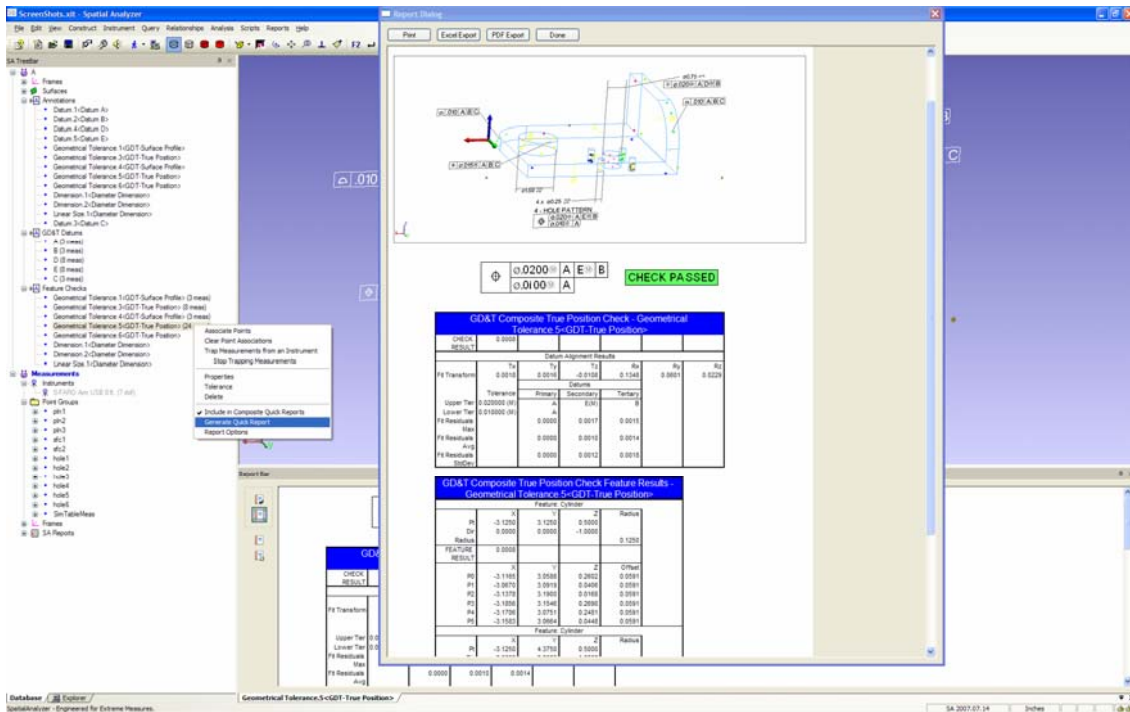
Geometrical Tolerance 1-GDT Surface Profile (2 meas)

SA 2007.07.14 **Index**

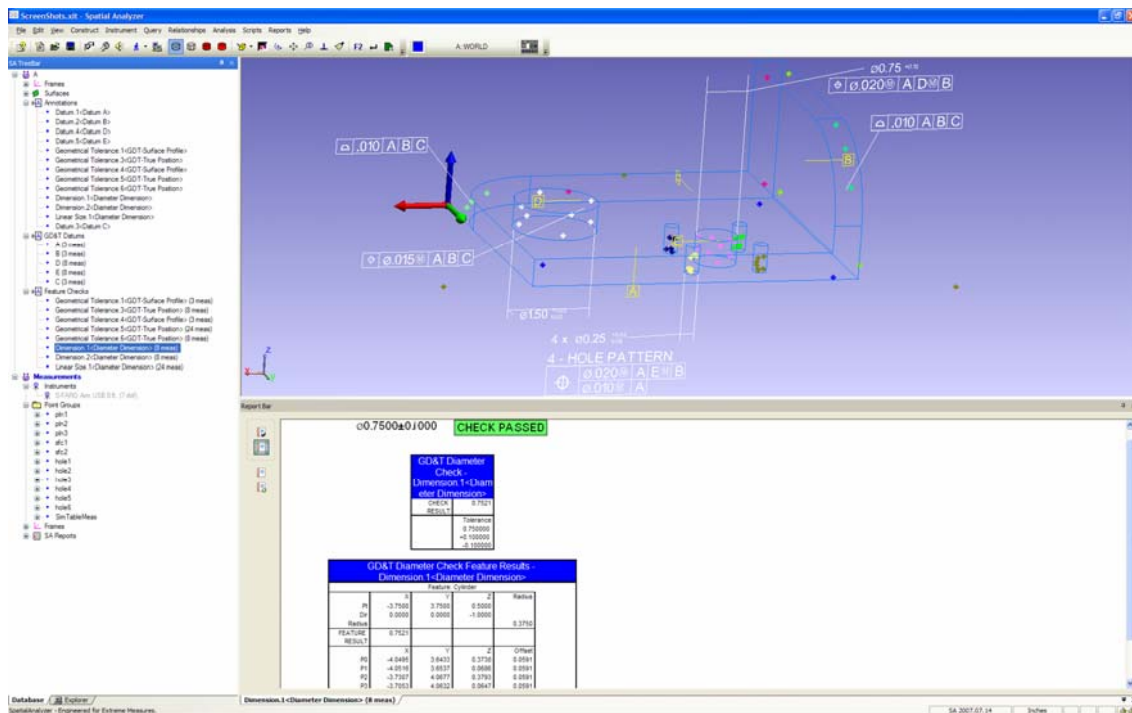




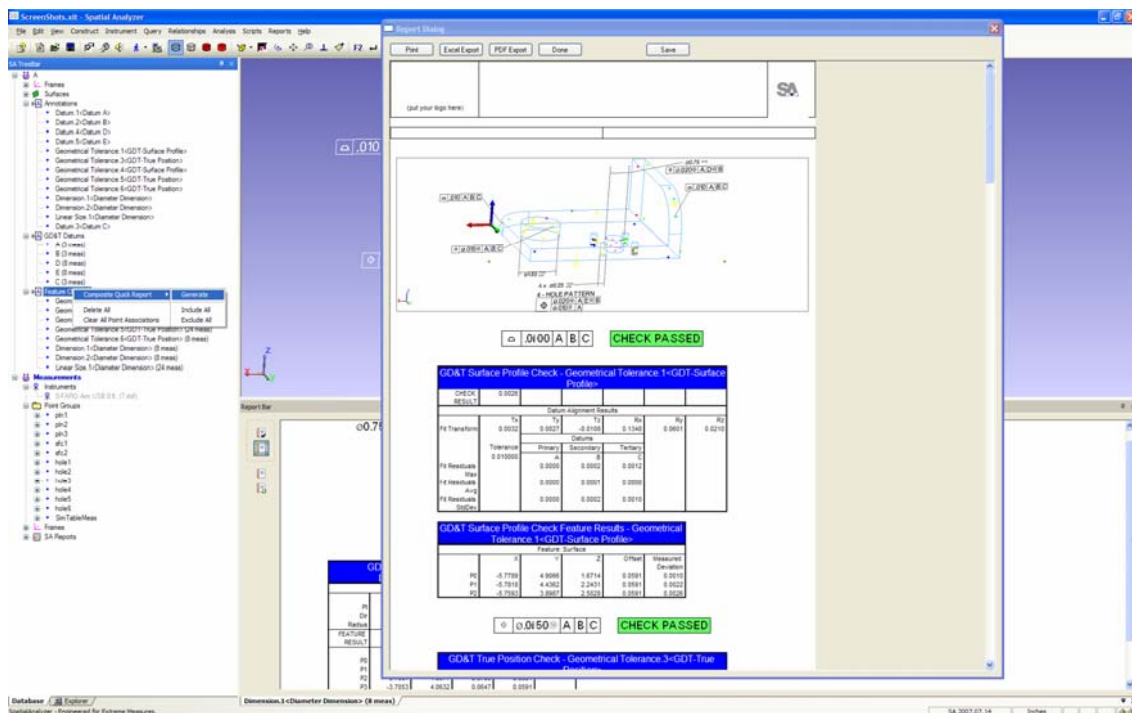
Here's an example of a compound multiple feature position check. Similar to the surface profile, test status is adjacent to the frame, followed by fit residuals and finally the feature check residuals. In this case, the feature residuals occur multiple times.



Here's an example of a diameter check.



To get an overall report of all the GD&T feature checks, right click on the section heading and generate a composite quick report.



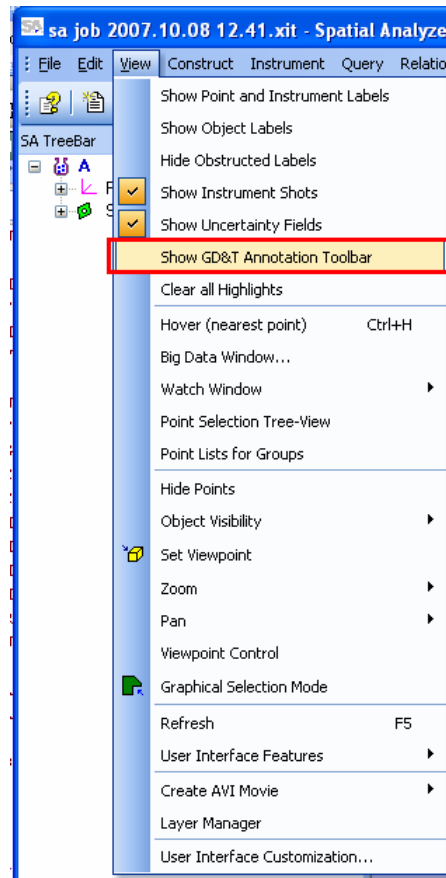
GD&T Toolbar

Overview

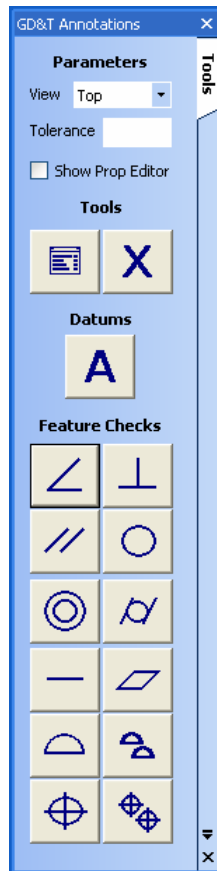
This feature builds on the recently added ability to manually create GD&T annotations. The toolbar allows you to quickly add datum and feature check annotations to a job as well as edit and delete them.

Getting Started

Start Spatial Analyzer, and open the undecorated sample part job file (GDTToolbar.xit). You must be running SA v2007.10.09 or newer to continue. From the menu, select View>>Show GD&T Annotation Toolbar.



The GD&T Toolbar will appear docked to the side of the main SA window.

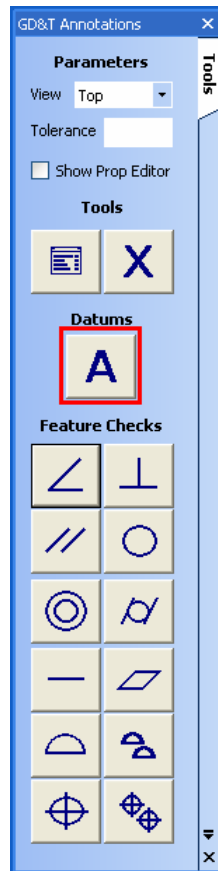


Select one of the pre-set views from the View pull-down. This selection controls the orientation used when creating the annotations. You can also enter a tolerance value which is used when creating feature checks.

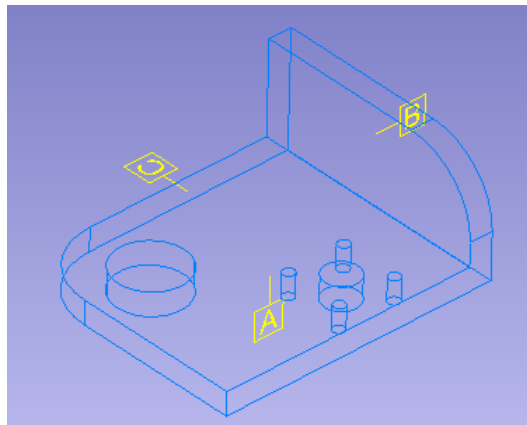
By default, when you use the toolbar to create a datum or feature check annotation, the annotation is created using default parameters and the values you enter in the Parameters area of the toolbar. If you'd like to customize the annotation, you can check the "Show Prop Editor". If checked, then after creating the annotation, the properties window will be displayed allowing further editing of the annotation.

Creating Datum Annotations

Click on the Datum button to initiate datum annotation creation.

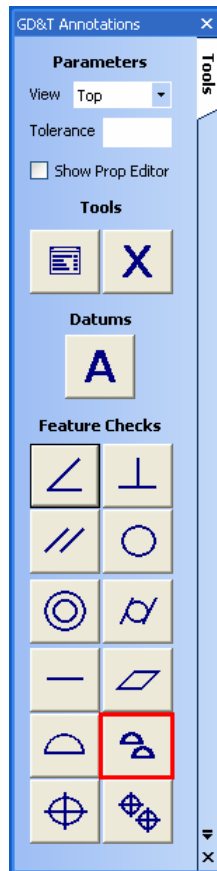


This will begin a selection mode in SA, prompting you to select the feature for the new datum annotation. Double-click a face on the imported part and it should highlight. Now hit ENTER. The datum name will automatically be chosen (A..Z) based on which datums are already in the job. Repeat this to add several datums.



Creating Feature Check Annotations

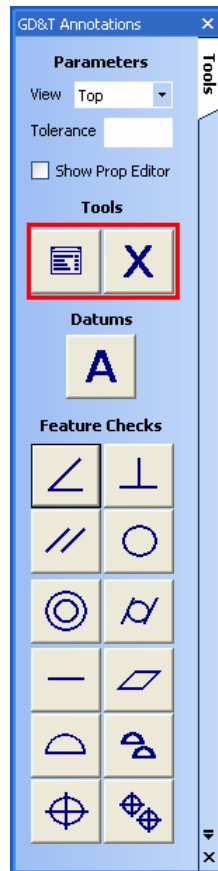
Click on the surface profile button to initiate feature check annotation creation.



Select one of the surfaces and hit ENTER. This will create the feature check annotation.

Editing/Deleting Annotations

Using the Edit and Delete buttons you can select annotations to edit (property window is displayed) or delete.



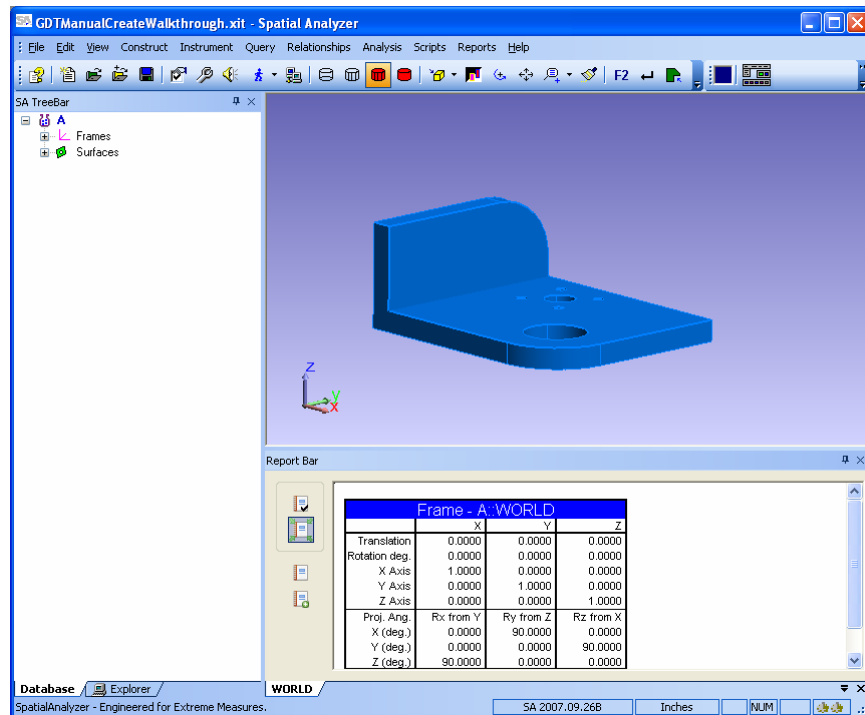
GD&T Manual Creation

Overview

This feature provides the user with the capability of creating GD&T annotations for datums and feature checks, removing the need to have the annotations already present in the CAD part file. Previously, it was necessary to have a CAD part file (specifically CATIA V5) which already had GD&T annotations in order to work with the GD&T features in SA. With this new functionality, users can import a CAD part and decorate it themselves with GD&T datum and feature check annotations. This guide walks through a sample CAD part showing how to decorate that part with GD&T annotations and then use the newly created annotations.

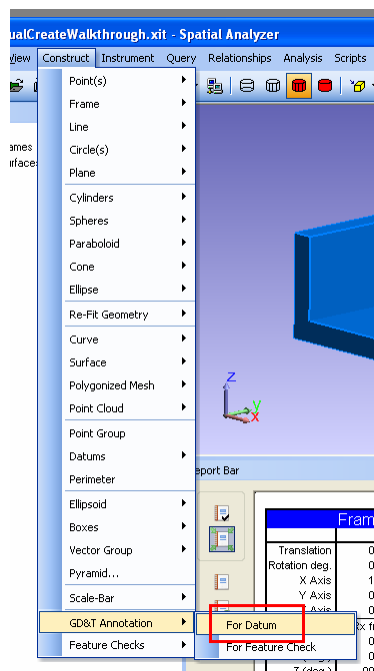
Getting Started

Start Spatial Analyzer, and open the undecorated sample part job file (GDTManualCreateWalkthrough.xit). You must be running SA v2007.09.27 or newer to continue. Switch to solid shading with edges display and you should see the below.



Creating GD&T Datum Annotations

Select “Construct>>GD&T Annotation>>For Datum” from the menu.



This will open the Datum Annotation Editor shown below. Fill in the Name as shown below. This will be Datum A.

GD&T Datum - A

Properties

(Identifier)

Name: A

CAD Feature

CAD Feature:

Placement

View: <current view>

TextHeight: 0.4

Name

The name of the datum

Create Annotation

It is now necessary to choose a feature in the part with which this datum will be associated. A feature may contain multiple faces (in the case of a cylinder with the 2 halves of the shape). The feature type you select will determine the datum type which eventually gets created. For this example, we are interested in creating a planar datum, so we will select a single planar face.

To select the feature for this datum annotation, select the “CAD Feature” field and you should see a button with “...” appear. Click this button.

GD&T Datum - A

Properties

(Identifier)

Name: A

CAD Feature

CAD Feature: ...

Placement

View: <current view>

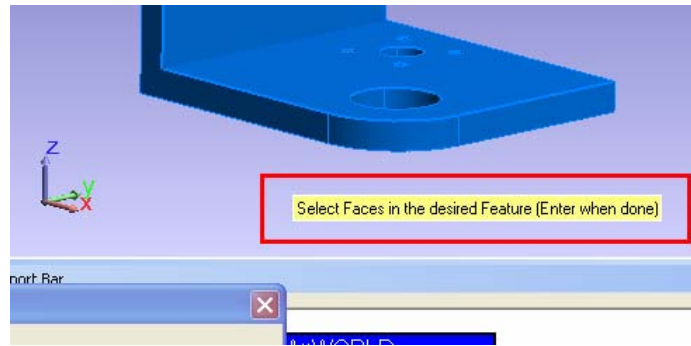
TextHeight: 0.4

CAD Feature

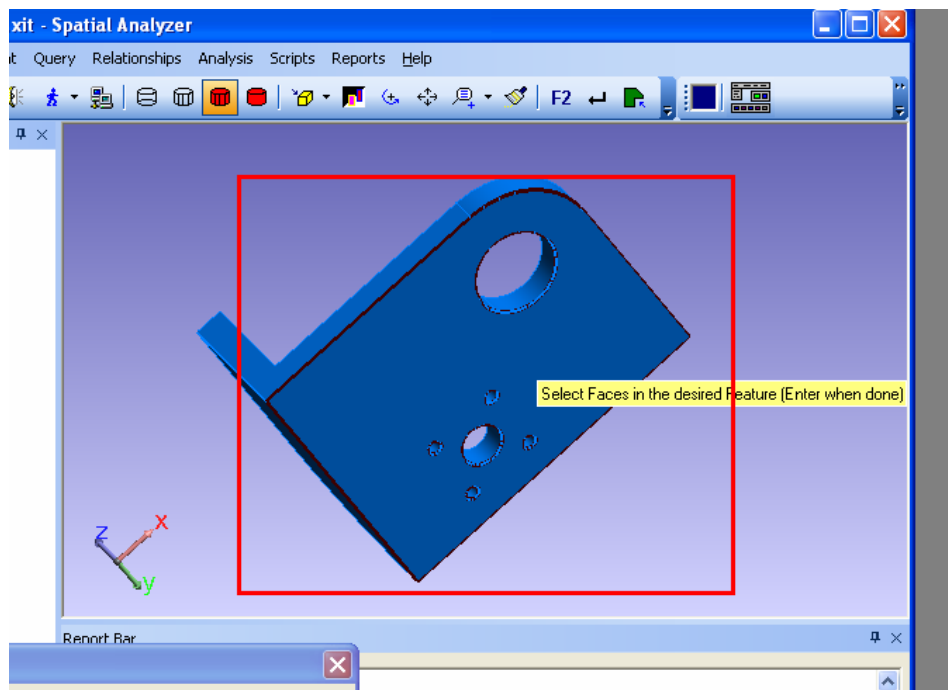
CAD Feature to Associate with this GD&T Datum

Create Annotation

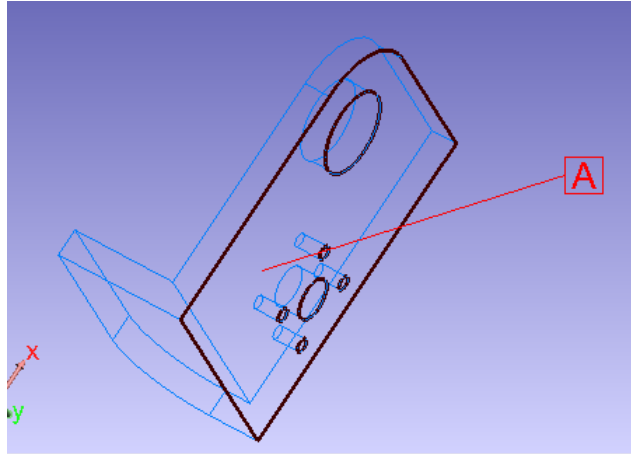
This will initiate an SA face selection mode as shown below. In this mode, you will need to double-click on the desired faces and when done hit the ENTER key.



Rotate the part around so you can see the bottom face and double-click on it. The face will highlight to confirm your selection. Since we're only selecting a single planar face for this datum, hit ENTER to complete the selection.

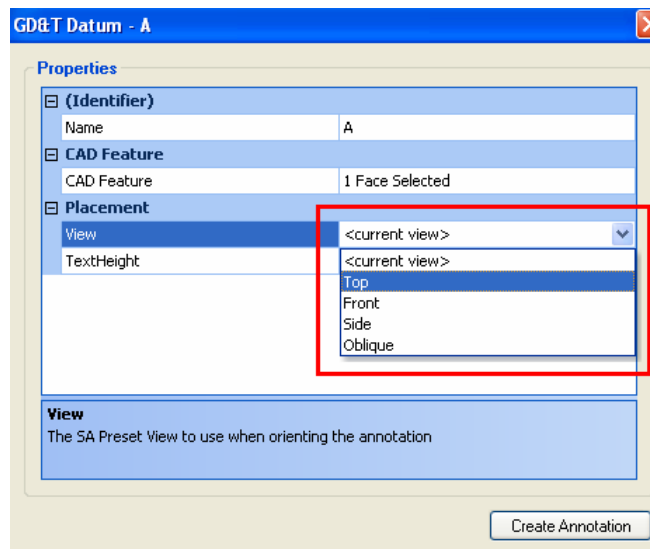


Switch to wireframe mode and you'll be able to see a rendering of the datum annotation. You can click and drag it around. While dragging it, the associated faces will highlight as a reminder of which faces the annotation goes with.



At this point, the annotation will render parallel to the screen. This allows you to rotate the view around while still maintaining a clear view of the annotation. Before creating the annotation, you'll need to pick one of the SA preset views to orient the annotation.

In the Datum Editor, use the View field to pick one of the SA Preset views.



The Preset View choice controls the rotation of the annotation. When picking a Preset View the annotation is rotated to be parallel to that view and with the screen when that view is selected.

The position of the annotation is controlled by dragging it around with the mouse. When you drag an annotation it moves in its plane (determined by the selected Preset View).

In addition to orienting and positioning the annotation, you can also customize the text size of the annotation using the TextHeight field as shown below.

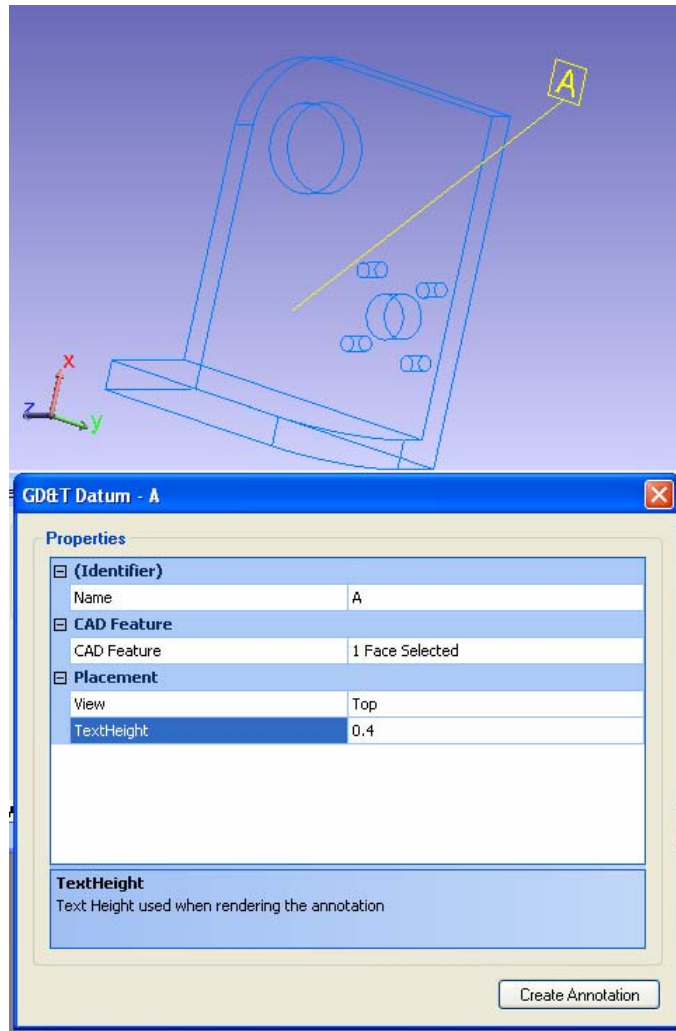
The image shows a software dialog box titled "GD&T Datum - A". It contains a "Properties" section with three expandable categories: "(Identifier)", "CAD Feature", and "Placement". Under "(Identifier)", the "Name" is set to "A". Under "CAD Feature", the "CAD Feature" is set to "1 Face Selected". Under "Placement", the "View" is set to "Top". The "TextHeight" field, located below the "View" field, is highlighted with a red rectangle and contains the value "0.4". Below the "TextHeight" field, there is a section titled "TextHeight" with the description "Text Height used when rendering the annotation". At the bottom right of the dialog box is a button labeled "Create Annotation".

Properties	
(Identifier)	
Name	A
CAD Feature	
CAD Feature	1 Face Selected
Placement	
View	Top
TextHeight	0.4

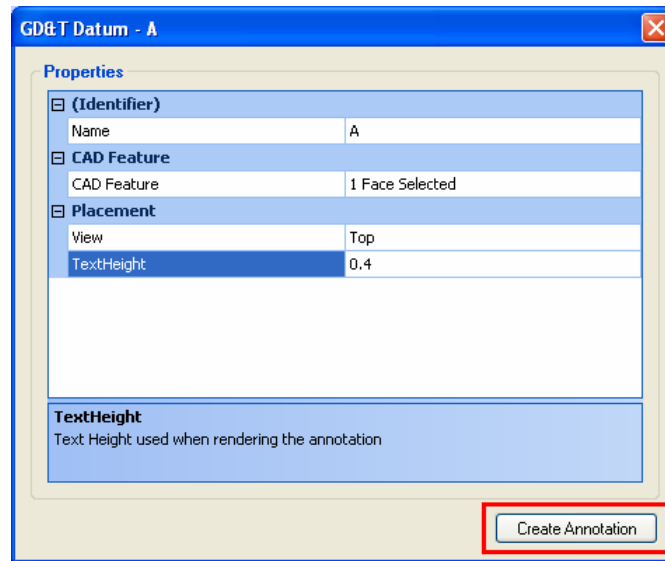
TextHeight
Text Height used when rendering the annotation

Create Annotation

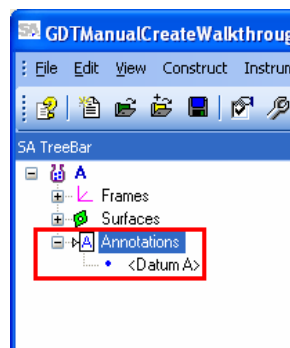
When finished the annotation should look similar to the below (assuming top view was chosen).



To complete the process, click the “Create Annotation” button.

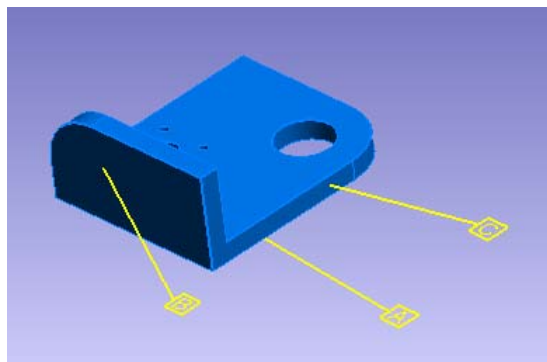


A datum annotation should be added to the job and the editor window should close.



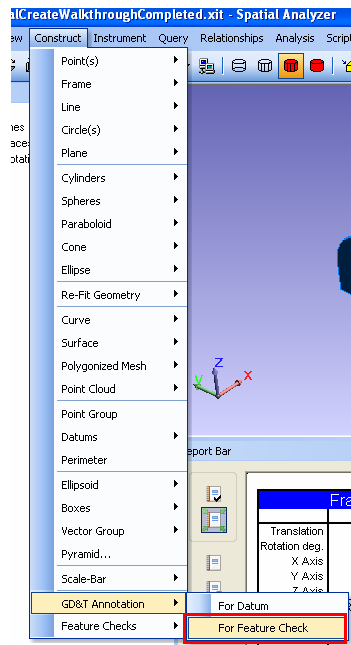
Repeat this process and create two more planar datums B and C. Datum B should be associated with the large side face and Datum C should be associated with the thin back edge face.

When finished, you should have something which looks similar to the below.



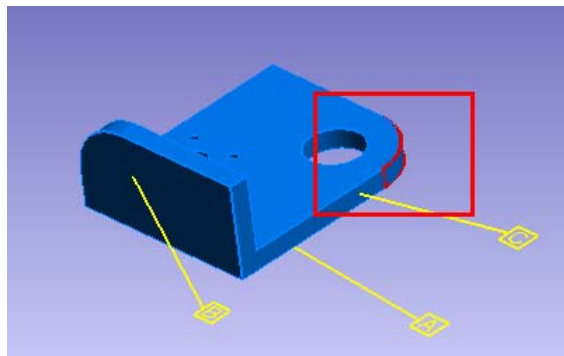
Creating a GD&T Feature Check Annotation

Select “Construct>>GD&T Annotation>>For Feature Check” from the menu.

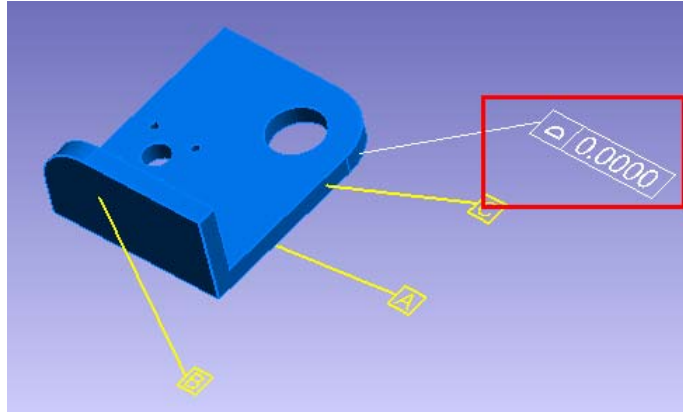


This will open the GD&T Feature Check Annotation Editor. Select “Surface Profile” using the drop-down in the Type field. Enter “SampleSurfProfileCheck” in the Name field.

Feature Checks can be associated with multiple features (for example, a true position which is associated with 4 hole features (each having 2 faces)). For now, enter 1 in the “# CAD Features” field and a new field, “CAD Feature #1” will be added. Use this field just as with the datum annotation creation to pick the faces which make up the desired feature in the CAD part. For this walkthrough, select the corner surface face adjacent to Datum C as shown in the below screenshot.



Once the feature is selected, the annotation will render in the 3D view (it also renders at the top of the window in 2D). You can now drag it around, choose the preset view for orientation, and modify the text height. Pick the top Preset view again so it ends up in the same plane as the datum annotations.



Now we'll need to specify datums for the feature check. Enter 1 in the “# Primary Datums”, “# Secondary Datums”, and “# Tertiary Datums” fields. As you do this, additional fields will be added allowing you to choose the individual datums. Choose “A” as the primary datum, “B” as the secondary datum, and “C” as the tertiary datum.

Now Enter a tolerance value. When done, the properties should look similar to the below (the tolerance value is scrolled off the bottom in the below screenshot but should just be whatever value is desired).

GD&T Annotation - SampleSurfProfileCheck

Annotation

⏏ 3.1416 A B C

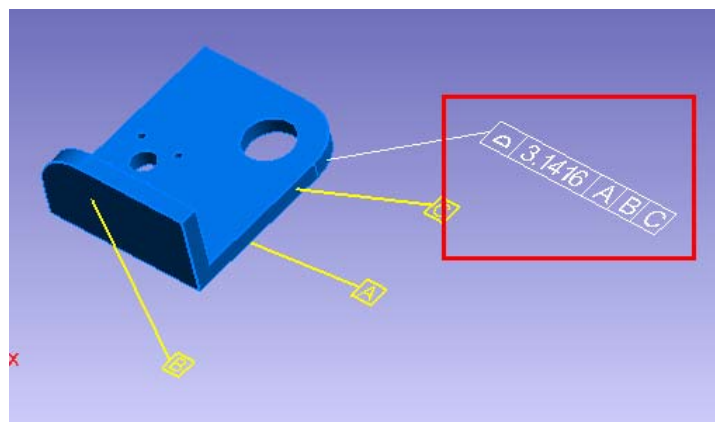
Properties

Type	Surface Profile
Features	
# CAD Features	1
CAD Feature #1	1 Face Selected
Placement	
View	Top
TextHeight	12.000
Datums	
# Primary Datums	1
# Secondary Datums	1
# Tertiary Datums	1
Primary	
Datum 1	
Datum	A
MaterialModifier	NoMCM
Secondary	
Datum 1	
Datum	B
MaterialModifier	NoMCM
Tertiary	
Datum 1	
Datum	C
MaterialModifier	NoMCM
Tolerance	
Tolerance	3.14159

Tolerance
Tolerance value

? Create Annotation

When finished the annotation in the 3D view should match the properties entered in the editor.



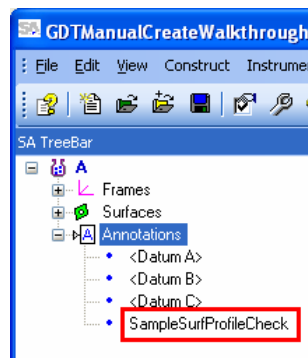
Click the “Create Annotation” button.

Properties	
Type	Surface Profile
Features	
# CAD Features	1
CAD Feature #1	1 Face Selected
Placement	
View	Top
TextHeight	12.000
Datums	
# Primary Datums	1
# Secondary Datums	1
# Tertiary Datums	1
Primary	
Datum 1	
Datum	A
MaterialModifier	NoMCM
Secondary	
Datum 1	
Datum	B
MaterialModifier	NoMCM
Tertiary	
Datum 1	
Datum	C
MaterialModifier	NoMCM
Tolerance	
Tolerance	3.14159

Tolerance
Tolerance value

Buttons: ? (Help), Create Annotation

An annotation should be added to the job and the editor window should close.



Using the GD&T Annotations

Now that the datum and feature check annotations are created, they can be used just as though they had been imported with the CAD part.

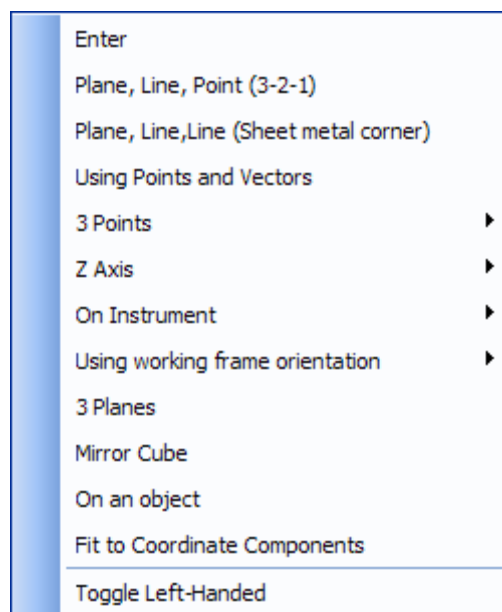
COORDINATE FRAMES AND SPATIAL TRANSFORMATIONS

SpatialAnalyzer will allow you to create any number of coordinate frames or systems. Each coordinate frame may be selected as the working frame. When a coordinate frame is designated as the working frame, it will be drawn in bold and the toolbar will display the name of the frame. While that coordinate frame is designated as the working frame, all coordinates will be represented relative to it. This allows you to take measurements in any arbitrary default coordinate system and represent them in a system that better describes the data.

All reporting functions will list coordinates and geometrical objects relative to the working coordinate frame. Coordinate uncertainties will also be presented in the working frame.

Creating New Coordinate Frames

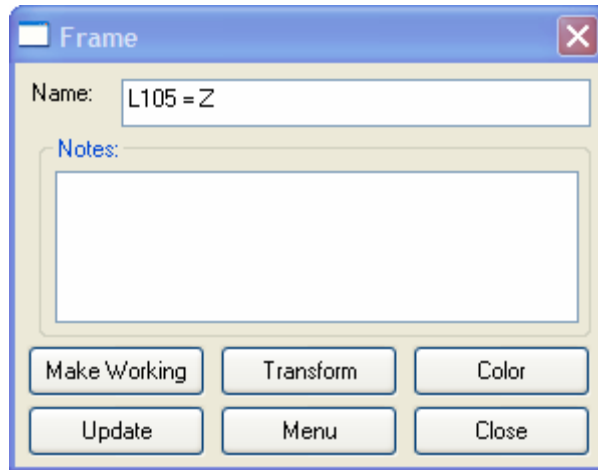
From the Construct menu, select the Frame option. There will be a list of available frame definition methods under that menu. Once a particular method is specified, a cursor prompt will appear to request the next action. The following sections will highlight the different frame construction techniques.



Frame Construction Methods

Manual Entry of Coordinate Frame Transformation (Enter)

If you select the “Enter” option under the construct frames menu, the frame dialog will appear. Assign the frame a name, and then press the Transform button. You will be able to move the newly created frame (relative to the current working frame) using the transformation dialog.



Frame Properties Dialog

Each object in **SpatialAnalyzer** has a field for notes. The properties page for each object allows you to enter in helpful information in this note field. As an example, this frame's note field states that it is to be used as a local construction frame to make the building process easier for a particular detail.



Transform Dialog

The transformation dialog will allow you to change both the position and orientation of the coordinate frame using a variety of conventions. For position, you can work in Cartesian coordinates, cylindrical coordinates, or spherical coordinates. For orientation you can choose Fixed XYZ, Euler ZYZ, or a variety of other orientation conventions. Please note that the coordinate frame's initial transformation will place it coincident with the current working frame. This means that you will need to move the newly created frame in order to distinguish it visually from the working coordinate frame in the graphics view.

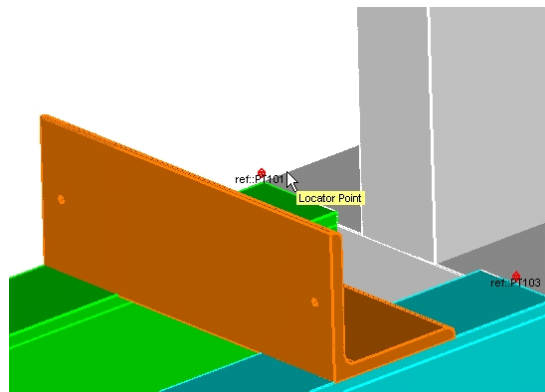
Position dialogs just like that shown in the figure above are available on the properties page for any of the objects in **SpatialAnalyzer**. If you need to adjust the position or orientation of an object, the process is consistent for all objects and is capable of making the fine-tuning adjustments you require.

Plane, Line, Point (3-2-1) Method

The Plane, Line, Point (3-2-1) frame construction method is used just like the name suggests. When you need to construct a frame from a point, line, and then a plane, **SpatialAnalyzer** helps by letting you complete the entire process graphically with Plane, Line, Point (3-2-1) function. It efficiently prompts you (with mouse prompts) for each piece of information in a consistent procedure.

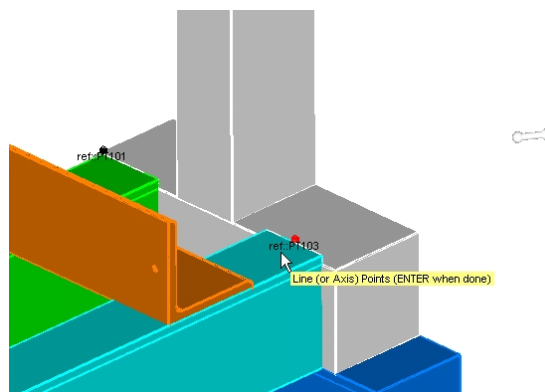
You can change the graphics view's perspective (e.g., zoom, rotate, translate etc.) at any point during the procedure. This makes selecting the right elements much easier because you can zoom in to pick the right point on one end of the object. Then zoom back out, translate and rotate the view so that you're able to see the other end. Zoom in to pick the next point.

The first piece of information this function needs is the Locator Point. The point you select will become the origin of frame. The mouse prompt asks you to select the Locator Point.



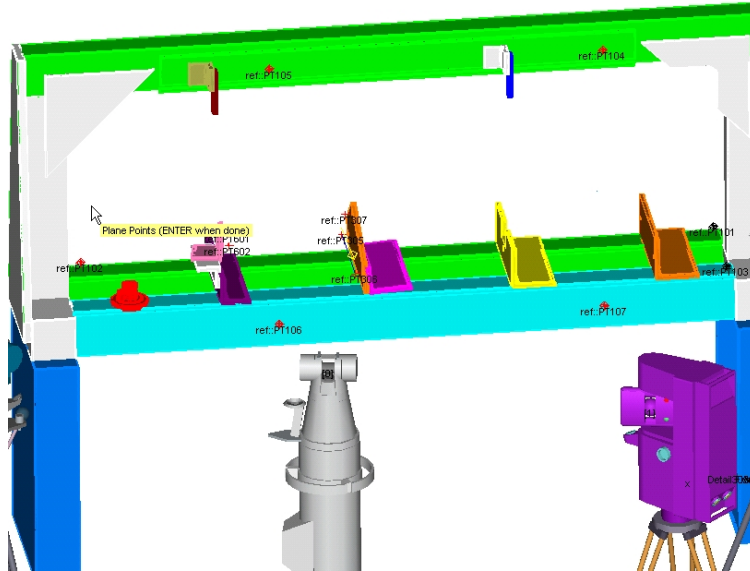
Locator Point step in the Plane, Line, Point (3-2-1) frame construction

After selecting the Locator Points the mouse prompt asks you to select the Line (or Axis) points. The procedure needs two points to define the axis direction. By selecting at least two points that define a particular direction you've completed the next step. Note, you can select a series of points for the direction, and **SpatialAnalyzer** will automatically best fit a line to these points, then use the line as direction for the axes.



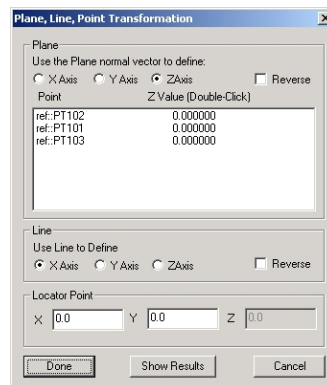
Line (or Axis) Points step in the Plane, Line, Point (3-2-1) frame construction

The next step in the procedure entails you selecting points and/or targets that define a plane that is normal to one of the frames coordinate axes.



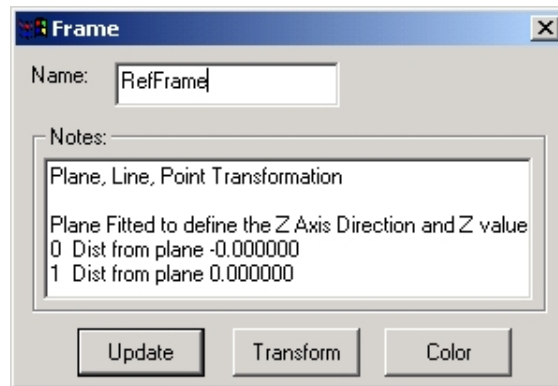
Plane Points step in the Plane, Line, Point (3-2-1) frame construction

This step follows the same process as Line selection process, the mouse prompt is asking you to select all the points that define a plane for the frame. You are again able to select more than the minimum of 3-plane points. If you do, **SpatialAnalyzer** will automatically best fit a plane through the points, then use that to help define the frame. After fitting the line and plane, the procedure shows you the fit results, and offers you a chance to change the frame parameters for the plane, line, and point.



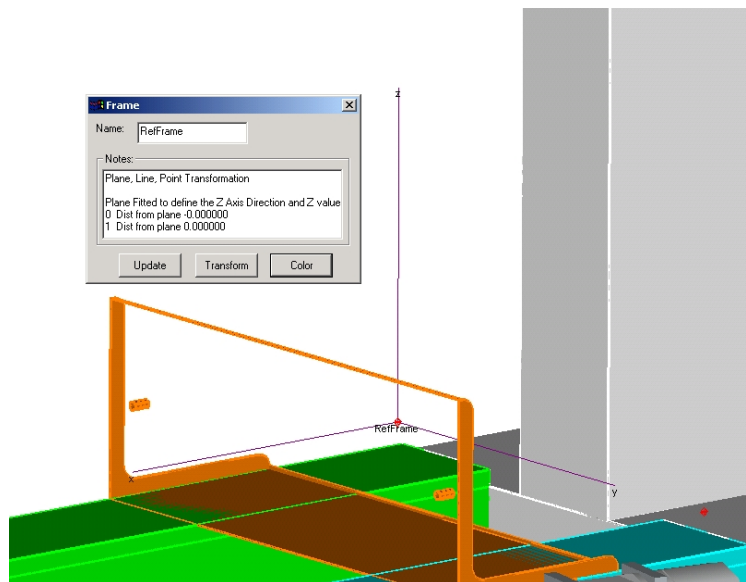
Plane fit and axes association step in the Plane, Line, Point (3-2-1) frame construction

After making your adjustments and hitting the Done button, **SpatialAnalyzer** creates a frame object in the graphics view to your specifications and then shows you its properties dialog.



Plane, Line, Point (3-2-1) frame's properties page

The properties dialog allows you to name the frame, assign it a different color, view and edit the transformation, and edit its note field. **SpatialAnalyzer** automatically puts the frame's parameters and fit results into the note field. So, if you need to refer back to find out how a particular frame was created or what the fit results were; you have it in the frame's note field. A frame's note field is presented on the frame's Quick Report in **SpatialAnalyzer**.



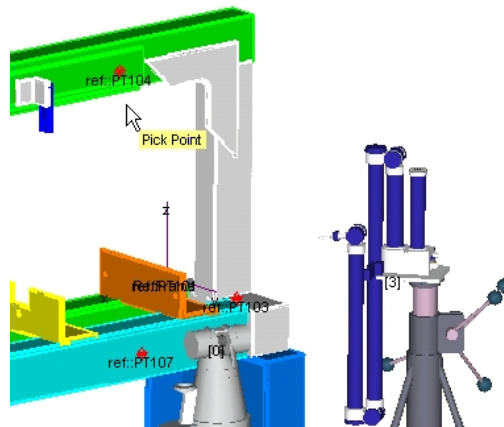
Graphics view of the Plane, Line, Point (3-2-1) frame

The figure shows the graphics view of the frame created using the Plane, Line, Point (3-2-1) function.

Using Points and Vectors Method

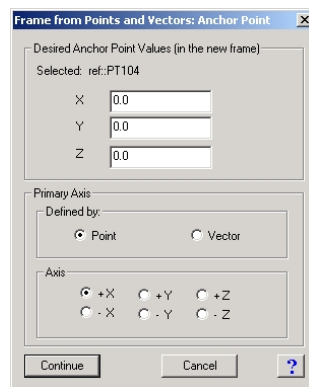
The Using Points and Vectors frame construction method helps you to piece together a frame from points and vectors. The graphics environment in **SpatialAnalyzer** makes this process fairly intuitive. Mouse prompts are provided for each step in the procedure and a dialog at the end allows you to make adjustments to each of the frame's parameters. The process concludes with the frames creation in the graphics view and a properties page that allows you to change each of the frames properties.

The first step is to pick an anchor point for the frame. The figure below shows the mouse prompt for this step.



Pick point (Anchor) in Using Points and Vectors frame construction

The process then asks you to define the anchor point's coordinates and how you want to define the frame's Primary Axis. The method supports both point and vector mechanisms for orienting the primary axis. The last group of controls is a set of Windows radio buttons. These radio buttons allow you to indicate which axis the vector will define. You can only select one of the six possible choices.

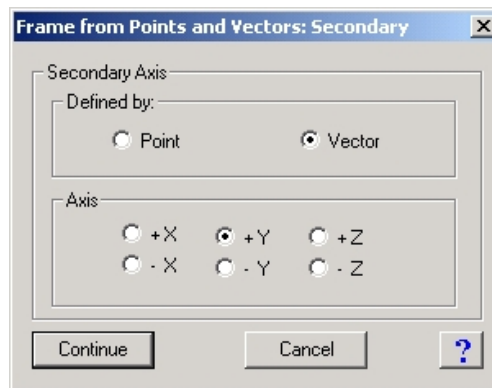


Frame from Using Points and Vectors, Anchor Point dialog

If you chose a point as the means for defining the primary axis in the Anchor Point dialog; the vector direction from the anchor point to this next point is used to define

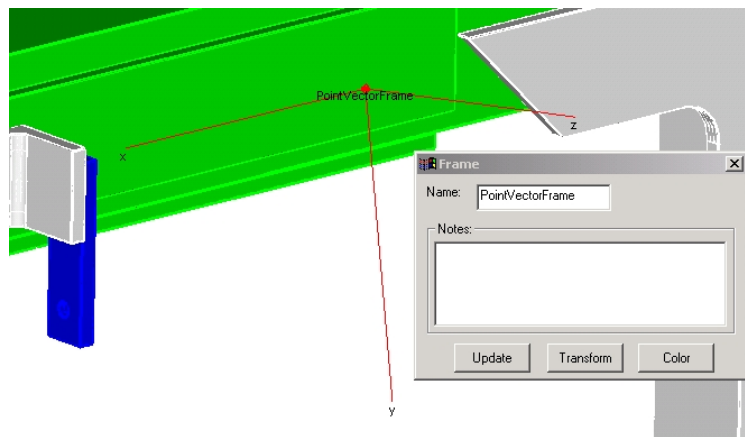
frame's primary axis. If you chose a vector to define the primary axis direction; then the primary axis will be defined by selecting any line in the database.

With the anchor point and primary axis defined, the next step is to define the secondary axis. The figure shows the dialog that allows you to choose the Secondary Axis definition method. If you want the secondary axis to be in the same plane as one particular point then choose the Point radio button. However, if you have a vector or line that has to be parallel with the secondary axis this method allows you to control that by choosing the Vector radio button.



Frame from Using Points and Vectors, Secondary Axis dialog.

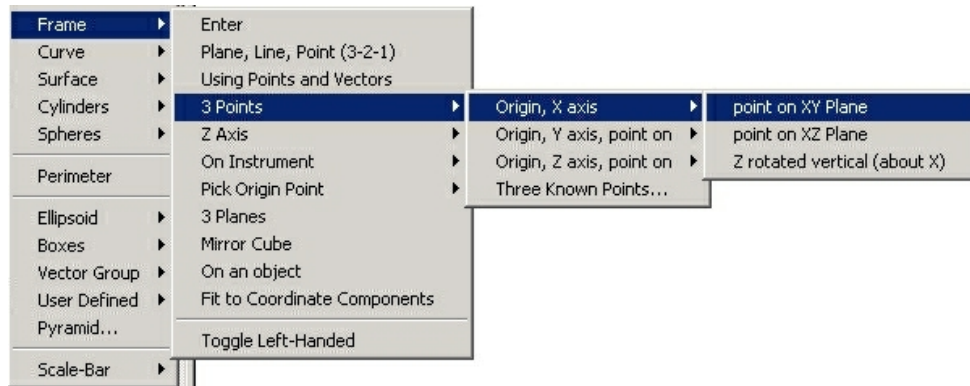
The lower radio buttons on the Secondary Axis dialog allow you to select which axis you want to define with the point or vector. Make your selection and then hit the continue button. After selecting the point or vector, the frame is created and its properties page is displayed.



Graphics view of the Using Points and Vectors frame.

Three Point Methods

There are a number of 3-Points methods for frame assignment. These methods are organized in the 3-Points menu selection.



Menu for constructing frames for directly form 3-points.

Each of the methods is discussed below:

Origin, X Axis

- Point on XY Plane
- Point on XZ Plane
- Z rotated vertical (about X)

Origin, Y Axis

- YX Plane
- YZ Plane

Origin, Z Axis

- ZX Plane
- ZY Plane

Three Known Points

The process for each of the methods above is basically the same, except for the Three Known Points method, which uses a different procedure to accomplish the same results. The following description applies to the first seven methods. The Three Known Points method is described in the section that immediately follows this one.

For any of the menu options for 3-Points frame construction beginning with “Origin,” the first menu level brings up a pop-up sub-menu. From this menu, choose the option that is appropriate for your application (e.g., Point on XY Plane). Mouse prompts ask you to select the Origin Point for the new frame. Select the point that you want at the origin of the coordinate frame. The next prompt (e.g., Point along the X axis) asks you to select the point that defines the secondary axis direction (relative to the origin point). After you select the second point, a final prompt (e.g., Point on XY plane) asks you to select a point that you want to lie on one of the frame’s coordinate planes.

A slight exception to this process is the Z rotated vertical (about X) method. This is a convenient frame assignment method for accomplishing several traditional alignment tasks. You are prompted to select a frame origin, then a point along the X-axis. The frame is immediately placed in the workspace so the origin is on the origin point and the X-axis passes through the X-axis point. The Z-axis is rotated so that it is most nearly aligned with the Z-axis of the current working coordinate frame (i.e., vertical).

After the frame is placed in the workspace, its property dialog appears. You can change

the frames name, its color, and make relative adjustments to its position and rotation with the Transform button.

The 3-Point frame construction methods in **SpatialAnalyzer** help you piece together frames in an intuitive graphical environment. The procedure is asks you to select an origin point, a point along the X, Y, or Z-axis, and a point on the corresponding plane. From this information, **SpatialAnalyzer** determines the coordinate transformation necessary to construct a frame consistent with your requirements.

Three Known Points

With this frame assignment method we assume that you know the coordinates (or partial coordinates) of three points in the workspace. This is often the case when you are building a fixture to a set of tightly held design values. Another way of expressing the above task is to say that you wish to find the “part coordinate frame” given that you know the design coordinates of three of your measured coordinates.

To begin this method, select the menu option. You will be prompted for each point in turn. After selecting the three points, the three known points dialog.

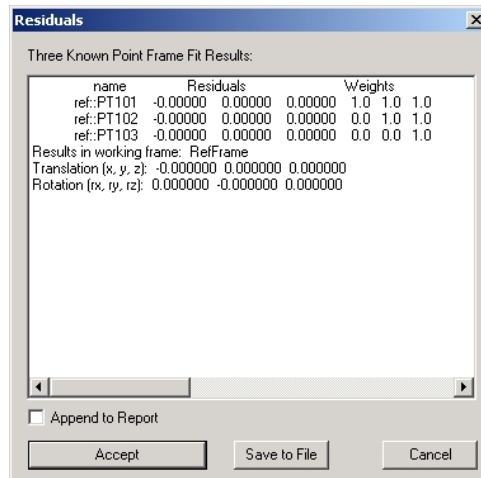
The 'Known Values' dialog box is used to define a new frame by specifying the coordinates of three points. It contains a table with columns for point references and their X, Y, and Z values, each with a checkbox to indicate if the value is a constraint.

	X Value	Y Value	Z Value
ref.:PT101	0.0 <input checked="" type="checkbox"/>	0.0 <input checked="" type="checkbox"/>	0.0 <input checked="" type="checkbox"/>
ref.:PT102	88.09754723 <input type="checkbox"/>	0.0 <input checked="" type="checkbox"/>	0.0 <input checked="" type="checkbox"/>
ref.:PT103	0.01768625 <input type="checkbox"/>	14.08768579 <input type="checkbox"/>	0.0 <input checked="" type="checkbox"/>

Buttons: OK, Cancel

Three Known Points frame construction dialog.

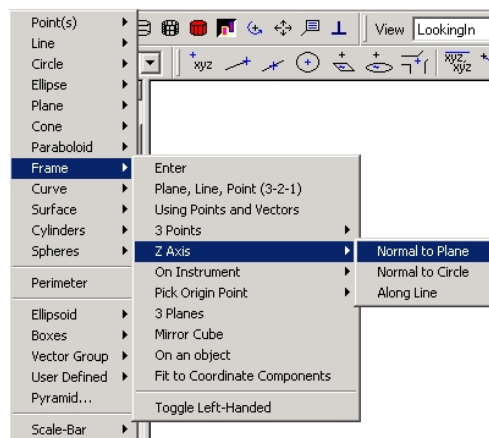
In the Known Values dialog the names of the selected points are on the left, followed by edit fields for the X, Y, and Z values of each point. Initially these values are populated with the current coordinates of the points in the working coordinate frame. Change the values to their design values. It is important to enter relatively good nominal values for each point. Failing to do so could, in some cases, result in the frame being improperly located. For each entered value that you wish to hold as nearly as possible, place a check in the corresponding box adjacent to each edit field. When you hit the OK button, the frame is computed and a report showing the transformation residual is presented. This data may be added to the Fit/Queries Report page if desired. You can accept or cancel the operation after inspecting the results of the fit.



Three Known Points Residuals Report Dialog

Z Axis Normal to Plane, Normal to Circle, or Along a Line

The Z-axis frame construction menu allows you to create a coordinate frame with the z-axis along one of several entities. These are the normal vector from a circle, the normal vector from a plane, or along a line.



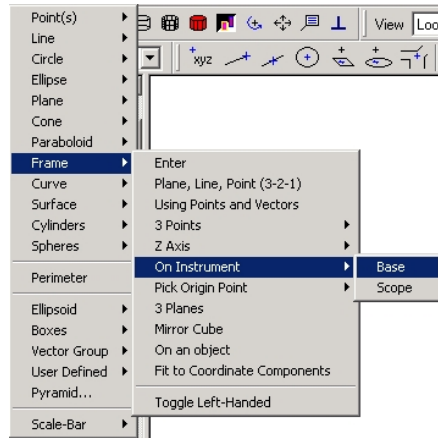
Z Axis Normal to Plane, Normal to Circle, or Along a Line menu

After selecting the entity for the Z direction, you will be asked to specify the origin point for the coordinate system. This point does not have to be along the Z-axis. The frame merely gets its orientation from the selection of the first object.

Next, select a point to determine the X-axis. This is referred to as a “clocking” point since it determines the rotation of the frame about the Z-axis.

On Instrument

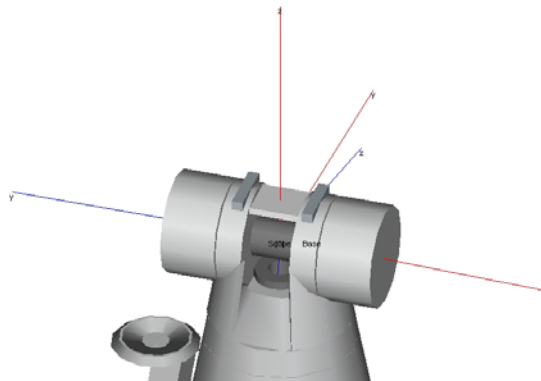
You may also place a coordinate frame on an instrument. This method is helpful if you need to determine the coordinates of a set of points relative to the instrument reference system.



On Instrument Frame construction menu options

There are two menu options when constructing a frame to align to an instrument's reference system. You may either place the frame on the instrument base, or on the instrument measurement center (e.g., scope, primary mirror, probe tip, center of rotation, etc). The frame on the instrument base is assigned so that it is “fixed” with respect to the instrument. The red frame in the figure was constructed with the Base option. This can be thought of as being attached to the instrument stand. That is to say, assigning a frame at the base of an instrument always results in the same frame regardless of the values of the instrument axes.

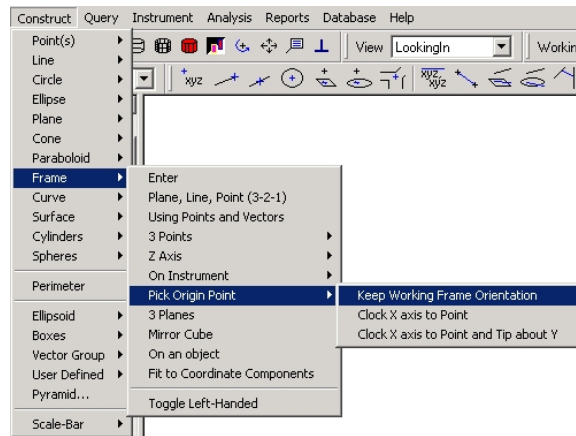
Conversely, a coordinate frame assigned with the scope method will place a frame so that the Z-axis of the frame points out along the scope of the instrument. The green frame in the figure shows the scope frame construction method. If you move the instrument, and do another scope frame assignment, it will be in a different location.



On Instrument Base frame (red) and Scope frame (green)

Pick Origin Point

These are three common frame assignments that utilize some information from new coordinates and some information from the existing working coordinate frame.



Pick Origin Point frame construction menu options

Keep Working Frame Orientation

This method essentially makes copies of the current working coordinate frame at specified locations. The operator is prompted to select an origin point and the frame assignment is accomplished such that the newly created frame has the same orientation as the working coordinate frame.

Clock X-axis to Point

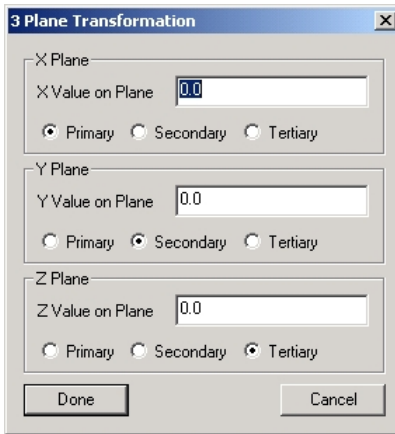
This method essentially performs the above frame assignment function and adds one more step. The final step in the process is to rotate the new frame about its own Z-axis until its X-axis points at the selected clocking point.

Clock X-axis to Point and Tip about Y

This method starts with the current working frame and asks you to select a frame origin, then a second point on which you want the X-axis clocked too. It then adds one final step. The final step in the process is to tip the frame about the Y-axis until the X-axis passes through the second point.

3-Plane Frame Method

A frame can be constructed from 3-Planes using the 3-Plane frame construction method in **SpatialAnalyzer**. The procedure asks you to select the X-plane, then the Y-plane, and finally the Z-plane. The 3 Plane Transformation dialog allows you to establish the X, Y, and Z component values for each of the three planes. Windows radio buttons allow you to define the precedence order (i.e., primary, secondary and tertiary) for the three planes.



3-Plane Transformation Frame construction dialog

After setting the component values and the relative order between the planes, hit the done button to create the frame. The frame's origin is set at the intersection of the three planes; the primary plane is used to lock the direction of its axis, while the secondary plane is used to clock its axial direction. The tertiary plane is used to orient the positive and negative directions for the last axis. The properties page for the new frame is presented to allow the frame's attributes to be customized for its specific use.

Mirror Cube Frame Assignment

Often, users wish to construct a coordinate frame based on the faces of a mirror cube. The Mirror Cube frame assignment method provides a convenient method for repeatedly, and accurately locating a coordinate frame with two instruments (typically theodolites). **SpatialAnalyzer** contains a powerful mirror cube frame assignment tool.

To accomplish a mirror cube frame assignment, point two theodolites at the mirror cube and auto collimate the instruments with the mirror faces. Record a shot from each theodolite at the cube using the same target name, just as if you were shooting the mirror as a target. If you desire reversed shots, reverse the scope and take another shot, or take multiple shots in each face. In **SpatialAnalyzer**, select Construct, Frame, and Mirror Cube. You will be prompted to select the target for the mirror. Pick the common target name you used for the measurements. The mirror cube frame assignment dialog will appear. At this point, you may press the Create Frame button to make the coordinate frame, or you may change the options.

Mirror Cube Coordinate Frame

Select the measurements to use in calculation

☒ X Axis ☐ X Axis

1 - Zeiss ETH 2 0 - Zeiss ETH 2

Theta=89.942721 Phi=89.942685 08/18/98 01 Theta=180.114797 Phi=89.885104 08/18/98 0

X 0.001000 Y -0.999999 Z 0.001000 X -0.999996 Y 0.002004 Z 0.002005

☒ Coordinate axes point outward (along theodolite axis)

Axis Error Weighting

Cube face angle 90.0 - Measured angle 89.828039 = Total angular error 0.171961

X Error Y Error

0.085980 deg. 0.085980 deg.

50% 50%

Compute Mirror Cube Frame

Frame Axes

X	-0.000501	-0.999999	0.001003
Y	-0.999998	0.000503	0.002007
Z	-0.002007	-0.001002	-0.999997

Create Frame! Cancel

Mirror Cube Frame Assignment Dialog

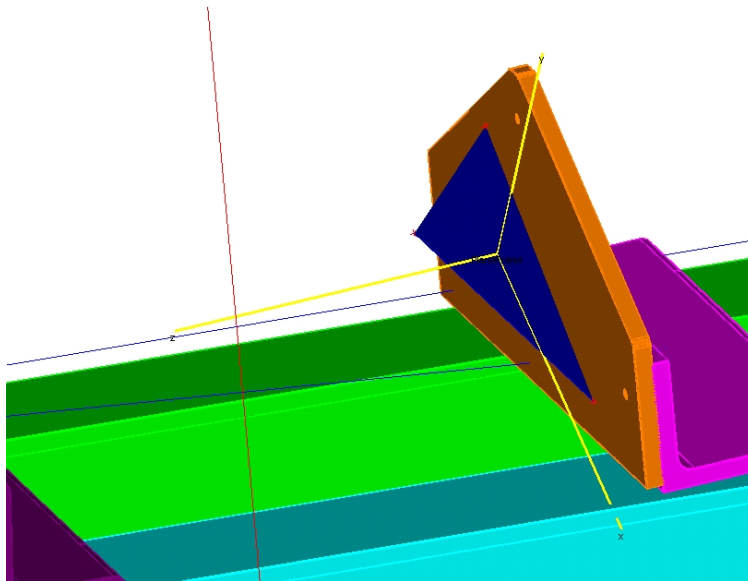
Possible Options:

- You may select which instrument represents the X-axis of the coordinate frame.
- You may select or deselect specific shots from each instrument. When there are multiple shots, their vector directions are averaged to calculate the vector displayed below the measurement box.
- The check box for coordinate axes determines whether the axes of the frame are outward along the theodolite vectors or inward toward the theodolites.
- The Axis weighting box allows you to control the distribution of the errors between the two axes.
 - Set the Cube face angle (normally 90 degrees). **Press Compute after doing this to update!**
 - The difference between the cube "ideal" angle and the theodolite vector angle will be displayed.
 - Use the slider bar to control how far the frame axes are away from the theodolite axes. Sliding the bar to the left places more of the error on the "right-side" instrument.
 - Choose carefully how to distribute the error. In some cases, it may be better to have no error on one axis since it is difficult to tell if the "error" is coming from the theodolite measurements or the mirror cube manufacturer.

On an Object Method

This method creates a frame on any of the objects in the **SpatialAnalyzer** database. The figure below shows a (yellow) frame created on a plane. The X and Y-axes are on the plane, and Z-axis is normal to the plane. This frame is then the starting point for a subsequent frame construction to clock the X and Y axes to be consistent with the components frame.

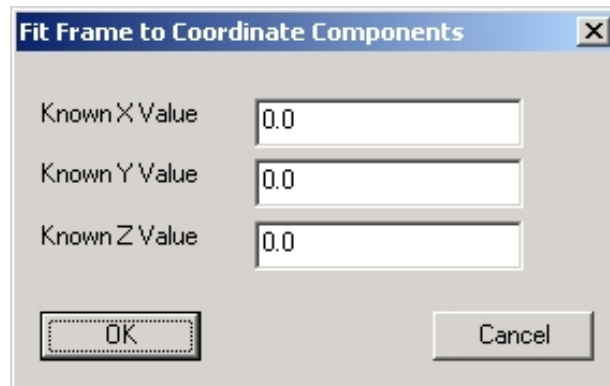
As a matter of convenience, all objects within **SpatialAnalyzer** have their own intrinsic coordinate frame. This frame is useful, and somewhat obvious for some objects. A paraboloid for example, has its frame located at the focus with the Z-axis pointing along the paraboloid axis. A Cylinder has its coordinate frame located at one end with the Z-axis along the cylinder.



Frame (yellow) on a plane

Fit to Coordinate Components

In this method of frame assignment, we assume that we have several points which share a known X coordinate, several points which share a Y coordinate, and several points which share a known Z coordinate. This is often used in a factory setting where a designer has established a reference block with one face representing a certain XY, one plane representing a certain XZ plane, and one face representing a certain YZ plane. Initially the user will be prompted for the points of known X value, then Y and Z in turn. After selection of the points is completed, the user will see the dialog shown below. Enter the known values and hit OK. The frame assignment will be made.

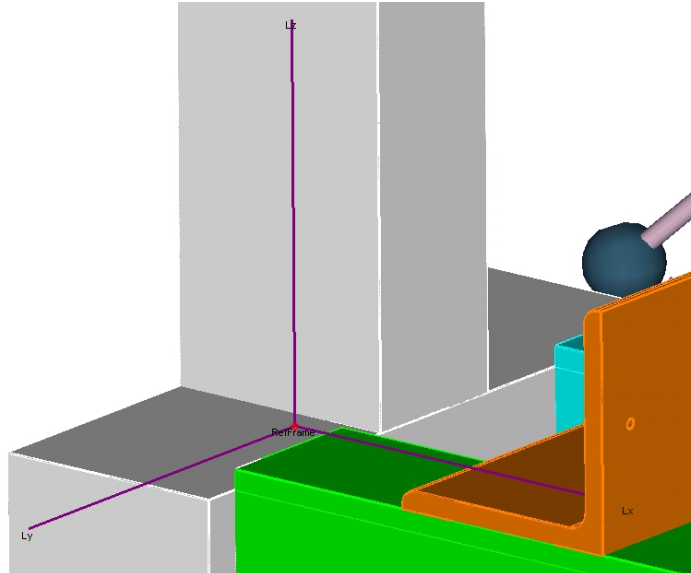


Coordinate Component Frame Fit

Toggle Left-Handed Frame Construction Method

All coordinate frames created using **SpatialAnalyzer** default to a right-handed convention. That is to say, if we take the vector cross product of the X-axis by the Y-axis, the Z-axis results. If, however, your application requires a left-handed system, you may change an existing right-handed system to a left-handed system.

This change will effectively reverse the Z coordinate axis and labels the frame axes Lx, Ly, Lz. To do this, select Toggle Left-handed from the Construct Frame menu. You will be prompted to select the coordinate frame you wish to change. After an “are-you-sure” message, the coordinate frame will be converted.



Left-Handed Coordinate Frame

Since the presence of left-handed coordinate frames in a model increases the risk of confusion, all reports will contain a special message attached to all left-handed coordinate frames. In addition, if the working coordinate frame is left-handed, a notice will be displayed on the header of report pages.

Changing the Working Coordinate Frame

The Working Frame toolbar is shown in the default **SpatialAnalyzer** environment. This toolbar contains a pull-down box for selecting coordinate frames. Any time a new frame is created, solved for, or imported from a CAD model – it appears on the Working Frame toolbar drop-down list. To change the active working frame, hit the Working Frame pull-down menu control and select the desired coordinate frame by name. The active working frame appears bold in the graphics view. If you wish to ‘click select’ a different working frame, instead of selecting it by name, select the ‘Pick from Graphics’ option from the Working Frame toolbar drop-down menu. Select the new working coordinate frame from the graphics view. The active Working Frame is the one you selected. Output from **SpatialAnalyzer** is now relative to this frame.

Basics of Spatial Transformations

Homogeneous transformations, transforms for short, can be used to represent the position and orientation of one coordinate frame (system) with respect to another coordinate frame. A general transformation matrix can be conveniently represented as a four by four matrix; only twelve of the sixteen numbers are typically used in this application³. This representation is unique, i.e., there is only one transformation matrix that describes a particular relative position and orientation. The problem with representing all transforms, as a matrix is that this representation is over-constrained. We know for example that a rigid object in space has only six degrees-of-freedom, three positional and three rotational. So, you simply don’t need all twelve numbers in the four by four matrix to represent the relative position and orientation. Let’s look at other more compact representations.

First of all, we need to define which frame is described relative to the other. Let’s assume we have frames **A** and **B**, and that we wish to describe frame **B** relative to **A**.

Position is easy. We simply describe the x , y , and z components of the origin of **B**. Note these terms are the distances along the x and y -axes of **A**. Also note the description of **A** relative to **B** would not, in general, be the negative of these previous values.

Rotation is not as easy as position. To describe rotations certain standard conventions have been adopted. These are by no means the only possible methods. The following sections will discuss different viewpoints that are commonly used to discuss transforms, then we will discuss simple unit rotations, and go on to develop compound rotation methods.

³ The four-by-four matrix is typically used because it is symmetric. Numerical routines can generally be made more efficient when the numerical pieces are symmetric.

The old viewpoint debate: Are we describing or transforming?

This section only becomes necessary because of the language that individuals use to discuss transforms. This is one of the most common sources of errors in dealing with spatial transformations. Transforms can be used in two different ways.

1) As a descriptor:

This is the way we will always talk about transforms in **SpatialAnalyzer**. This is the description of one frame relative to another. For example, frame **B** relative to frame **A**.

2) As a transform operator:

In this case we ask the question what operations we do to frame **A** to rotate and translate it to the position and orientation of **B**. In some cases, it's natural to think about transforms in this way. For example, if you want to describe the position of a box as it moves, it's natural to think about the rotations and translations that it is subjected to over time.

The final result is a little counter intuitive, but essentially, it doesn't matter which view point you use, they produce the exact same results. You just need to be aware of the fact that despite the language used the math is the same. Many people incorrectly interpret one as the inverse of the other.

Unit Rotation Matrices

Before we can discuss compound rotations we need to develop the building blocks of unit rotations. These are simply rotations that occur about any given principal coordinate axis (X, Y, or Z). They are as follows:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Or, for the particular example of an X rotation of 30 degrees, a Y rotation of 60 degrees, and a Z rotation of 90 degrees:

$$R_x(30) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & -0.500 \\ 0 & 0.500 & 0.866 \end{bmatrix} \quad R_y(60) = \begin{bmatrix} 0.500 & 0 & 0.866 \\ 0 & 1 & 0 \\ -0.866 & 0 & 0.500 \end{bmatrix} \quad R_z(90) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Compound Rotations

Using the building blocks of the above unit rotations, we can now proceed to develop more complex compound rotations. As we stated earlier, a rotation matrix is unique. However, the descriptive methods we often use to describe a rotation are not. The same physical orientation can be described in an infinite number of ways. We'll explore the most common below.

Fixed Angle XYZ

In this case, we will start with frame A and B coincident. We will then rotate frame B about the X-axis of A (X_A), then about the Y-axis of A (Y_A), then about the Z-axis of A (Z_A). We can express this as

$$R_{XYZ}(30,60,90) = R_z(90)R_y(60)R_x(30) = \begin{bmatrix} 0.000 & -0.866 & 0.500 \\ 0.500 & 0.433 & 0.750 \\ -0.866 & 0.250 & 0.433 \end{bmatrix}$$

Fixed Angle ZYX

In this case, we will start with frame A and B coincident. We will then rotate frame B about the Z-axis of A (Z_A), then about the Y-axis of A (Y_A), then about the X-axis of A (X_A). We can express this as

$$R_{ZYX}(90,60,30) = R_x(30)R_y(60)R_z(90) = \begin{bmatrix} 0.000 & -0.500 & 0.866 \\ 0.866 & -0.433 & -0.250 \\ 0.500 & 0.750 & 0.433 \end{bmatrix}$$

Euler Angle XYZ

In this case, we will start with frame A and B coincident. We will then rotate frame B about the X-axis of B (X_B), then about the Y-axis of B (Y_B), then about the Z-axis of B (Z_B). Note that the axes of frame B do rotate during the operations. We can express this as

$$R_{XTZ}(30,60,90) = R_x(30)R_y(60)R_z(90) = \begin{bmatrix} 0.000 & -0.500 & 0.866 \\ 0.866 & -0.433 & -0.250 \\ 0.500 & 0.750 & 0.433 \end{bmatrix}$$

Euler Angle ZYX

In this case, we will start with frame A and B coincident. We will then rotate frame B about the Z-axis of B (Z_B), then about the Y-axis of B (Y_B), then about the X-axis of B (X_B). Note that the axes of frame B do rotate during the operations. We can express this as

$$R_{ZTX}(90,60,30) = R_z(90)R_y(60)R_x(30) = \begin{bmatrix} 0.000 & -0.866 & 0.500 \\ 0.500 & 0.433 & 0.750 \\ -0.866 & 0.250 & 0.433 \end{bmatrix}$$

Euler Angle ZYZ

In this case, we will start with frame A and B coincident. We will then rotate frame B about the Z-axis of B (Z_A), then about the Y-axis of B (Y_B), then about the Z-axis of B (Z_B). Note that the axes of frame B do rotate during the operations. We can express this as

$$R_{ZYX'}(90,60,90) = R_z(90)R_y(60)R_z(90) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -0.500 & 0.866 \\ 0 & 0.866 & 0.500 \end{bmatrix}$$

Equivalent Angle Axis

This representation is a little different from the others given in that it does not result from the combination of unit rotations. This convention stems from the fact that for any given rotation there exists a unique axis of rotation and magnitude of rotation. If we rotate about a unit vector K and amount θ , we find that the following matrix results:

$$R(K, \theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & x k k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ x k k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}$$

where $v\theta = (1 - c\theta)$, and k_x, k_y, k_z are the components of unit vector K . Through an inverse analysis we can extract the (K, θ) for any rotation matrix.

Quaternion (also known as Euler parameters)

Quaternion is simply another way of expressing the information that is inherent in the equivalent angle-axis representation. Namely,

$$\begin{aligned} \varepsilon_1 &= k_x \sin\left(\frac{\theta}{2}\right) & \varepsilon_3 &= k_z \sin\left(\frac{\theta}{2}\right) \\ \varepsilon_2 &= k_y \sin\left(\frac{\theta}{2}\right) & \varepsilon_4 &= \cos\left(\frac{\theta}{2}\right) \end{aligned}$$

MEASUREMENT ERROR AND UNCERTAINTY

The world is full of uncertainty, much of which has a direct effect on the kinds of predictions we want to make in 3D Metrology. No serious metrology work should be completed without estimating the uncertainty in its results. **SpatialAnalyzer** is able to objectively estimate the uncertainty of your measurements and present them graphically and on reports. In this chapter, we will discuss various aspects of uncertainty and measurement error, including the error types, instrument and operator error, how to reduce measurement error and thereby reduce measurement uncertainty.

Understanding where some of the errors in measurements come from, and why we can never eliminate them completely is an important part of good metrology practice. The differences between your measured values and the true values are the result of an expected combination of real-world imperfections in the measurement process. There are many contributing factors to measurement imperfection; a few of them are vibration, instrument resolution and instability, human measurements, air currents, thermal fluctuations etc. Due to these and other factors, no matter how carefully a measurement is performed, it will always contain some error.

Measurement Error

There are three basic types of error, Systematic, Random and a Blunder. The first two are measurement errors and are described below. The third, a Blunder, is not really a measurement error. It is separated because; this type of problem can be minimized or even eliminated with proper procedure that is rigorously adhered too. Blunders are basically mistakes, and are not classified as measurement errors. An example is when an operator mislabels a point. This is an error, but it is not a measurement error.

What is Systematic Error?

Systematic error is an error in a measurement, which follows some mathematical or physical law. They are caused by any factors that systematically affect the measurement of the variable across the sample. If the conditions causing the error are measured, a correction can be calculated and then applied to eliminate the systematic error.

Systematic errors will generally remain constant in magnitude and algebraic sign if the conditions causing them are not addressed. Because they tend to be either positive or negative, systematic errors accumulate. Since they tend to accumulate, systematic errors are sometimes called cumulative errors or more-technically, **bias**.

What is Random Error?

After the systematic errors have been eliminated, the errors that remain are called random error. Random errors are generally small, however they can never be completely eliminated. They do not follow physical laws like systematic errors. They are just as likely to be positive, as negative, and therefore they do not affect the average if enough data is collected. The important property of random error is that it adds variability to the data but does not affect average performance for the group. Because of this, random error is sometimes considered **noise**.

Reducing Measurement Error

So, how do you reduce measurement errors, random or systematic? The first thing to consider is to make sure your instruments are properly maintained and compensated. To ensure they are operating properly, always perform field checks before and after the measurement job. Every instrument interface in **SpatialAnalyzer** contains a number of field check procedures.

Compensate the instrument using the instrument manufacturer's procedure. If the field check measurements do not fall within the specification defined by the instrument manufacturer; compensate it then check it again. If this process fails the specification check a third time; contact the instrument's recommended service center to inquire about how the instrument should be repaired.

SpatialAnalyzer will provide you with feedback concerning the consistency and degree of noise on your measurements in the instrument interface. For example the standard deviation for a sample of points used to compute a shot or observation. By monitoring this information you can typically ascertain the degree of random error that the instrument, operator, and environment are imposing on your measurements.

Training the operators that are using the equipment is possibly the most important way that you can reduce measurement error. The data collect by untrained operators can be less reliable because they are more likely to inadvertently introducing error.

When you collect the data, you should double-check the data thoroughly. The graphical environment of **SpatialAnalyzer** allows you to do this visually by rotating, panning, and zooming the measurements against the nominal CAD design. Statistical reports on fits can be saved and printed from the Reports Fits/Query menu option. These should be checked to verify that the results are within the expected tolerance range.

By collecting and using more than the minimum amount of data, you can use the extra information to confirm your results. **SpatialAnalyzer** is adept at dealing with redundant data in each of the fitting routines. Each routine then produces a report that allows you to evaluate the errors in the result. A suite of queries for comparing objects (measured to reference) can help illustrate the systematic and random components of error in the measurements.

Finally, one of the best things you can do to deal with measurement errors, especially systematic errors, is to use multiple measures of the same object. That includes points, planes, surfaces, spheres etc. To take it one step further, this is a particularly useful technique if you use different measurement technologies that don't share the same systematic errors. **SpatialAnalyzer** is able to interface to a wide array of measurement systems, using one to check another is a robust systematic error-trapping scheme. You are able to collaborate with multiple measurement systems to collect more accurate and reliable measurements.

Sources of Measurement Error

There are many possible sources for error in a measurement system. These include operator error, instrument error, and setup error. The end result is a certain degree of

coordinate uncertainty. The purpose of this chapter is to explain these errors and the way in which **SpatialAnalyzer** can objectively assist you in knowing how big they are, where they are coming from, and how changes in the measurement process affect the final measurement uncertainty. With objective measurement uncertainty information that is easy to access and understand, you can effectively manage the measurement process.

Generally there are three sources of measurement uncertainty that should be quantified for measurement uncertainty; they are Operator, Instrument, and Setup. Each contributes uncertainty in almost every measurement job. Their contributions can vary significantly depending on the application. **SpatialAnalyzer** is able to characterize and combine each these components in a process that meets the International Organization for Standardization (ISO) Standards for expressing measurement uncertainty⁴. Each of the three contributing components is discussed below.

Operator Error

To a great extent the nature of operator error is dependant on the instrument system being used. For a theodolite, operator error can result from improper target sighting or focusing. For a tracker, an operator could accidentally lift the target (SMR) off of a surface while scanning data. Both of these examples could be small in magnitude and hard to detect simply by looking at the data. Operator error cannot always be distinguished from other errors. For this reason, it is a difficult error component to quantify. Depending on the manner in which the errors occur they could be systematic or random. It is important, however, to understand that this error does exist and that good measurement practices can seek to minimize or eliminate these errors.

For a theodolite network, sighting errors will contribute to the shot quality value for each target. It is important to note, however, that the shot quality may also include other errors. If, for example, the theodolite network was not properly setup, shot quality errors could result even if the operators were able to perfectly center the crosshairs on the target. The operator could, for example, be perfectly centered on a target with a perfect setup, and still have a shot quality error due to the setup and its dependency between instruments, operators, and the relative geometry between the instruments and the target.

Perfect shot quality does not necessarily indicate a good measurement. For this reason, it is necessary to determine the sensitivity of the target location relative to the instrument errors. An example would be a two-theodolite network where the instruments are close together. It is possible to get a good shot quality value even though the setup is highly sensitive to instrument precision errors. But, that does not mean the points coordinates have small uncertainties.

Differences in how an operator uses an instrument and the degree of variability that an operator has between the first and last shot of the day contribute to measurement uncertainty. Quantifying this contribution and its effect is done based on the operator's

⁴ The International Organization for Standardization (ISO) standard for expressing measurement uncertainty is called the "*Guide to the Expression of Uncertainty in Measurement*", or GUM, 1995

experience and the degree that the operator can affect a particular instrument's measurements. To conclude, operator error cannot always be distinguished from other errors. For this reason, it is a difficult error component to quantify. It is important, however, to understand that this error does exist.

Instrument Error

There are two potential components that contribute to instrument error. Instrument Resolution is concerned with the physical limitations of sensors used to detect angular or linear relationships. Instrument Calibration is concerned with how well adjusted the instrument is (either physically or through a compensation algorithm) to the nominal instrument design.

Instrument Resolution

Fundamentally, most metrology instrumentation either measures angles or distances. Both angles and distances occur in a continuum. There are an infinite number of distinct angles between 0 and 1 degree. We could, for example, count 0, 0.1, 0.2..., or we could count 0, 0.01, 0.02,....., or continue with finer and finer increments. Although there are an infinite number of angles between 0 and 1 in the real world, there are a finite (or fixed) number of angles as interpreted by a measurement sensor. As a crude example, there might be only 16 angles between 0 and 1. This would mean that our sensor counts with the following sequence 0/16, 1/16, 2/16,....(or in decimal form, 0, 0.0625, 0.1250,). If you were measuring a 0.1000 degrees with this sensor, you could never get closer than 0.1250 degrees and you would have a resulting error of 0.0250 degrees. In practice, we never really know what the is (why measure at all if we did?). So the only way we can address the error is to talk about the probable error based on the instrument resolution. Instruments with high resolution have low probable errors. Conversely, instruments with low resolution have high probable errors. We should also note that the concept of "high" or "low" resolution is a relative measure. Instruments are not easily classified as good or bad, superior or inferior. The goal is to ensure that a particular choice of instrumentation is appropriate to the task at hand.

By their nature, instrument resolution errors are random but bounded. Sometimes the sensor will be closer to the next higher value and sometimes it will be closer to the next lower value, but it is not possible that the resulting error could be greater than one half of the finest increment that can be measured.

Instrument Calibration

Most modern measurement instruments have internal compensation factors. These factors are used to represent discrepancies from the nominal design of the instrument and the actual measurement device. These factors are usually determined by following a manufacturer's compensation, (or as it is frequently called), calibration procedure. Once the procedure has been executed, the calibration factors in the instrument are updated, and applied to all subsequent measurements.

For the case of theodolites, great care is taken during manufacturing to insure that the horizontal and vertical axes of the device intersect at a point. In addition, these axes should be at 90 degrees to each other. Despite the careful manufacture of these devices, misalignment and offsets do creep into the final product. By sighting points from both faces of the instrument, it is possible to compute the misalignment. Once the

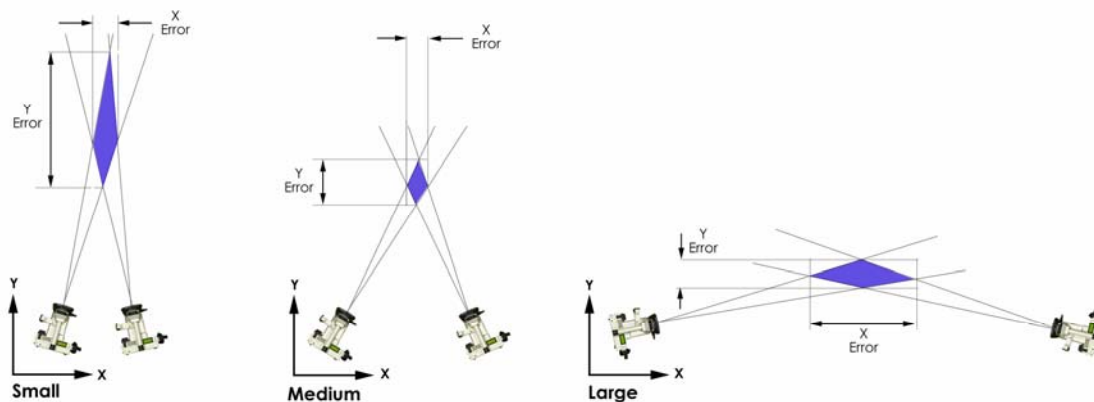
misalignment is known, it is (typically) used stored in the instrument. Future measurements are then compensated with the by the instrument to adjust the horizontal and vertical angles from the sensors to represent the instrument as if it were perfectly manufactured.

Of course, errors may occur during this calibration process. These errors will affect all subsequent setups and measurements.

Uncertainty contribution from the Survey or Job Setup

The way in which the measurement instruments are positioned relative to the object being measured can affect the measurement uncertainty. Let's use a photogrammetry analogy to explain the point.

The error envelopes in the figure below show small, medium and large apex angles. The lines emanating from the cameras represent the error bounds on the image measurement of the point. The intersection area of the two sets of lines represents the error bounds of the coordinates of the measured point. It is clear that at small apex angles ($<45^\circ$) the error is mostly in the direction perpendicular to the instrument baseline, at medium apex angles ($45-90^\circ$) the errors are about equal in all three directions, and at large apex angles ($>90^\circ$) the error is increasingly in the direction parallel to the instrument baseline.



Apex-angle Error Envelope Diagram

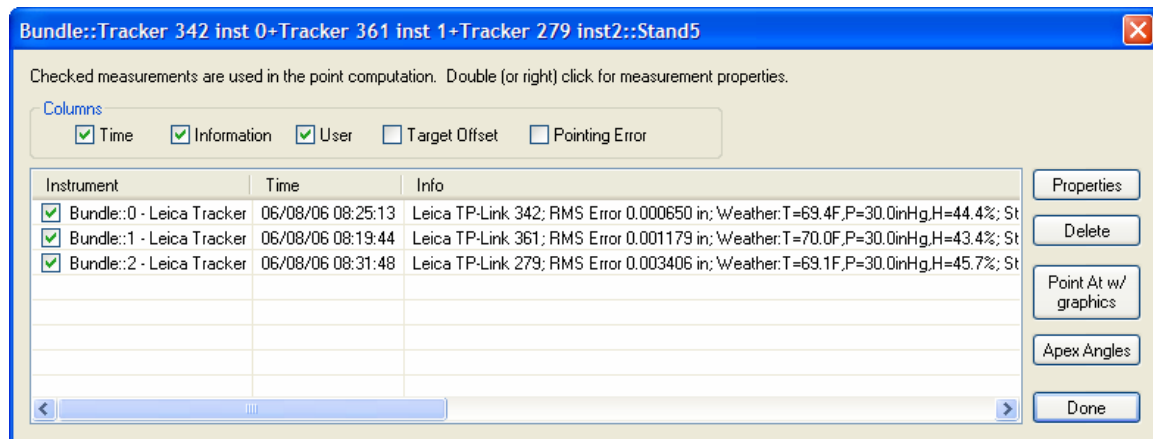
Coordinate Uncertainty

SpatialAnalyzer provides powerful coordinate sensitivity analysis functions. Essentially, uncertainty is determined by injecting normally distributed, random error into the measurement system and evaluating the effect of the error on the coordinate data. The amount and type of error that is inserted is based on the precision of the measuring device.

To illustrate the sensitivity analysis system, let's consider the case of a tracker network. Suppose a particular point was shot with 4 instruments. **SpatialAnalyzer** will display the X Y Z coordinate of the measured point and the Angular Pointing Error in the targets

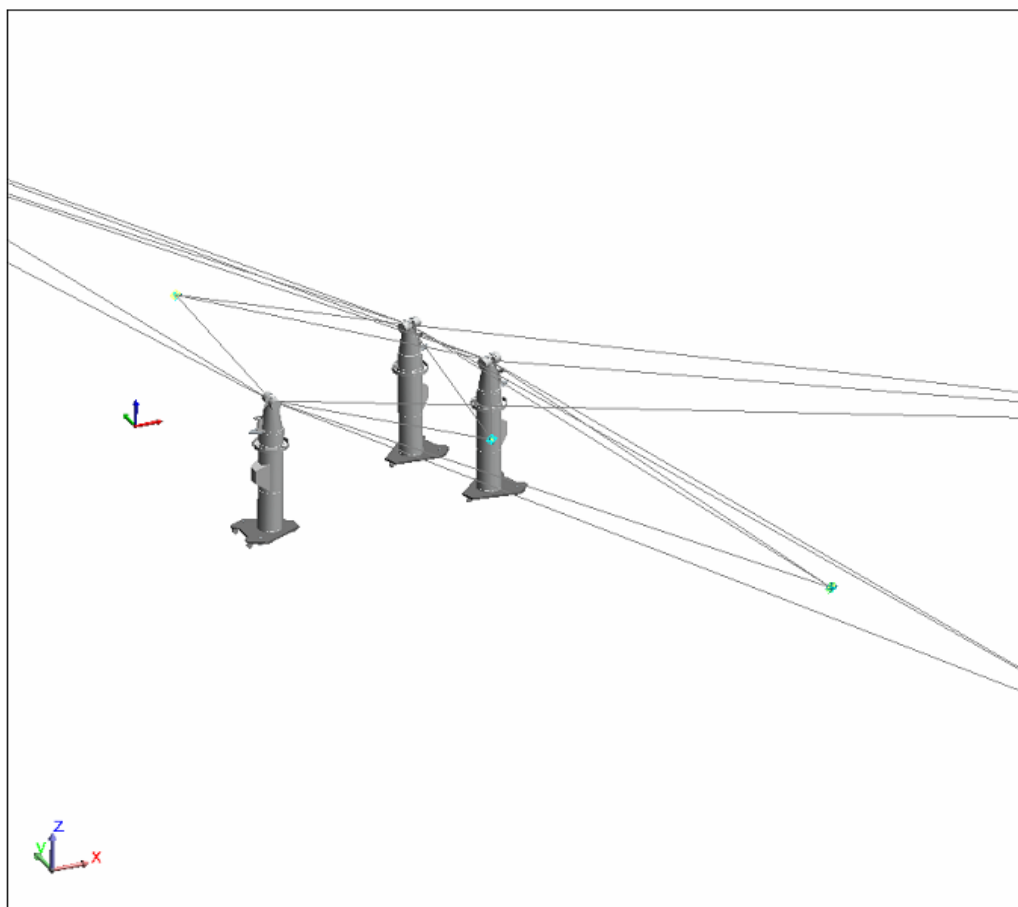
property window. The RMS pointing error is the angular distance between each of the tracker shot vectors and the computed point location.

If you double-click the coordinate to view more information, and press the Details button, the program will pause briefly and display a Details window. This window will contain the XYZ coordinate values as well as the coordinate uncertainty for X Y and Z. Also notice the confidence interval and sensitivity iterations values. The sensitivity iterations are the number of times random error was inserted and the coordinate was recomputed. This value may be changed in the User Option's dialog from the tab.



Target's Details Dialog

Once the random perturbations of the measurements have been performed and the results tabulated, **SpatialAnalyzer** samples the x y and z variation of the coordinate value in the working coordinate frame. These samples are subjected to statistical analysis to determine the standard deviation. Once the standard deviation is determined, it is scaled based on the confidence interval to yield the uncertainty. Values for each measured targets can be computed with the Analysis >> Coordinate Uncertainty >> Create Point Uncertainty Fields function. The figure below shows an example Quick Report with point uncertainties. These uncertainties were computed at the default 68% confidence interval.



Group - Bundle::Tracker 342 inst 0+Tracker 361 inst 1+Tracker 279 inst2							
Point Name	X	Y	Z	Ux	Uy	Uz	Mag
Stand0	357.5004	-163.6130	6.1531	0.0008	0.0010	0.0011	0.0017
Stand1	207.8194	-353.3290	2.4244	0.0045	0.0017	0.0047	0.0067
Stand2	163.3905	-202.7019	-0.6747	0.0015	0.0008	0.0017	0.0024
Stand3	107.3388	-51.7131	-2.1779	0.0005	0.0005	0.0005	0.0009
Stand4	57.8582	94.7064	-4.2084	0.0005	0.0005	0.0005	0.0008
Stand5	6.5245	249.6159	-6.0587	0.0019	0.0008	0.0019	0.0028
Stand6	-51.8881	403.8083	-9.0842	0.0041	0.0015	0.0045	0.0063

Quick Report of Point Group with Uncertainties

The standard deviation is defined as a one-sigma confidence value. This is based on several statistical assumptions. These assumption include the fact that the standard

deviation, “s”, is a valid estimator of sigma. If we took an infinite number of samples of the data, s would indeed accurately represent sigma. Since we are not patient enough to allow the computer to sample forever, we choose some number of samples that we deem to be adequate. Statisticians present several rules of thumb that may be used to choose an appropriate number of samples. 30 samples are often taken to be an adequate sample set, though the Analyzer defaults to 100 samples.

Once the specified number of samples has been used to determine the standard deviation, the deviation is scaled by the user-specified sigma value. If, for example, the user selects a confidence interval of 95%, the standard deviation will be multiplied by the sigma value that corresponds to the 95% confidence interval. In this case, 1.96 sigma will be used. The result of the multiplication will be the coordinate uncertainty value that contains 95% of the statistical population.

The mapping between the user’s specified confidence interval and the corresponding sigma value is based on the assumption that the data is normally distributed. Given that assumption, the mapping of sigma values is determined by computing the area under the normal distribution curve. The area is computed over the bounds that correspond to the desired confidence interval percentage.

Measurement Perturbations

The coordinate uncertainty analysis is based on the variation in the point caused by introducing error into the measurement values. The amount of error that is introduced depends on both the instrument precision and the pointing error for the point under consideration. The instrument error is the angular encoder resolution of the horizontal and vertical instrument axes.

The pointing error is the angle between the measured theodolite shot and a “perfect” shot from the instrument directly through the computed point. Since the computed point is the best-fit coordinate for all of the given measurements, the pointing error may be thought of as the amount a particular measurement differs from the other measurements of a particular coordinate.

Suppose a particular coordinate is computed based on 3 shots from a theodolite network. In order to determine the uncertainty for the point, the measurements must be perturbed enough times to yield a statistically valid sample set. For each perturbation sample, all of the measurements will be randomly perturbed. The encoder resolution of its axis plus the angular pointing error of the measurement perturb the first measurement’s horizontal angle. The same formula is used to perturb the vertical axis.

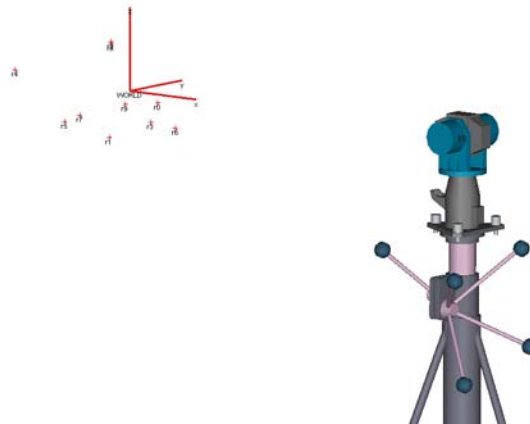
Once all of the instruments have been perturbed, the coordinate is re-computed using the perturbed measurements. This counts as one sample. The process is repeated until the required number of samples has been taken. Next, all of the samples are analyzed to determine their statistical parameters.

Uncertainty Analysis and Visualization

SpatialAnalyzer is able to compute and report measurement uncertainty estimates for each of the instruments it interfaces too. Looking through a long list of computed uncertainty estimates can sometimes yield little insight into the specific effects that it

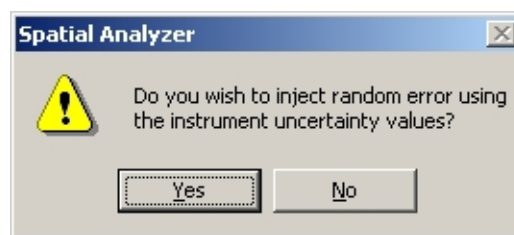
might be having on the measurement job. To enable users to make tangible use of the uncertainty estimates, **SpatialAnalyzer** can generate a cloud of points (around a target) that represents the target's region of uncertainty. The point cloud is called a sensitivity cloud.

Let's use a laser tracker as an example. The figure shows a laser tracker and a series of points to be measured.



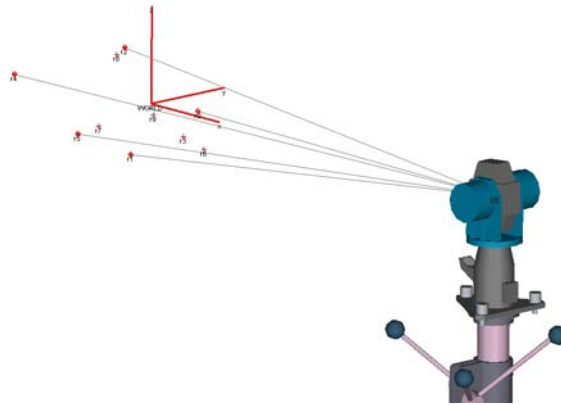
Laser Tracker and points to be measured

Using the **SpatialAnalyzer**'s simulation mode, we can fabricate measurements to a number of the points; in this case 5 points were measured. The simulation engine in **SpatialAnalyzer** allows you to inject random error into the fabricated measurements using the dialog. The amount of random error is based on the instruments uncertainty parameters from its properties dialog. For a laser tracker instrument, the horizontal and vertical angles are perturbed from perfect measurements along with the range component.



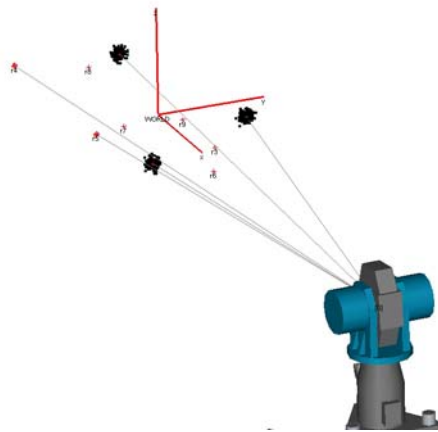
Random error injection request dialog for measurement simulation

The figure below shows the fabricated measurements at the points. The measured points are now considered targets in **SpatialAnalyzer**, since they now have observations.

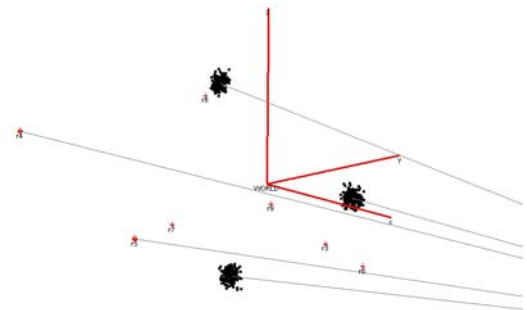


Instrument and measured targets

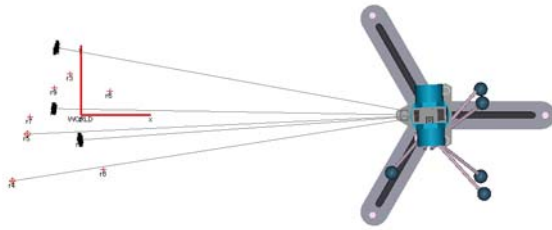
To help you visualize the uncertainty of any of the measurements in the job, **SpatialAnalyzer** creates point clouds around the targets. The cloud's volume models the geometric dimensions of the target's uncertainty and gives you an understanding of its orientation relative to the object being measured. It also allows you to evaluate the potential effects of the measurement's uncertainty. The figures below show different view perspectives of three of the targets and their sensitivity clouds.



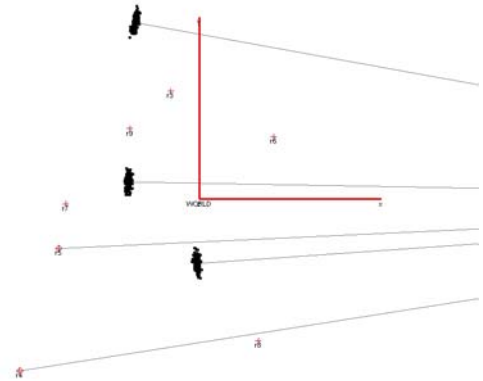
Instrument and targets with sensitivity clouds



Instrument and targets with sensitivity clouds, close-up



Instrument and targets with sensitivity clouds, top view



Instrument and targets with sensitivity clouds, top view, zoomed

Communication between the users of 3D metrology systems and their customers can be greatly improved with this uncertainty visualization capability. It provides an objective real-world perspective of the uncertainties in measurement in a fashion that enables people that are new to these concepts to understand them and make confident metrology decisions, which is one of the fundamental goals of **SpatialAnalyzer**.

BEST-FIT COORDINATE TRANSFORMATIONS (OR 7-PARAMETER TRANSFORMATIONS)

Best-fit transformations are widely used in coordinate metrology to consistently and confidently align measurements into a coordinate frame that is meaningful to a user. Measurements made with an instrument should be presented to users in the part coordinate frame. The best-fit transformation routine in **SpatialAnalyzer** is a good method to do that.

The best-fit transformation solves for the difference between an instrument's measurements of a group of points and the expected coordinates for those points. The difference is the "Transformation." After it is solved for with the best-fit transformation routine, **SpatialAnalyzer** shows you the difference and allows you to apply the transformation to your measurements of the points and to the other objects in the database. Once applied, **SpatialAnalyzer** is able to report those measured points and any new measurements in a part coordinate frame that is meaningful to you.

The Best-Fit Coordinate Transformation alignment technique has many applications; for example, to combine sets of coordinate data or locate an instrument in the part coordinate system using surface features imported from a CAD system. Simply stated, this alignment technique helps you determine the transformation that when applied to one set of coordinates causes the match-up between the two sets to be optimum or in other words, "Best Fit."

The term "Best Fit" is the common name given to a mathematical procedure known as a Least Squares Adjustment. The transformation solved for and then applied to measurements is determined using least-squares optimization methods. A least squares optimization seeks to minimize the square of the distance between each pair of points in the two coordinate sets. The following section on Least Squares Adjustment is intended to give you a brief introduction on how it works and why it is used in **SpatialAnalyzer**.

Least Squares Adjustment

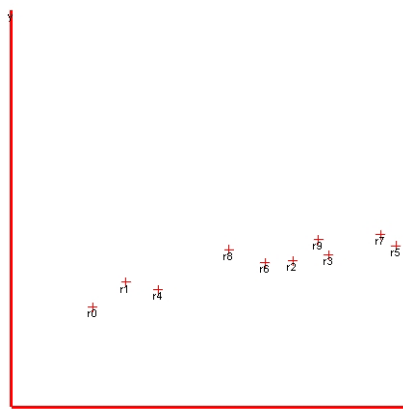
Least squares is a procedure for adjusting observations containing random errors. It is by no means a new method. Karl Gauss, a mathematician in the latter part of the eighteenth century, developed it. It was not used much until computers became commonplace because of the lengthy calculations. **SpatialAnalyzer** uses it in many routines (transformations, best-fit lines-planes-circles etc), to provide the best and most consistent answer for your fitting problems.

For a group of equally weighted observations (or measurements in our case), the basic condition that is enforced in least squares adjustments is that the sum of the squares of the residuals is minimized. Residual is another name for the difference between the measured value and its most probable value. So for 3D coordinate metrology, one example of a residual would be the difference between the true X-coordinate for a point and the actual X-coordinate measurement.

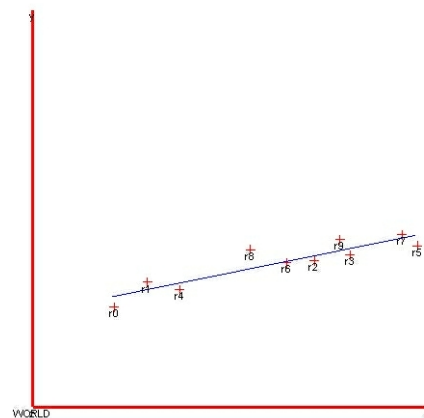
The method of least squares is a criterion for fitting a specified model to observed data. In this case, the specified model is the reference set of points (or expected values), and the observed data are typically your measurements of the reference set of points.

Let's use a common use of the least squares adjustment as an example. The least squares line routine is a commonly used method for computing the best-fit line from a series of measurements of a linear function. The distance your car travels when you're driving at a particular speed is an example of a linear function.

In this example, you are trying to drive your car at a constant speed and measure how far you have gone and the time it took you to get there. While trying to do all three things at the same, it is likely that a little error is going to creep into the process. That error is generally referred to as random error. Least squares adjustment techniques are really good at dealing with this type of error and are able to produce a reasonable and consistent answer in spite of it. An example is shown in the figures. The time is plotted on the X-axis and the distance on the Y-axis.



Time vs. Distance measurements



Least squares line fit to the measured points

The points are the paired measurements made of time verses the distance traveled, which is linear in this case because you were trying to maintain a constant speed.

To find what is considered the “best line” through the data the least squares line routine is used. In this case, the residual that gets minimized is the difference (squared) from a measured point to the line that best describes the distance you traveled verses the time it took. The least squares line routine will find a line, such that the sum of the distances (squared) from the data points to the line is the minimized. The figure shows a line that best fits those points; it was solved for with the best-fit line routine in **SpatialAnalyzer**. The slope of the best-fit line is actually the average speed that you traveled on your trip. This line is the most consistent way of determining where you probably were at any point in time during your trip.

The least square line adjustment was able to use this set of measurements even through each point had some error, and it produced a high-quality and reliable answer. This ability of dealing with measurements that have at least some component of random error and still being able to produce consistent and representative answers is why the technique is used in **SpatialAnalyzer**.

Best Fit Spatial Transformations

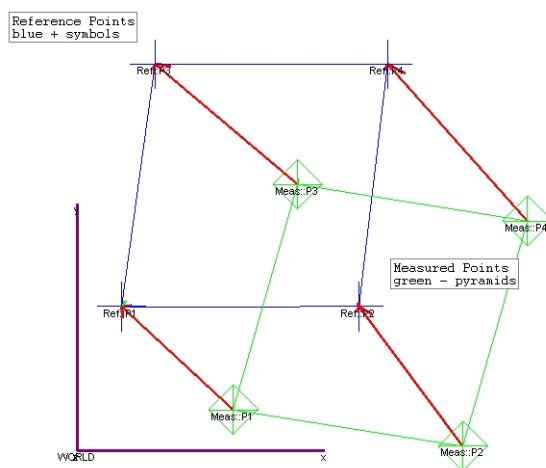
Least squares spatial transformations are the primary purpose for the Best-Fit Coordinate Transformations routine in **SpatialAnalyzer**. The primary difference between the best-fit line and the best-fit spatial transformations is that the residuals are now the distances between pairs of measured points and known references for those points.

As discussed previously, points in **SpatialAnalyzer** are organized into groups. This construct is particularly useful for the best-fit transformation option since the best-fit will attempt to best-fit common points using common point's names. This results in several requirements:

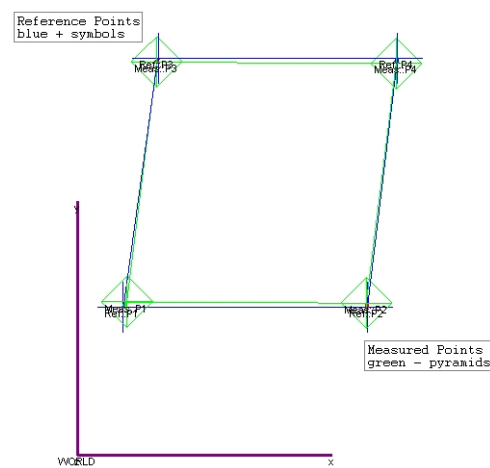
- The two sets of coordinates that will be best-fit must reside in separate groups. (for more information on groups, see the System Architecture chapter)
- There must be at least three points in each group with matching names.

If you have two groups of points that meet these requirements, you are ready to proceed.

Figure below shows the relationship between a set of measurements (green – pyramid symbol points) and their respective reference values (blue – plus symbol points). The red arrows from the measurements to the references indicate the difference between the two sets. Using the Best-Fit Coordinate Transformation routine in **SpatialAnalyzer** you can align the measurements to the reference points.



2D spatial difference between measurements and reference points



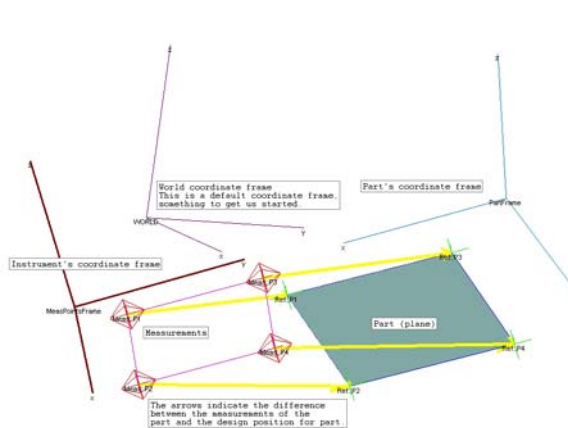
Applied least squares 2D transformation

The figure Applied least squares 2D transformation shows the location of the measurement set after the best fit is applied. Just as in the best-fit line example, the

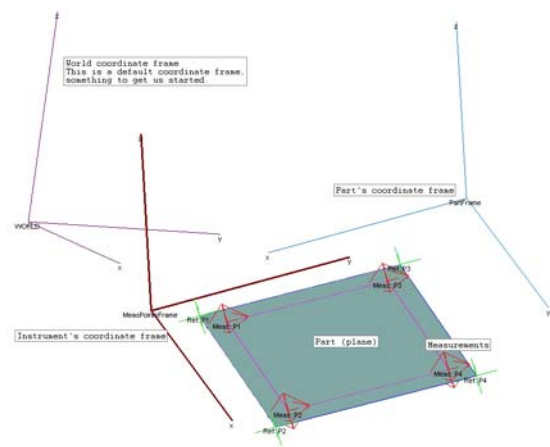
alignment in this case is optimized so that the sum of the residuals (squared) between the measurements and the known reference values are minimized.

The fit is a consistent and reliable alignment between the measurements and the known reference values. It accomplishes the reliability and consistency for a number of reasons. Specifically it was performed with four points. An approximate alignment is possible with just two points. By using four points to transform, the random error that exists in the measurements is effectively averaged out. Without the redundant information (i.e., extra 2 points), the process becomes an alignment, which can be distorted by the random error. The fit was also a strong one because the points were distributed around the boundaries of the part. This makes the fit geometrically stronger. If the points had all been in a line, the fit would have been strong along the line, but weak in areas away from the line.

The process and consequences are the same when best fitting points in 3D. The next example is illustrated in the figures below. Targets (shown as red pyramids) are transformed on to the known reference points (shown as green plus symbols.) Yellow arrows indicate in the initial difference between the targets and the known reference points.



3D spatial difference between measurements and reference points

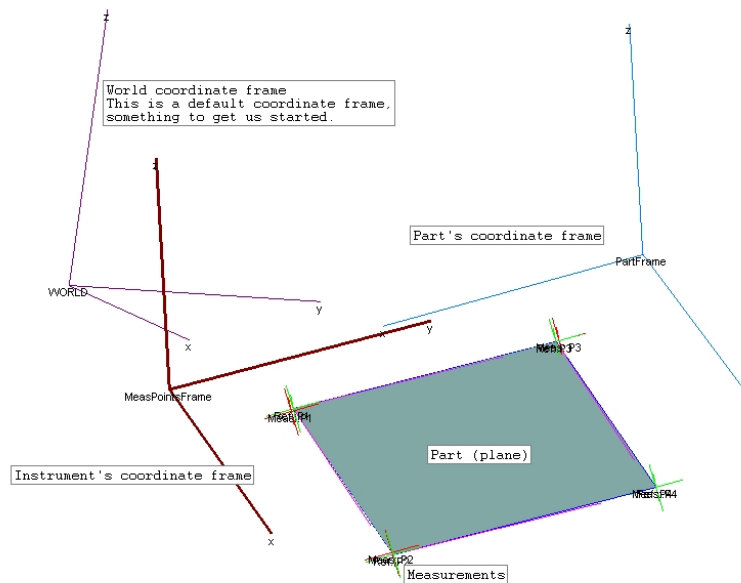


Applied 6 parameter least squares transformation

An instrument coordinate frame (brown) is shown moving with the targets into the parts coordinate frame (blue). By applying the transformation to the instrument, future measurements can be reported in the part's frame.

The term 3D is a bit misleading in this case, because the Best-Fit transformation typically optimizes the spatial relationship between the two sets of points for 6 parameters, not 3, as the name indicates. The first three are the X, Y, and Z translations between the point sets. Pitch, Roll, and Yaw rotations are the last three. These rotations are referred by a number of different names within **SpatialAnalyzer**. Roll is most frequently referred to as rotation about the X axis (Rot X), Pitch is commonly called rotation about the Y axis (Rot Y), and Yaw is typically identified as rotation about the Z-axis (Rot Z.) Counter-clockwise is considered the positive direction of rotation.

The graphical environment in **SpatialAnalyzer** allows you to quickly see the alignment between the two set of points. In this case, the graphical view shows that the measurements have a systematic error in them; specifically the scale between the two sets is significantly different. The measurement targets are all inside the perimeter of the known reference points. This suggests that the error has a particular algebraic sign, which is a property of a systematic error. If it is systematic, we should be able to compensate for it. **SpatialAnalyzer** allows you to do that by providing a scale compensation parameter in the Best-Fit Transformation routine. A seventh parameter is used to compute the scale between to set of points. The figure shows the results of solving for the scale parameter using the best fit routine.

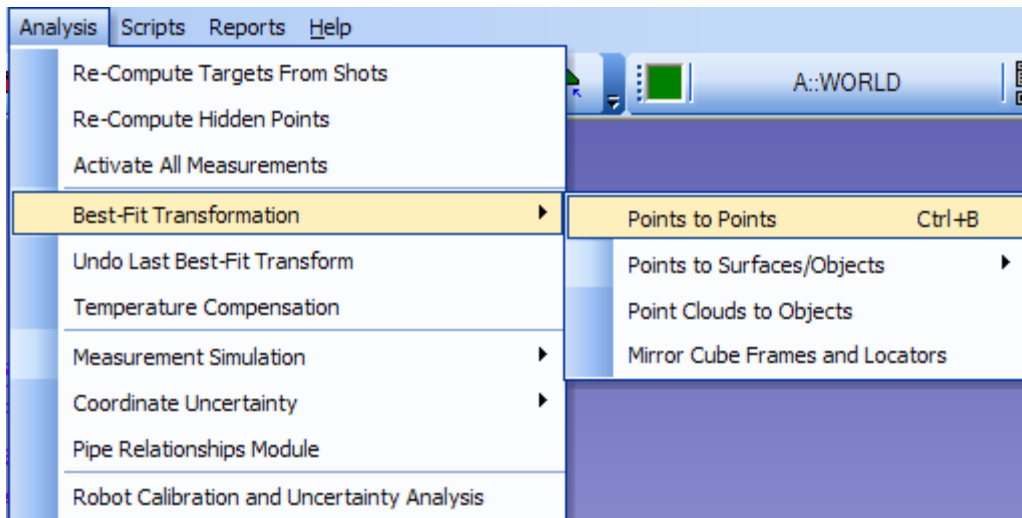


Applied 7 parameter least squares transformation, including scale

The systematic scaling, is effectively removed with this option. Care should be used when selecting the fitting options; the following sections cover these options in detail.

User Interface

Under the Analysis menu, select the Best-Fit Transformation item. This will give you several options. For this example, select points to points. There are other options that allow you to determine the transformation that will best-fit points to other geometries such as NURB surfaces. These will be discussed in other sections.



Analysis Menu

Next, you will be prompted to select the reference group. Pick the group of points that is not to be moved (or transformed), but will instead be used as a reference for the other group. Pick the corresponding group. This is the group that will be transformed to cause a best-fit of the matching points. After picking both groups, the best-fit transformation dialog will appear.

Best-Fit Transformation

Degrees of Freedom

☒ X
☒ Y
☒ Z
☐ Scale

☒ Rx
☒ Ry
☒ Rz

Set Scale

Tolerance

0.02

Apply

Columns

☒ Nominals
☐ Actuals
☐ Weights

Reporting

Append to Report

Apply Transformation

Cancel

Results	X	Y	Z	Mag.
Count	6	6	6	6
Max Error	630.3333	302.1552	122.9999	709.7503
RMS Error	461.3938	191.1082	122.9962	514.3293
StdDev Error	505.4316	209.3485	134.7356	563.4195
Max Error (adj)	630.3333	302.1552	122.9999	709.7503
RMS Error (adj)	461.3938	191.1082	122.9962	514.3293
Transformation				
Translation	0.0000	0.0000	0.0000	0.0000
Rotation (Fixed XYZ)	0.0000	0.0000	0.0000	
Rotation (Euler xyz)	0.0000	0.0000	0.0000	
Rotation (Angle axis)	0.0000	0.0000	1.0000	0.0000
Scale Factor				1.000000
Matrix	1.0000	0.0000	0.0000	0.0000
	0.0000	1.0000	0.0000	0.0000
	0.0000	0.0000	1.0000	0.0000
	0.0000	0.0000	0.0000	1.0000


Na...	Nom X	Nom Y	Nom Z	dX	dY	dZ	dMag
<input checked="" type="checkbox"/> P0	17.2213	11.4281	168.1351	-257.8643	21.8310	-122.9999	286.5302
<input checked="" type="checkbox"/> P1	17.0409	11.4281	57.7048	-257.6424	21.6578	-122.9991	286.3170
<input checked="" type="checkbox"/> P2	299.6820	1.5856	165.4944	-627.8245	300.4198	-122.9935	706.7836
<input checked="" type="checkbox"/> P3	301.5355	1.5856	58.3801	-630.3333	302.1552	-122.9941	709.7503
<input checked="" type="checkbox"/> P4	136.6397	11.4281	179.3492	-418.1855	134.0203	-122.9973	456.0361
<input checked="" type="checkbox"/> P5	139.4516	11.4281	45.6748	-421.9822	136.6642	-122.9935	460.2971

Best Fit Transformation (Points to Points) Dialog

Controlling Best-Fit Transformations

Control over the best-fit transformation is accomplished with a number of items in the Best-Fit Transformation dialog. The results are presented in the same dialog which allows you to try different options and see the effects before apply the transformation.

By default it shows you the difference between the two groups of points before a transformation. To solve for the best-fit orientation between the two groups hit the Re-fit

button . This will compute the transformation, and update the results.

SpatialAnalyzer User Manual

239

Best-Fit Transformation

Degrees of Freedom

☒ X
☒ Y
☒ Z
☐ Scale

☒ Rx
☒ Ry
☒ Rz

Set Scale

Tolerance

0.02

Apply

Columns

☒ Nominals
☐ Actuals
☐ Weights

Reporting

☒
☐
☐

Append to Report

☒
☐
☐
☐

Apply Transformation

Cancel

Results

	X	Y	Z	Mag.
Count	6	6	6	6
Max Error	0.0038	0.0029	0.0026	0.0049
RMS Error	0.0019	0.0016	0.0013	0.0028
StdDev Error	0.0021	0.0017	0.0014	0.0031
Max Error (all)	0.0038	0.0029	0.0026	0.0049
RMS Error (all)	0.0019	0.0016	0.0013	0.0028
Transformation				
Translation	-96.4499	-203.2764	123.0400	256.4424
Rotation (Fixed XYZ)	0.0018	-0.0099	-110.0301	
Rotation (Euler xyz)	-110.0301	0.0017	-0.0099	
Rotation (Angle axis)	-0.0001	-0.0001	-1.0000	110.0301
Scale Factor				1.000000
Matrix	-0.3425	0.9395	0.0000	-96.4499
	-0.9395	-0.3425	0.0002	-203.2764
	0.0002	0.0000	1.0000	123.0400
	0.0000	0.0000	0.0000	1.0000

Na...	Nom X	Nom Y	Nom Z	dX	dY	dZ	dMag
<input checked="" type="checkbox"/> P0	17.2213	11.4281	168.1351	0.0012	-0.0013	-0.0003	0.0018
<input checked="" type="checkbox"/> P1	17.0409	11.4281	57.7048	0.0014	-0.0000	0.0005	0.0015
<input checked="" type="checkbox"/> P2	299.6820	1.5856	165.4944	0.0007	-0.0019	-0.0005	0.0020
<input checked="" type="checkbox"/> P3	301.5355	1.5856	58.3801	-0.0011	0.0009	-0.0011	0.0018
<input checked="" type="checkbox"/> P4	136.6397	11.4281	179.3492	-0.0038	0.0029	-0.0012	0.0049
<input checked="" type="checkbox"/> P5	139.4516	11.4281	45.6748	0.0016	-0.0006	0.0026	0.0031

Best Fit Transformation Dialog after solving

Set the tolerance to 0.003-inches [0.075 mm] and press the Apply button. This will update the tolerance coloring. You should see several points colored as out of tolerance.

SpatialAnalyzer User Manual

240

Best-Fit Transformation

Degrees of Freedom
☒ X ☒ Y ☒ Z ☐ Scale
☒ Rx ☒ Ry ☒ Rz

Tolerance

Columns
☒ Nominals ☐ Actuals ☐ Weights

Reporting
 ☐ Append to Report


Results	X	Y	Z	Mag.
Count	6	6	6	6
Max Error	0.0038	0.0029	0.0026	0.0049
RMS Error	0.0019	0.0016	0.0013	0.0028
StdDev Error	0.0021	0.0017	0.0014	0.0031
Max Error (all)	0.0038	0.0029	0.0026	0.0049
RMS Error (all)	0.0019	0.0016	0.0013	0.0028
Transformation				
Translation	-96.4499	-203.2764	123.0400	256.4424
Rotation (Fixed XYZ)	0.0018	-0.0099	-110.0301	
Rotation (Euler xyz)	-110.0301	0.0017	-0.0099	
Rotation (Angle axis)	-0.0001	-0.0001	-1.0000	110.0301
Scale Factor				1.000000
Matrix	-0.3425	0.9395	0.0000	-96.4499
	-0.9395	-0.3425	0.0002	-203.2764
	0.0002	0.0000	1.0000	123.0400
	0.0000	0.0000	0.0000	1.0000

Na...	Nom X	Nom Y	Nom Z	dX	dY	dZ	dMag
<input checked="" type="checkbox"/> P0	17.2213	11.4281	168.1351	0.0012	-0.0013	-0.0003	0.0018
<input checked="" type="checkbox"/> P1	17.0409	11.4281	57.7048	0.0014	-0.0000	0.0005	0.0015
<input checked="" type="checkbox"/> P2	299.6820	1.5856	165.4944	0.0007	-0.0019	-0.0005	0.0020
<input checked="" type="checkbox"/> P3	301.5355	1.5856	58.3801	-0.0011	0.0009	-0.0011	0.0018
<input checked="" type="checkbox"/> P4	136.6397	11.4281	179.3492	-0.0038	0.0029	-0.0012	0.0049
<input checked="" type="checkbox"/> P5	139.4516	11.4281	45.6748	0.0016	-0.0006	0.0026	0.0031


Best Fit with Tolerances

Uncheck P4, the point that is out of tolerance. This will invalidate the fit results.

Press Re-Fit. 

Press  This will display a brief list of tips for using other features of this interface.
 Press OK to leave this dialog

Highlight P1 and P2 of the other points. Do this by clicking on P1, then holding Control and clicking on P2

Press  This will allow you to edit the properties for both of the selected items.
 Notice only cells that are identical for all those selected are populated.

Set the Y coordinate weight to 0, and the Z weight to 0.2. Press OK. This will invalidate the fit results



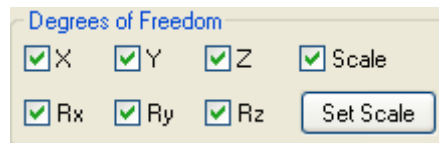
Press Re-Fit.

Click in the point list outside the populated area. This will remove the selection and reveal the point coloring of the edited points. Notice that the Y 0.00" [0 mm] coordinate and error is displayed in light-gray. This means the weight is zero. Also notice the Z 20.0" [500 mm] values are blue. This means the weight is other than 1 or 0.

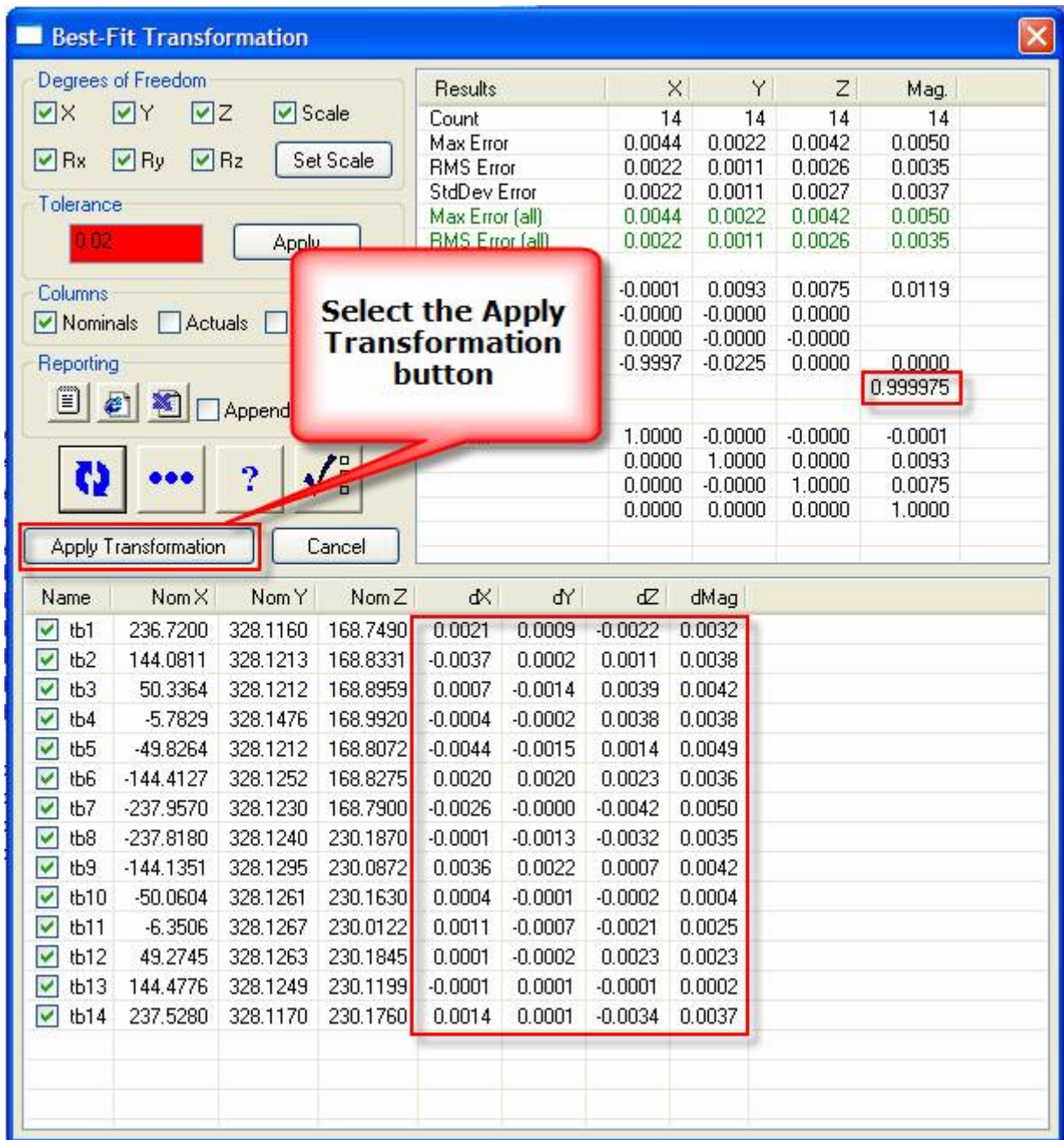
Hover over one of the blue Z cells. It will show you the actual weight values.

Best-Fit Scale

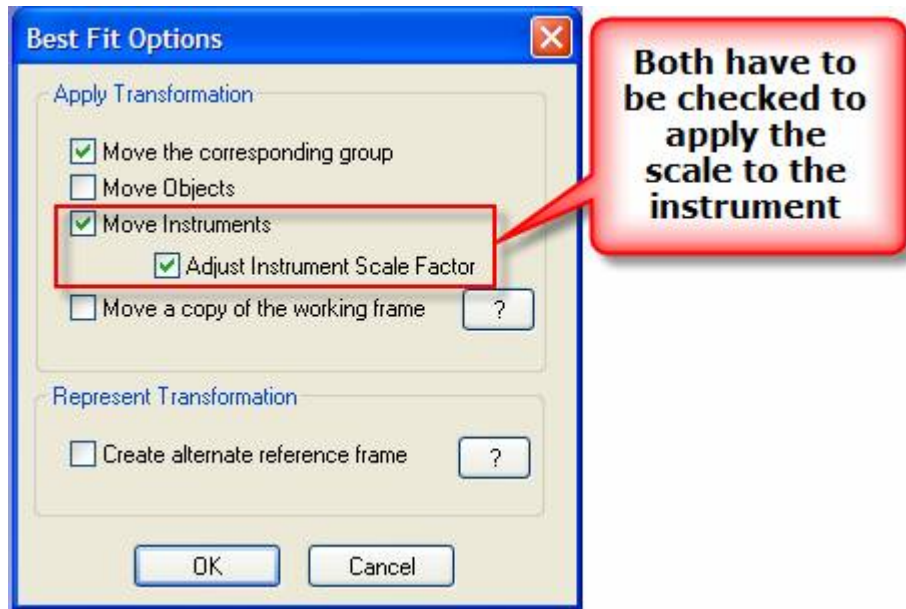
The points to points best fit is used to solve for scale differences between the point sets. Apply the scale factor to the instrument station for all prior and subsequent measurement to be scaled to the reference points.



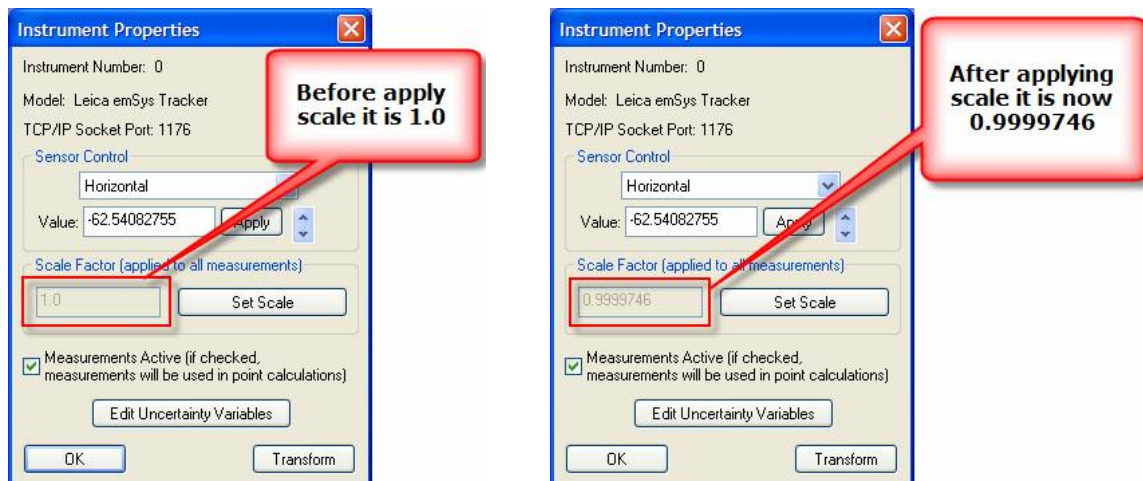
There are two ways to perform the Pts to Pts best fit. One is with the Instrument >> Locate >> Best Fit and the other is with the Analysis >> Best Fit Transformation >> Points to Points. The process outlined below uses the Analysis functions.



The scale factor is applied to corresponding/measured groups, instruments and objects. The choice for applying the transform including the scale factor is made on the Best Fit Options dialog. It is presented after selecting the button.



If the Adjust Instrument Scale Factor is not checked the instrument scale remains at 1.0. The optional Adjust Instrument Scale Factor was not checked the example on the left below. After apply the transform this can be confirmed by checking the instrument properties dialog. In this case the instrument scale is 1.0. The example on the left shows the difference when applying the scale factor to the instruments. The instrument properties dialog shows its scale is 0.999975.

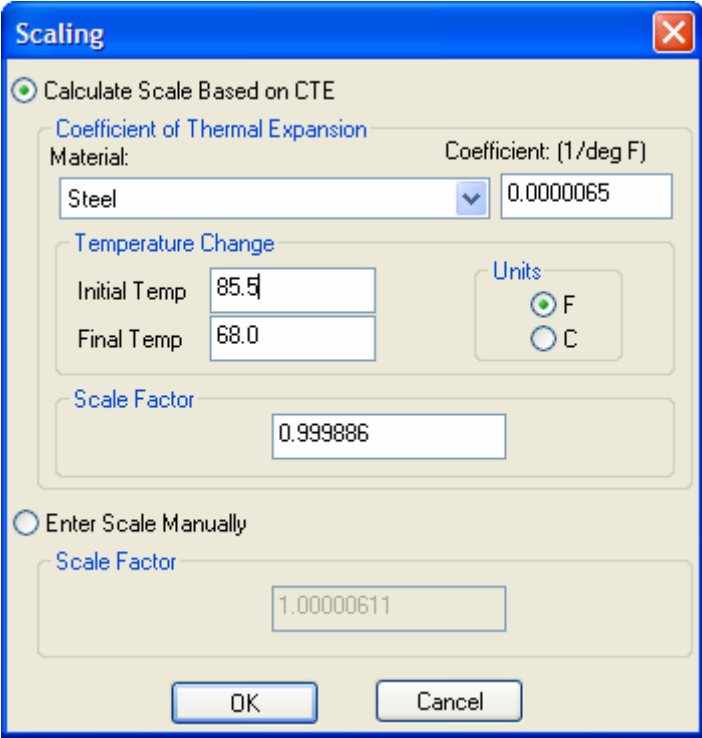


The Instrument >> Locate >> Best Fit function automatically applies the scale factor to the instrument.

Best-Fit Set Scale

A Set Scaling function is available during the best fit process. This scaling process is either one computed based on material CTE and a temperature delta or by manually entering a scaling factor. To use either of these methods select the **Set Scale** button

on the Points to Points Best Fit Transformation dialog. The set scaling dialog is shown below.


The image shows a 'Scaling' dialog box with a blue title bar and a close button. It has two radio buttons: 'Calculate Scale Based on CTE' (selected) and 'Enter Scale Manually'. Under 'Calculate Scale Based on CTE', there is a 'Coefficient of Thermal Expansion' section with a 'Material' dropdown set to 'Steel' and a 'Coefficient: (1/deg F)' text box with '0.0000065'. Below this is a 'Temperature Change' section with 'Initial Temp' (85.5) and 'Final Temp' (68.0) text boxes, and a 'Units' section with 'F' (selected) and 'C' radio buttons. A 'Scale Factor' text box shows '0.999886'. Under 'Enter Scale Manually', a 'Scale Factor' text box shows '1.00000611'. At the bottom are 'OK' and 'Cancel' buttons.

After setting the scaling factor select the OK button.

Best-Fit Results Reporting

To create a report of the transformation results. We can always check the Append to Report option in the dialog. This will put the report at the end of the SA Fit/Query report. If, however, you want a report with a different format, we can use the ASCII, HTML, and EXECL options built into the best-fit dialog.

The options available in this section provide a number of output possibilities.


The image shows a 'Reporting' section with three icons: a document, a document with a magnifying glass, and a document with a cross. To the right of these icons is a checkbox labeled 'Append to Report'.

Best Fit Reporting Options

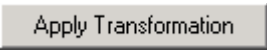

The output includes a listing of each point; fit status, deviations, weights, and the fit results.

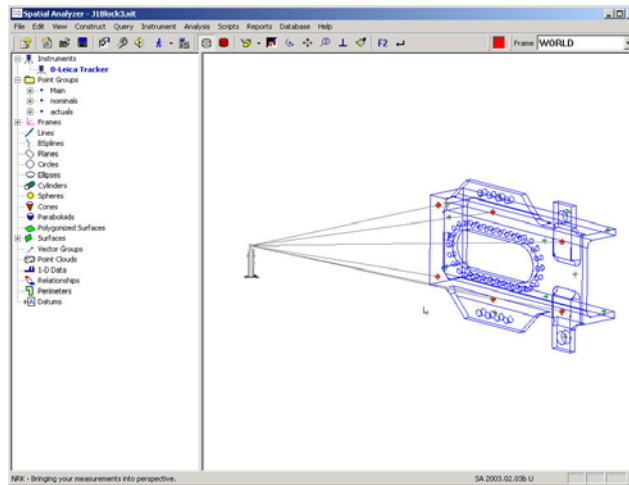
Control Best-Fit Report Details

First, let's show a little more information in the point list. Check ☐ Actuals to turn on the Actuals column. Also press ☐ Weights. You can resize the dialog to show the entire table, but this is not necessary for the report.

Press , the ASCII text report option. Pick a filename. The file will be written, then notepad will open and display its contents. The same applies for HTML, and EXCEL.

Applying Best-Fit Transformations

Now, press . This will close the dialog, and transform the instrument into the best-fit location. Back on the main toolbar, press Autoscale, , to show the instrument and the part in the graphical view.



Results after apply Best-Fit

This dialog will allow you to change the point selection, control the weighting of each coordinate in the fit, and it also reports the results relative to the current working frame in the lower left corner. Each component of the dialog will be discussed in the following section.

Best-Fit Dialog Functionality

This section will detail each component of the dialog and describe its functionality. The dialog is divided into four basic sections; a grid where the points and their respective residuals and weights are available for you to control is the first area. A results section is shown in the lower left corner. There are two output buttons on the right of the dialog, Chart Residuals and Print Table. The Compute button allows you to re-compute the fit after making changes. Finally, you can choose to apply the fit or Cancel it with Apply and Cancel buttons, respectively.

Grid (Points, Deviations, Weights, and Error)

The points that have common names between the groups are shown in the grid control in the middle of the dialog. Weighting allows you to weight the individual components for each point independently, or you can choose to weight the total importance of the point. The X -100" [-254 mm], Y 0.00" [0 mm], and Z 20.0" [500 mm] fields allow you to apply a weighting to points by axis.

Excluding and re-including points from the fit is controlled with the check box next to a point's name. Hitting the Compute button after making changes re-optimizes the entire fit with only those points with a checked box. No contributions are used from points that do not have their boxes checked. To add the point back in, re-check the box, and then hit the Compute button again. The default condition is to use all of the points that have the same names.

Checking the Allow Scale Change check box (above the grid) and hitting the Compute button lets the best-fit transformation routine optimize for the scale between the two sets of points. The scale parameter is known as the 7th parameter. To return to a non-scaled state, un-check the Allow Scale Change check box and hit the Compute button. The scale parameter will be reset to 1.000000.

Each of the points in the fit has three degrees of freedom, and therefore you can assign a weight to each of them using the Weighting grid columns. A weight is also provided for the total. If a weighted fit is needed, typically either the components (i.e., X -100" [-254 mm], Y 0.00" [0 mm], and Z 20.0" [500 mm] or the total is weighted.

Weighting mechanisms are typically explained with an example. **SpatialAnalyzer** assumes a default weight of 1.0. So if a reference point's X -100" [-254 mm] axis value is not known with much confidence, weighting that point's X -100" [-254 mm] component with a low number (e.g., 0.001) would let the best fit routine get the other components closer to their true values. Conversely, if the X -100" [-254 mm] axis values for a point is known with a lot of confidence, (more than the other components of the point), then weighting it with a higher number (e.g., 1000) forces the best fit to pay more attention to it. Basically it comes down to a higher weight means you have more confidence in it, and therefore the best fit routine is going to pay more attention to it. A low weight on a component allows it to float, i.e., the routine doesn't try hard to make it match the estimate.

Results Section

The transformation results section shows the computed best X -100" [-254 mm], Y 0.00" [0 mm], and Z 20.0" [500 mm] translation, with the relative change in scale, and the three rotations needed to optimize the position and orientation between the two sets of points. These results are relative to the current working frame.


Results	X	Y	Z	Mag.
Count	6	6	6	6
Max Error	0.0016	0.0007	0.0019	0.0020
RMS Error	0.0009	0.0000	0.0006	0.0013
StdDev Error	0.0010	0.0000	0.0006	0.0014
Max Error (all)	0.0016	0.0007	0.0019	0.0020
RMS Error (all)	0.0009	0.0000	0.0006	0.0013
Transformation				
Translation	-96.4496	-203.2765	123.0420	256.4434
Rotation (Fixed XYZ)	0.0016	-0.0102	-110.0301	
Rotation (Euler xyz)	-110.0301	0.0020	-0.0101	
Rotation (Angle axis)	-0.0001	-0.0001	-1.0000	110.0301
Scale Factor				1.00000
Matrix	-0.3425	0.9395	0.0000	-96.4496
	0.0000	0.0000	0.0000	203.2765

Results section of the Best Fit Transformation (Points to Points) Dialog

The initial and current maximum errors are displayed with the initial and current RMS errors on the last two lines of the Results section. These results will be appended to the Fit/Query Report if you select the Append to Report check box before the transformation is applied.

Compute Button

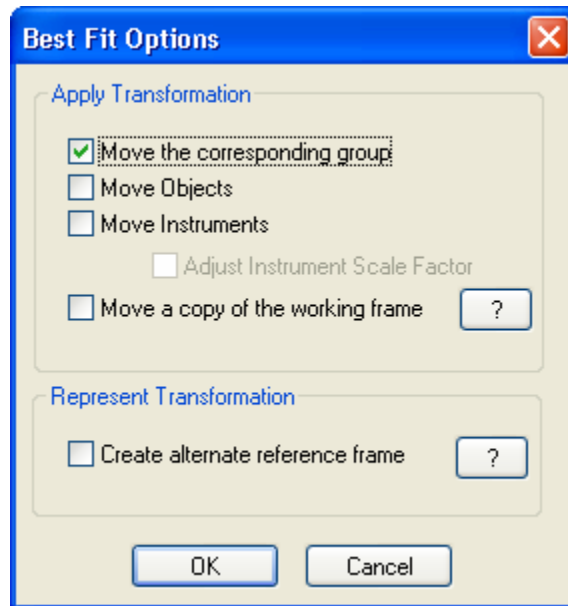


Once all the settings have been properly configured, pressing the  button will re-run the optimization, with your new settings. The grid, results, and chart objects will all be updated.

There are several criteria available to control the subtleties of the fit, they include weighting, changing the scale, removing and adding points into the fit. If changes are made to these attributes, then the use the Compute button to update the fit before applying it.

Apply Best Fit Transform

The Apply button closes the best-fit transformation dialog and initiates the application of the transformation. A dialog is presented to identify the objects to which the transformation will be applied. These options are discussed below:



Applying the Best Fit to Objects Dialog

- **Move the Corresponding Group:** This is checked by default. It transforms the corresponding group of points using the computed transformation.
- **Move Objects:** Check this if there are other objects in the model that you wish to transform. After hitting the apply button, you will be prompted to select the objects.

- **Move Instruments:** This will allow you to bring an instrument into a part coordinate system or to bring two instruments into the same coordinate system. The Adjust Instrument scale option will also apply the scale factor (from the scale-free transformation) to the instrument. NOTE: This will affect the computation of the coordinates produced by the instrument.
- **Move a copy of the working frame:** If you need to use the transformation in a subsequent operation check this option to create a frame that will be displaced from the current working frame by the solved transformation. This option makes a copy of the current working frame then moves the copy by the transformation solved for during the best-fit. By making the copy the active frame, the values before transforming can still be reported.
- **Create alternate reference frame:** This option creates an alternate reference frame by copying the working frame and transforming it by the inverse of the transform computed in the best-fit process. This option is normally used when nothing is moved, yet the values need to be reported in the transformed orientation.

Applying the Transformation

Hitting the OK Button on the closes the best-fit transformation dialog. Next, the action you selected will begin occurring. The corresponding group will animate its move from its current location to the transformed location. Next, additional objects will be transformed. Finally, instruments will be moved. Animation is used in each of these processes to help the user visualize the transformation and in doing so provide a good check of the transformation process. If it doesn't appear to be the right move, then the user can double check the work and thereby build confidence in the process.

If you made a mistake with the options you selected, you may select the Undo Last Best-Fit Transformation option from the menu. Also, the best-fit process will perform a backup before it begins. You can also revert to the backup file.

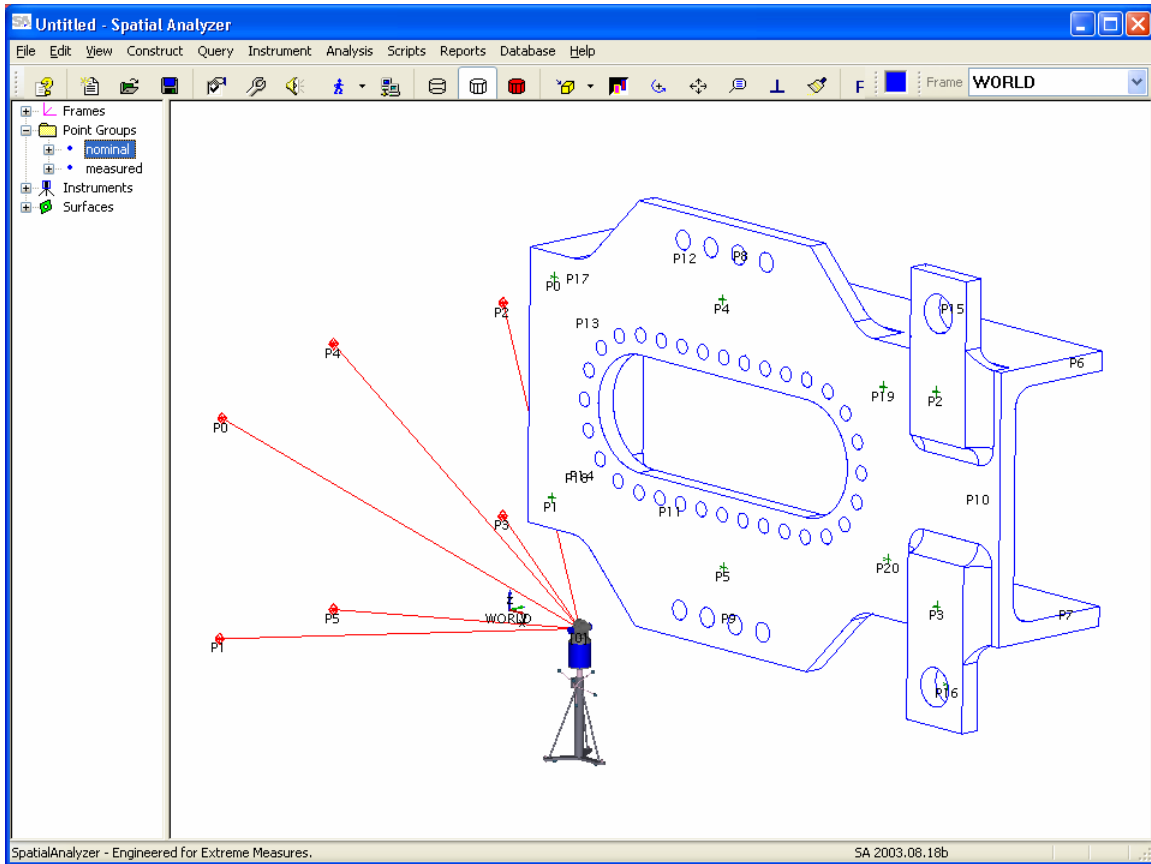
Locating an Instrument using a Best-Fit Transformation

As discussed previously, points in SpatialAnalyzer are organized into groups. This construct is particularly useful for the best-fit transformation option since the best-fit will attempt to best-fit common points using common point's names. This results in several requirements:

- The two sets of coordinates that will be best-fit must reside in separate groups.
- There must be at least three points in each group with matching names.

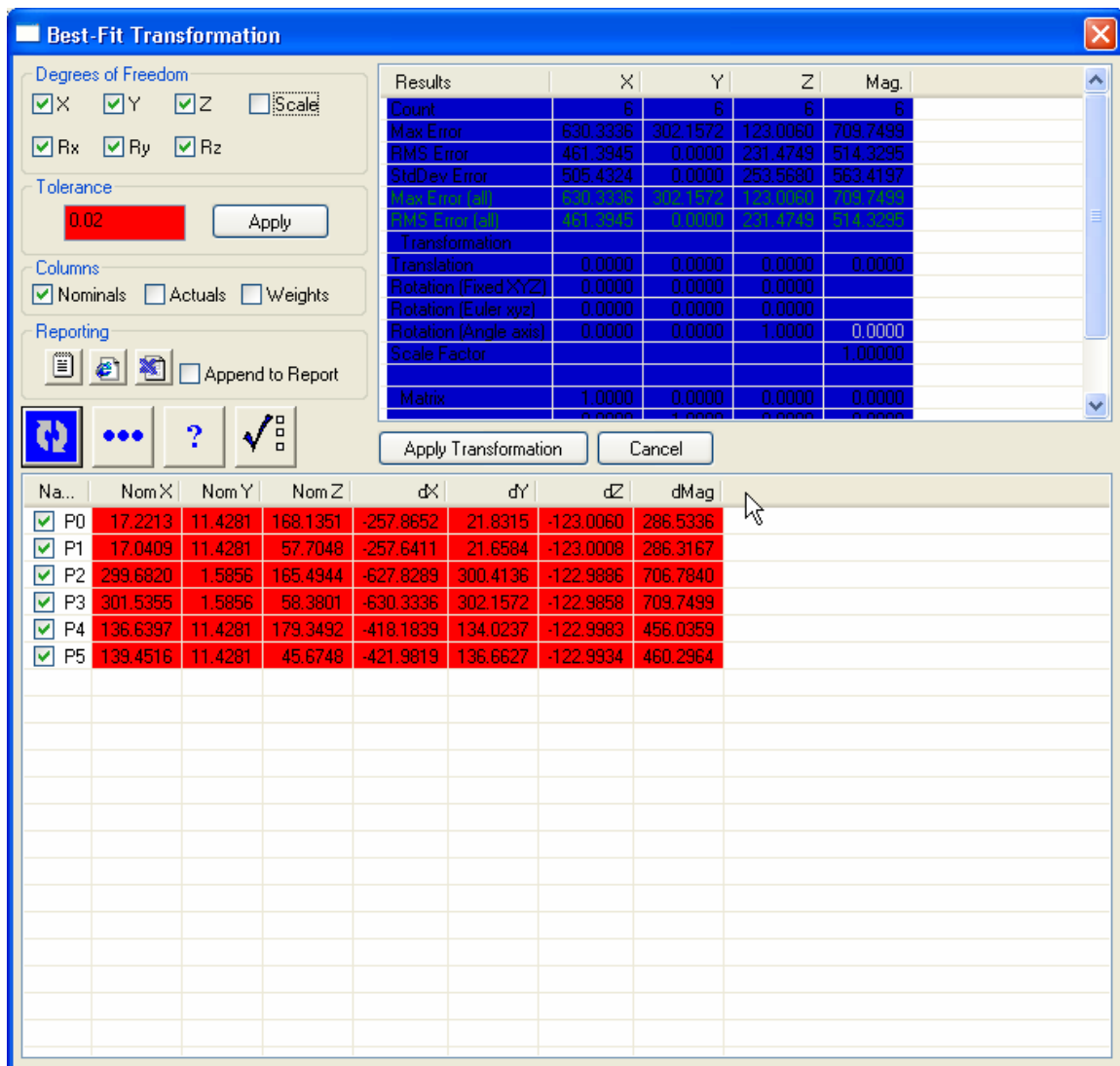
If you have two groups of points that meet these requirements, you are ready to proceed.

The following figure demonstrates a common situation. There are reference points (blue) relative to a part (or possibly WORLD) coordinate system, and measured points (red) that were observed using the laser tracker. In this case, let's suppose we want to bring the measured data into the coordinate system of the reference points for the purpose of comparison, or subsequent measurements.



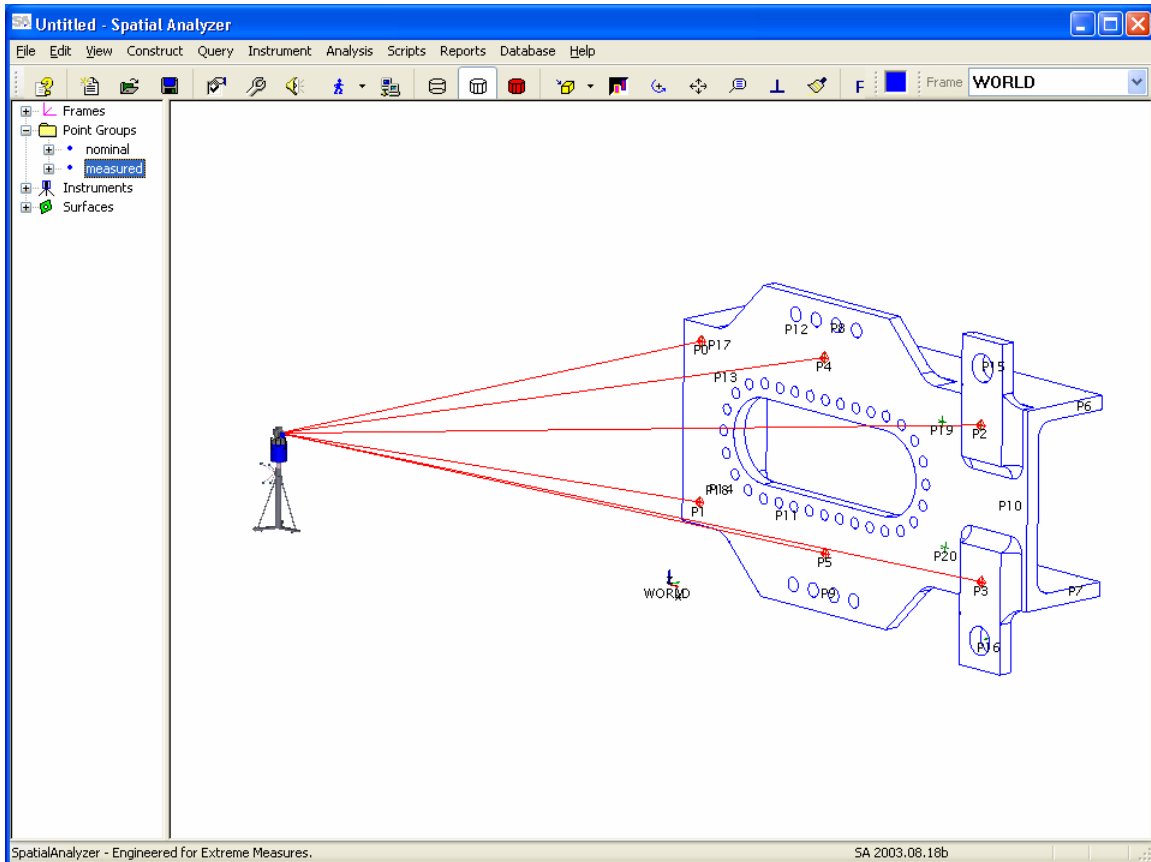
Locating an Instrument

In order to perform this best-fit transformation, select **Instrument->Locate->Best-Fit**. This will prompt you to select the reference (or nominal) group. Pick the group of points that is not to be moved (or transformed), but will instead be used as a reference for the other group. In this case, that is “Ref”. Next, pick the corresponding (or measured) group. This is the group that will be transformed to cause a best-fit of the matching points, in this case “Meas”. After picking both groups, the best-fit transformation dialog will appear:



Results of Instrument Location Best-Fit Transformation

This dialog shows the results of a best-fit transformation with the current settings. Using this interface, you can eliminate points from the fit, change their weights, allow scaling, and then re-compute the transform. There are also options for setting the weights based on instrument uncertainty or stored values. For this example, and most basic transformations, check the residuals, cull points if there is a good reason, then press Apply. The instrument will be transformed so that the point sets fit as seen in the next figure:



Graphical View of Located Instrument

At this point, the instrument is located to the nominal (reference) coordinate system. This means you can continue measuring other features and compare the results directly to other nominal features. Since you are located, you can also use watch windows to track variation in real-time, auto-measure the closest target, etc.

Advanced Spatial Transformations

Surface transformations

In previous sections, we discussed using a best-fit transformation to locate an instrument relative to a known set of nominal points. In this section, we will extend that concept and demonstrate how to locate an instrument to a known surface geometry by measuring points on the surface. This method is called a “point to surface best-fit transformation”. This is a powerful method for locating measurements to a CAD model.

SA supports different types of point to surface transformations...

Quick Align – Provides for a quick orientation by selecting 6 corresponding points on both the surface and the measured point set. This is usually used to quickly transform the instrument and measurements to the CAD part since the

part is usually in global coordinates as opposed to the measurements which are in instrument coordinates.

N-Point Full Fit – finds the optimal transformation that minimizes the errors between the measured points and the CAD surface. This method contains many options for tolerance fits and other advanced fitting methods.

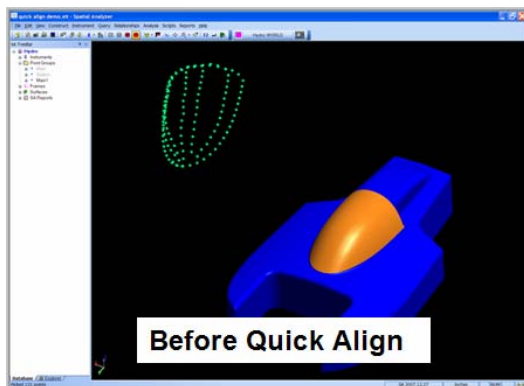
Move Objects (and Instruments) by Minimizing Relationships – moves objects and instruments based on minimizing deltas in relationships. One or more relationships and their respective weights are used to determine the optimal transformation minimizing errors modeled by the relationships. This technique is able to optimize complex configurations easily.

GD&T Align – moves objects and instruments based on minimizing measurements to GD&T datums. After measuring at least three datums select the primary, secondary, and tertiary datums. Options to select additional object to move with alignment are available on the Align dialog.

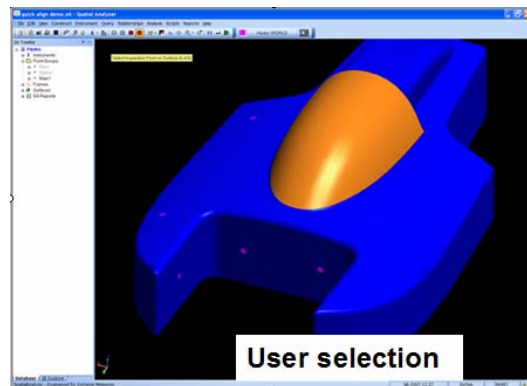
Different fit types are helpful based on the application. In some cases a Quick Align is all that is required to visualize the setup. If you plan to use only the surface geometry to locate your instrument to the CAD, you will run both of these fits. First the Quick Align to bring the instrument into a tight fit based on 6 pts and surfaces. Second, use one of the other techniques to refine the fit.

Performing a Quick Align

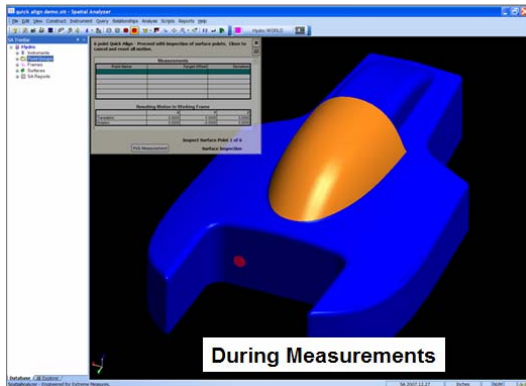
Select either Analysis >> Best-Fit Transform >> Points to Surfaces >> Quick Align or Instrument >> Locate >> Quick Align. A step by step example for the Quick Align is shown below...



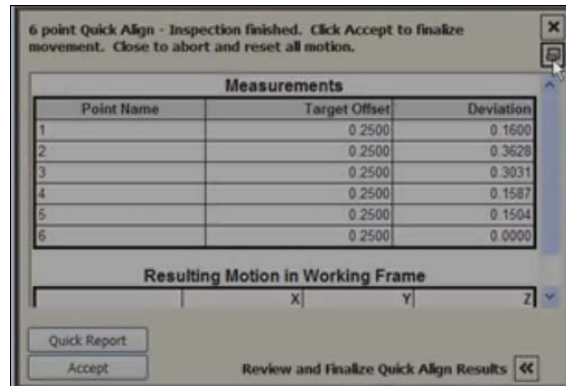
1. Before aligning... select Instrument and any additional objects to move with the instrument



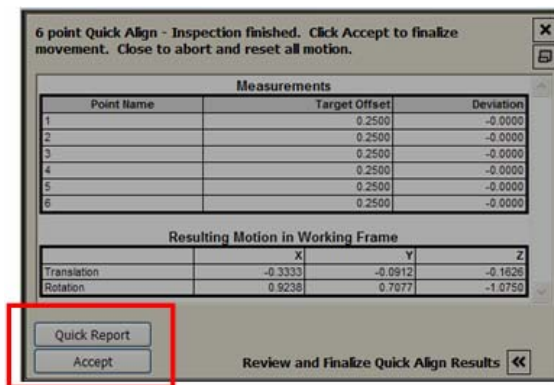
2. Select 6 points at key locations on CAD surfaces



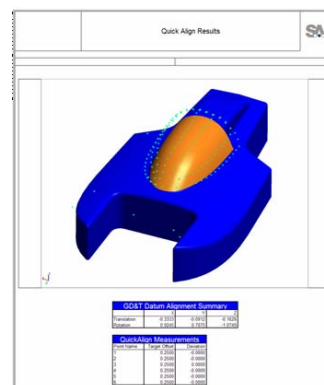
3. Start instrument interface and a single point measurement mode... The view changes to show each nominal target in sequence.



4. As each point is measured in sequence the instrument transform is updated.



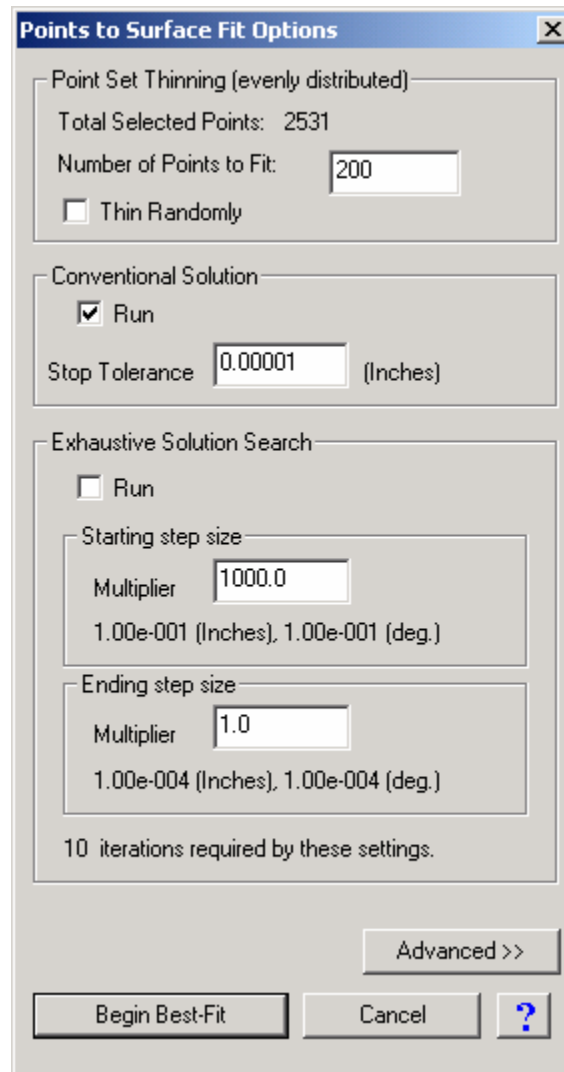
5. As measurements are recorded, fit delta's are shown in the dialog... backup and re-measure to correct errant measurements. Make a Quick Report and Accept alignment when ready.




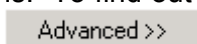
6. Make a Quick Reports to record the results

Performing the N-Point Full Fit


Once in rough position, select **Analysis->Best-Fit Transform->Points to Surfaces->N-Point Full Fit**. You will be asked to pick the points to use in the fit. After doing this, you will need to select the surfaces to drive the fit. Then, the point to surface fit dialog will appear.

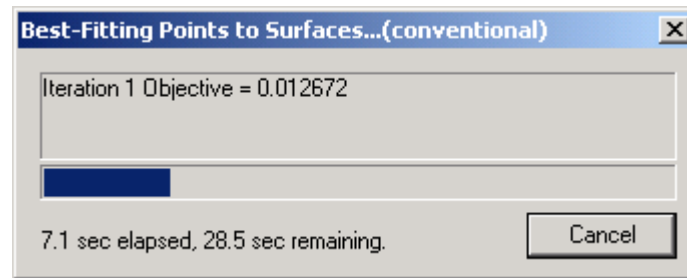


Points to Surface Fit Options Dialog

This dialog contains many options. To find out more, click on the  button. It will pull up a help page. In addition, the  button will expand the dialog to the right and display the full set of fitting parameter options. For now, let's focus on the basic operation:

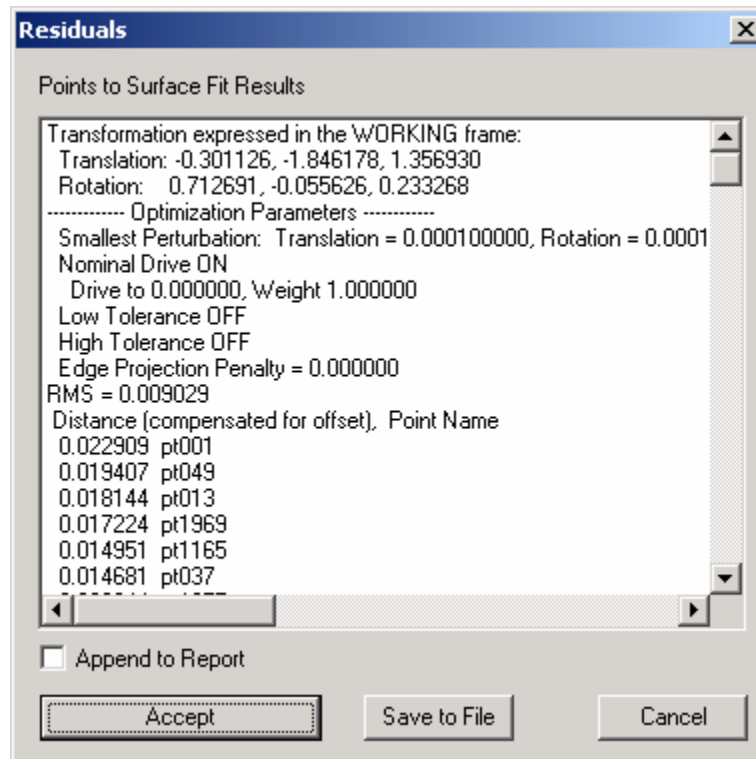
The point thinning options allow you to limit the number of points used in the full fit. Suppose, for example, you select 1,000 points for the fit. Though you can run a fit with 1,000 points, it could take awhile depending on the surface complexity and other options settings. To sub-sample the points, just enter the number you DO want to fit into the "Number of Points to Fit" field. If you had 1,000 total points and you enter 200, it will thin the data, taking every 5th point to get the requested 200 points. There is also an option to thin randomly. This will pick 200 random points from the set instead of taking every 5th point. In this case, we have 2,531 points selected. Let run a fit with 200, as shown in the previous figure.

Press  to start the optimization. You will see a progress bar showing the progress of the fit, and the objective function. It will step until the fit is optimal.



Best Fit Progress Dialog

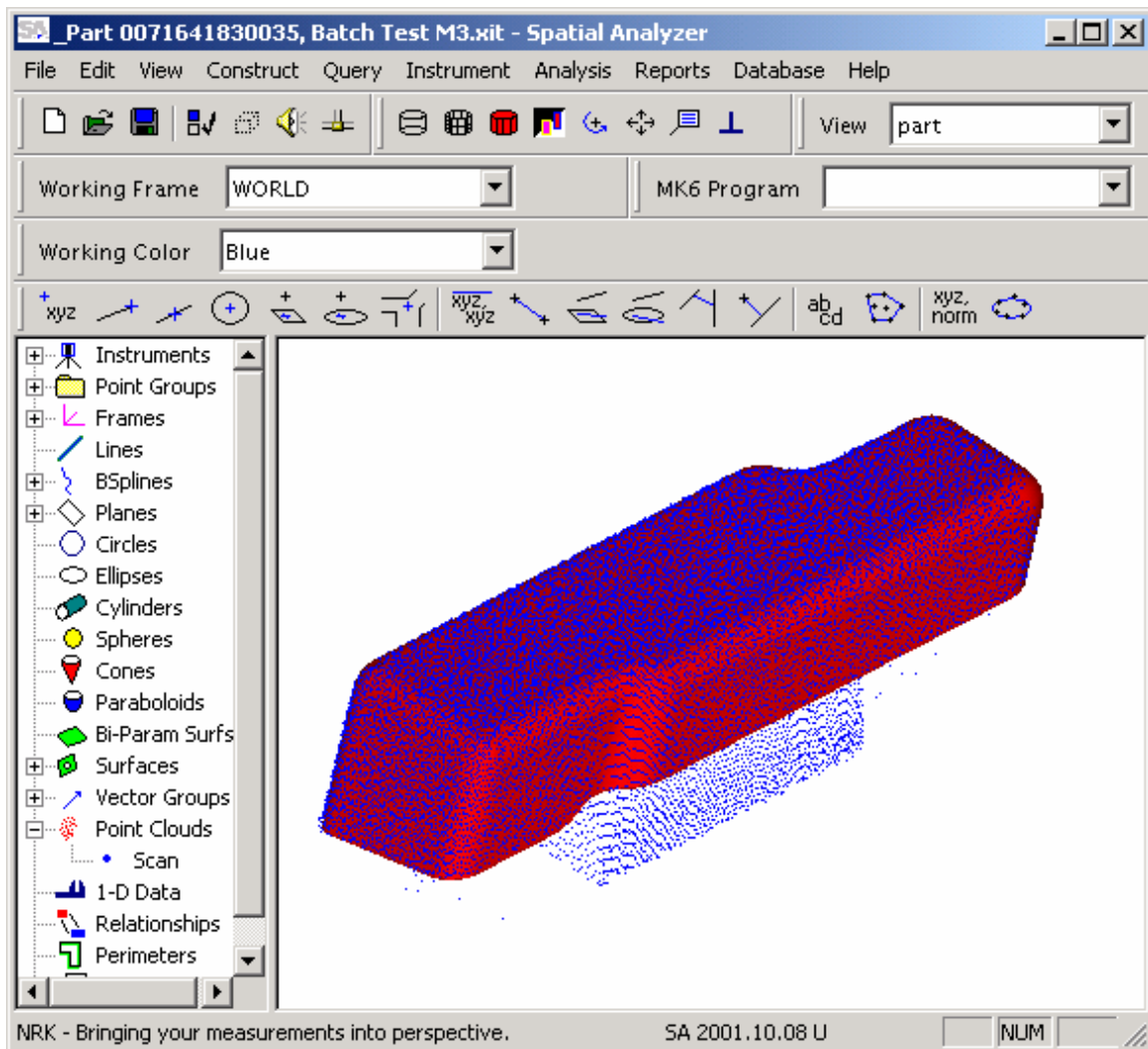
Once the fit is completed, the results dialog will appear:



Best Fit Residuals Dialog

This will show a summary of the fit parameters, the resulting transform, and the residual errors. If you do not like the fit, press Cancel and the process will stop. Or, to use the transformation, press Accept.

As with the 6-Point fit process; the N-Point Full Fit process prompts you to select the Objects to move, and then you can select which instruments to move. Select the instrument and the transform will be applied:



Graphics view of the Best Fit results

At this point, we are oriented to the coordinate system without measuring any tooling points! Instead, we have located based on CAD geometry.

Move Objects (and Instruments) by Minimizing Relationships

Aligning measurements and instruments to CAD features with relationships is relatively straight-forward. Construct relationships between measurements and their respective features. Use the Move Objects by Minimizing Relationships function to move instrument stations that made the measurements of the features.

The process works by minimizing deltas modeled in relationships. Moves (of objects and instruments) are computed to drive the delta's toward smaller and smaller values. Configure one or more of the relationship's weights to control/determine the optimal

transformation based on which features are most critical. This technique is able to optimize complex configurations easily.

GD&T Align

Measurements of datum made by Instruments are aligned to the feature with the right-click “Align” function on the GD&T Datum treeview. Select the primary, secondary, and tertiary datums from a list of measured datum. Options to select additional objects to move with the alignment are available on the Align dialog. A Quick Report option is also available with align function.

BUNDLE ADJUSTMENT

A bundle adjustment may be used to refine measurements acquired using a spherical measurement system. These include theodolites, laser trackers, laser scanners, etc. Basically, a bundle adjustment will attempt to find the instrument transformations (positions and orientations) that yield the minimum combined pointing error.

Initial Guess

Since the bundle adjustment algorithms use the current locations of the instruments as a starting guess, it is best to “rough” position the instruments. To do this, either enter the instrument transformations manually, or use the Drag Instruments function from the Instruments menu. It is best to place the instruments near their final location, as this will speed the bundle solution. If using theodolites with level compensators, you will use the option to force instrument vertical during the bundle adjustment. If this is the case, the virtual instrument models should be perfectly parallel to each other before initiating the bundle adjustment.

Scale-Bar Database

For instruments that measure angles and distances, it is not necessary to use scale-bars in the bundle process. They are required for theodolite networks, however, because without them, the instruments will scale freely.

The Spatial Analyzer contains a scale-bar database. This is where you can enter the various scale-bars you have placed into the measurement area. Essentially what this database does is it associates target names with a scale-bar distance and uncertainty.

To modify the database, select the Scale-Bars tab from the User Options page. The next window will list all of the current scale-bars and their information.

To add a scale-bar, press the Add button and a new record will be generated. Complete the information required and press the Apply button. You should see the information in the list window update.

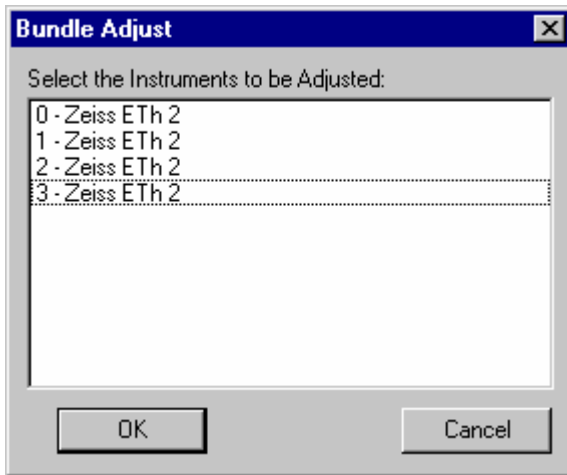
Since an accurate bundle solution for theodolites requires at least one scale-bar, make sure this database is populated before attempting a bundle. In addition, make sure the targets for the end of the scale-bar have been shot and included in the list of targets to be included in the bundle solution.

Note, however, that you may populate the scale-bar database at anytime before attempting to bundle adjust. It is not necessary to have this database in place before taking measurements.

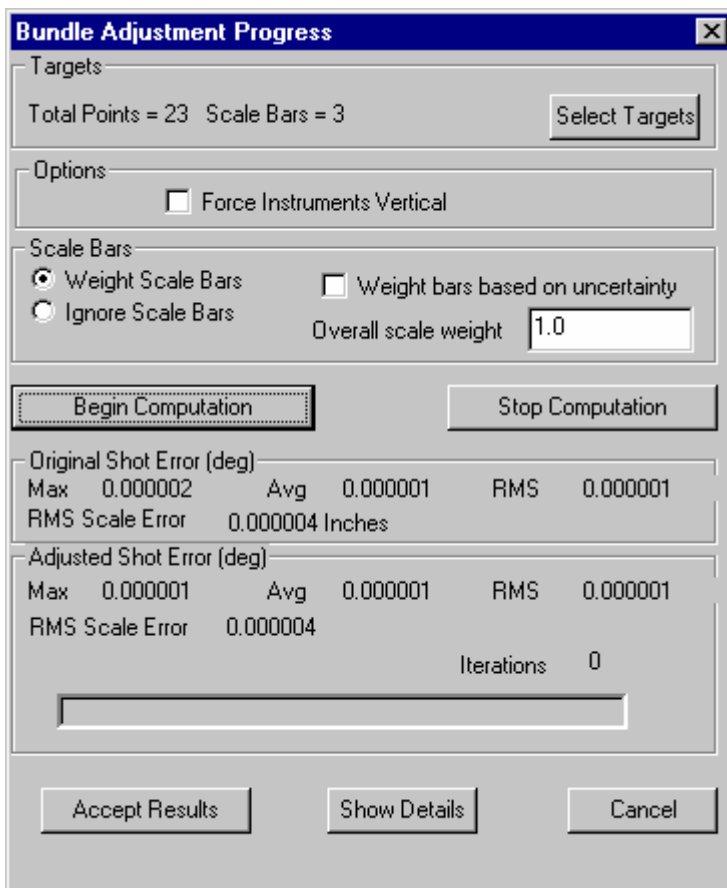
Running the Bundle Algorithm

Select Bundle Adjust from the Instruments menu. Next, select the instruments you wish to bundle using the dialog. The instruments that you select are the instruments that will be moved in order to refine the network solution. If you have 4 instruments, you will only have to bundle 3 because the first instrument may remain fixed while the others are adjusted relative to it. You may, however, select all of the instruments. This will make

the solution take longer, however, because the entire network will be allowed to float in space. Next, the Bundle Adjustment Dialog will appear.



Bundle Adjustment Instrument Selection



Bundle Adjustment Dialog

This dialog will allow you to configure the settings for the bundle, run the optimization, and view the results.

Selecting Targets to Bundle

Once the Bundle Adjustment window appears, you may select the targets to be included in the bundle. The Spatial Analyzer will default to including ALL of the targets. You may reduce the number of targets for computational efficiency or target accuracy reasons.

Scale-Bar Weight Factors

In the Bundle Adjustment window, you will notice a section for Scale Bar Weight factors. This section allows you to control the manner in which scale-bars are incorporated in to the final solution.

You may change the state of the check-box to determine whether scale-bars are weighted based on their uncertainty. If this box is checked, the relative uncertainties in the Scale-Bar database will be used to weight those scale-bars with a higher accuracy more than those scale-bars with a lower accuracy (or a higher uncertainty). Basically, this check-box controls how multiplier scale-bars are compared to each other during the bundle solution.

In addition, you may enter a numerical weight that is applied to ALL scale-bars during the solution. This value defaults to 1.0, but may be increased or decreased at the user's discretion. Note, however, that this value is independent of the state of the check-box. You may weight scale-bars based on uncertainty and/or apply an overall weight. This is because the overall weight is applied after the uncertainty weight is applied.

Viewing System Errors

The Bundle Adjustment window displays both the Original shot errors and the adjusted shot errors. This gives you a measure of how much the bundle adjustment has improved the overall system error.

A maximum, average, and RMS error is displayed. These values characterize the combined pointing errors for each target that is included in the Bundle Adjustment. Each target has a pointing error, given in degrees. This is the amount that the measurements miss the best-fit computed target. The bundle adjustment attempts to minimize the combined pointing error by simultaneously adjusting both the position and orientation of the instruments.

Therefore, if there are 30 points in the bundle, the maximum error is the largest pointing error for all of the points, the average is the average of that set, and the RMS is the root-mean-square of the set.

If you wish to see the errors itemized by target, press the Show Details button, and you will see a list of the targets and their errors. You may wish to select this option before beginning the bundle to inspect the list.

Beginning Computation

Press the Begin Computation button, and the Bundle Adjustment will begin. As progress is made, the graphical display will be updated, and all of the numerical values in the Bundle window will reflect the system improvement.

When the algorithm has found a minimum, it will stop and issue a message to the user. Look at the details of the bundle to inspect the individual target errors as well as the scale-bar errors. If these are suitable, press the Accept Results button in the Bundle window, and the changes will be made active. The system is now bundled!

If, however, the results are not satisfactory, and you wish to start over where you began initially, select Cancel from the Bundle window. This will reset the instruments to their initial locations and re-compute the target locations.

Local Minima

Whenever you attempt to minimize a set of complex equations, the issue of local minima arises. Because you start from an initial guess and seek to minimize the error until you can minimize no more, you do not know that you are at the lowest possible minima for the entire solution space.

For this reason, it is possible for the bundle adjustment algorithms to find a solution that, while it is a minima, is not the minima you were searching for.

SpatialAnalyzer contains a full 3-D graphical model of the measurement environment will help to decipher some of the local minima. It is important, however, to start the instruments near to their optimal location (i.e. use a good guess) to minimize the chance of a local minima.

Weighting

There are several weighting factors described above. These control the importance of different parameters on the overall optimization process. There are several other weighting factors outside this dialog that control the process as well.

In the User Options dialog, under the Units tab, there is a button for Angle-Distance Weights. This button will allow you to control the relative weight of angular measurements compared to distance measurements. This will only affect the bundle if you are using instruments that can measure distance (laser trackers for example).

UNIFIED SPATIAL METROLOGY NETWORK

Large-scale manufacturing operations are increasing their reliance on portable coordinate measurement devices. These devices include electronic theodolite networks, total station systems, video photogrammetry systems, laser tracking systems, laser scanning systems, and portable coordinate measuring machines. This increase in portable metrology is driven by cost and efficiency. Commercial airframe construction is a good example of this trend. Instead of relying on elaborate holding fixtures and precise tooling, manufacturers are using the component parts as the tooling, and verifying design conformance with portable coordinate measurement devices⁵. This design paradigm shift eliminates the need to manufacture and maintain complex, costly check-fixtures. It does, however, introduce portable metrology into the production process and therefore requires a rigorous accounting of the uncertainty in the design conformance measurements.

Much work has been done to understand and quantify the performance of various measurement systems. The manufacturers of the measurement devices publish performance specifications. These laboratory specifications are not, however, representative of the actual performance of the instrument in typical in situ conditions. Many significant effects are ignored including operator contributions to the uncertainty and the variability of real-world shop floor measurement environments. In addition, the manufacturer uncertainty statements often do not address the geometric nature of the coordinate uncertainty but instead provide a volumetric statement based on a spherical uncertainty at each point.

Many large-scale measurement processes require more than a single measurement instrument. Examples include commercial airplane production and shipbuilding. These applications necessitate either a combination of various measurement devices or the relocation of a single instrument throughout the measurement volume in order to acquire the necessary data. Users often combine measurement systems by tying individual measurement systems together based on common reference points, and then assume they are still working within the instrument's published uncertainty. Alternatively, many users apply heuristics to determine the uncertainty as they progress along a chain of measurements. These methods provide poor approximations of uncertainty in all but the most simplistic cases. Even in cases where only a single placement of an instrument is used, its measurements are typically tied in to a reference coordinate system. The uncertainty from this tie-in process is often ignored.

The ISO (International Organization for Standardization) standards focusing on "Global Product Specification" require that part measurements be described by two numbers⁶. The first is the result of the measurement and the second is the stated uncertainty. This uncertainty statement represents the estimated variability in the result. This specification mandates uncertainty statements in order to provide traceability for measurement results. In addition, it is recommended that measurement systems provide uncertainty statements in order to be considered "accredited" systems (Forbes and Harris, 2000).

⁵ Muske et. al., 1999

⁶ ISO, 1995; Swyt, 2000

The National Institute of Standards and Technology states that “a measurement result is complete only when accompanied by a quantitative statement of its uncertainty.”⁷ This is derived from the ISO⁸ and the corresponding American National Standards Institute publication.

These requirements and recommendations reflect good practice in the production environment. Expensive part rework and costly delays can result from the decisions that are made based on unreliable coordinate measurement data. These decisions should be backed by a rigorous knowledge of the uncertainty in the measurements that are used to pass or fail parts.

Basic USMN Process

The primary output from USMN is a Composite Point Group. This group contains the optimal point coordinates after solving the network. Inputs to USMN include the instruments and their measurements. The measurements are from multiple stations to common targets. In other words, observations need to be made from different stations to a set of commonly named target locations. The observations from different stations are made into different point groups but have the same target name.

USMN manipulates instrument positions to get minimal measurement closures on the common points. The optimization process uses the estimates of the instrument uncertainties and the range of each observation to weight the individual contributions for each measurement.

There are three individual processes in USMN which are used for three distinct purposes on the measurement network. The first of these functions solves the network. It optimizes the complex multi-station measurement network to solve for the best station and target locations. It produces the best point coordinates from the measurements.

The second component analyzes instrument performance within the network or as USMN refers to it uncertainty. With the optimal network (stations and target locations) USMN computes instrument uncertainty estimates based on the measured network of targets. This analysis characterizes and produces reports for the instrument and individual station performance estimates from the actual measurement network.

The third piece estimates target uncertainty. With inputs for instrument uncertainty and the actual geometric network of measurements USMN computes estimated target field uncertainties.

Observations and Point Group Requirements for USMN

When setting up a network for analysis with USMN there are several organizational requirements for instruments, measurements, point groups, and point names. These are relatively simple in construction as they follow the process commonly used to “Locate an Instrument.”

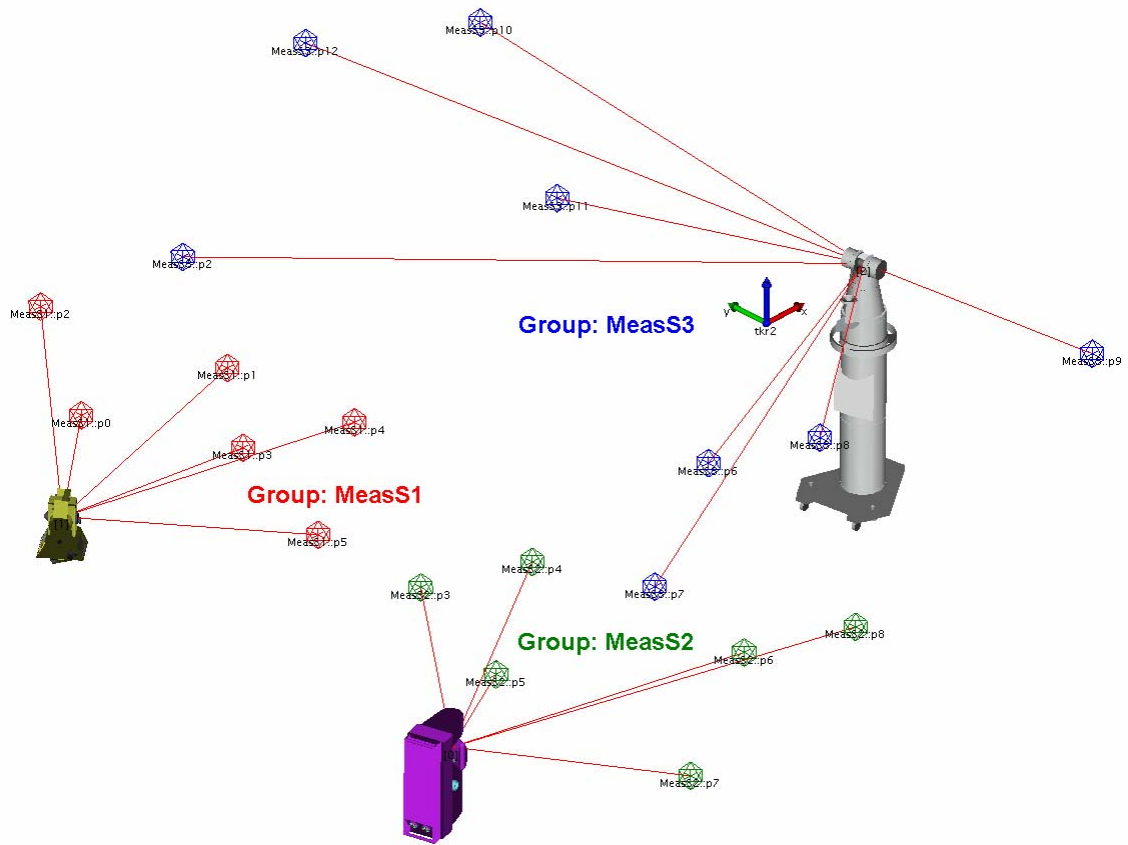
⁷ Taylor and Kuyatt, 1994: NIST TN/1297

⁸ ISO, 1993

The requirements for USMN analysis are as follows:

- Measured point names are matched across the entire job to form the spatial metrology network.
- Individual point names must be common between groups as USMN will associate measurements from different instruments based on point names, e.g., MeasS1::P1, MeasS2::P1, ... MeasSn::P1 will be combined to compute the composite point P1.
- Points without measurements are ignored, unless the point group is included as a 'Nominal Group.'
- A specific point should only have one instrument measuring it.
- An instrument can make more than one measurement to a specific point. As an example an instrument can have a front and back sight to a point or an instrument might have n front sight measurements to a point.
- Measurements that are marked in the Spatial Analyzer database as 'Ignored' are not included in the USMN network. The measurement details dialog for a point provides an interface to determine a measurement's status.

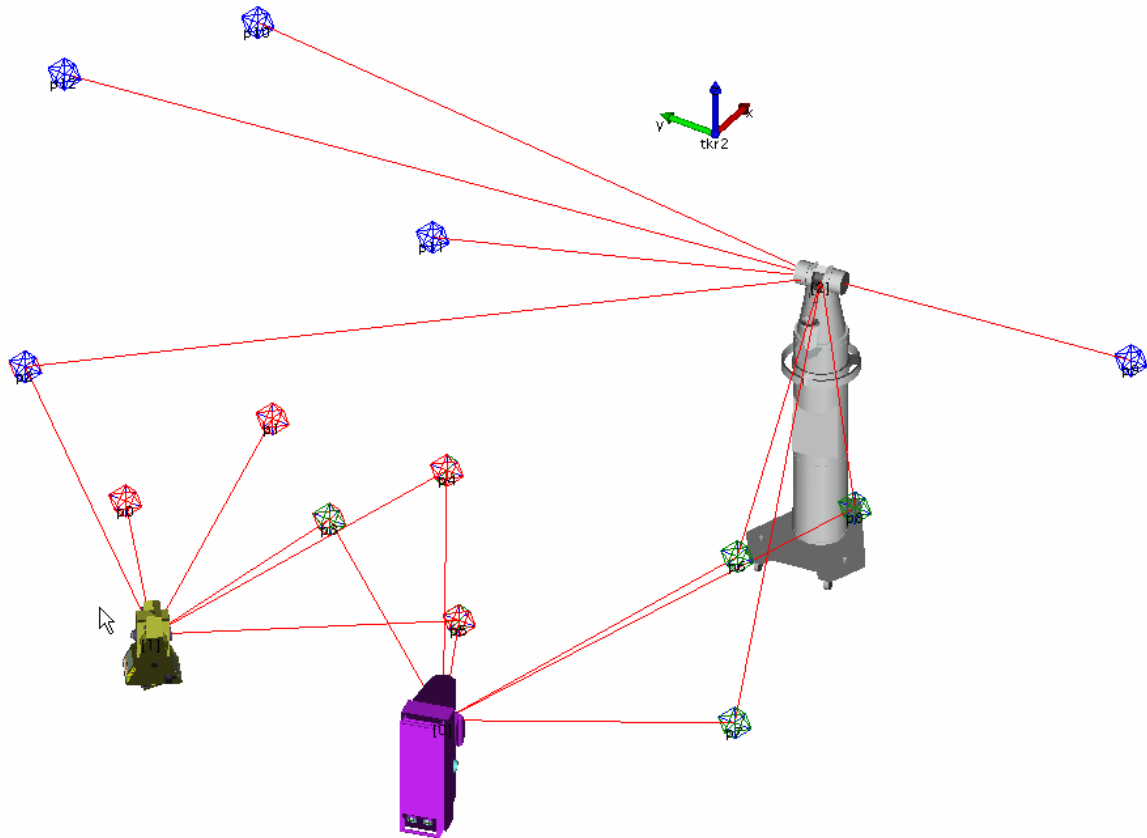
Please note a group name does not need to include a specific reference to the instrument that made measurements to the points contained within it. The designation of S1, S2 ... Sn used in the example was to illustrate a convenient means of organizing the data for USMN analysis.



Measurement and Point Group Requirements for USMN

The figure above shows an example of the measurements and point group requirements for USMN. The Total Station Theodolite instrument has made measurements into a point group called “MeasS1.” This instrument has made measurements to points p0, p1, ... p5 in the “MeasS1” group. The SMX tracker measurements are to points p3, p4, ... p8 in a group called “MeasS2.” So between the groups “MeasS1” and “MeasS2” there are three points that have the same name p3, p4, and p5. When USMN is started it scans the point groups and looks for common point names and then establishes the metrology network based on points that have common names.

The Leica Tracker instrument has six measurements to points in the group called “MeasS3.” Points in this group include p6, p7, ... p12 and p2. Common point names exist between the points in group “MeasS3” and groups “MeasS2” and “MeasS1.” The common names between the “MeasS3” group and the “MeasS2” group include p6, p7, and p8. Point p2 appears in groups “MeasS1” and “MeasS3.”



Instruments, Measurements and Point Network

The figure shows the network of instruments and points along with the measurements which networks them together.

Typical Process Flow with USMN

Typical measurement and analysis process flow using USMN is outlined below:

1. Setup Instrument and measure targets
 - a. Common targets from different stations need to be measured into different point groups
 - b. USMN matches point names between stations so avoid naming non-common targets with names that might be matched
2. Use Instrument >> Locate >> Best Fit to roughly position the new instrument station
 - a. Nominals group = prior common target measurements
 - b. Meas group = current stations measurements of common targets
3. Repeat Step 1 - 2 for each instrument station position
4. Solve network with USMN
 - a. Configure at least one station as fixed (unchecked)
 - b. Exclude any targets that are not to be considered in the network solution
 - c. Select Best-Fit and Solve
 - d. Eliminate outlier observations above 200%
 - e. Solve

- f. Verify residuals are within acceptable bounds
 - g. Repeat steps d and e until step f is satisfied
 - h. If Uncertainty Fields are required for targets, configure the Uncertainty Field Analysis options and select Begin
 - i. Select the option to de-activate observations weighted too 0.
 - j. Produce the USMN composite group
5. Transform with Analysis >> Best Fit >> Points to Points.
 - a. Nominals Group == Reference monuments
 - b. Meas Group == USMN Composite Group
 - c. Solve X-form
 - d. Verify residuals are within acceptable bounds
 - e. Apply X-Form:
 - i. USMN Composite Group
 - ii. Instruments
6. Report USMN Composite Point Group coordinates and uncertainties

The outline above is a typical process for measuring an object with multiple stations.

The USMN component is also used in instrument calibration processes where the Instrument Uncertainty Analysis tools on the USMN dialog are important. Chart and tables of data broken down by instrument type, station and measured component are available. Exports to Excel allow additional analysis.

USMN with Point Groups

USMN is able to use a control point set as an instrument when solving the network. When the control point set has uncertainty estimates USMN uses them to appropriately weight their relative contribution. The control point set can be used to tie instrument stations together from within USMN. A typical application for this method is when stations can not see a satisfactory set of common pts. It does not happen frequently however when it does this is an excellent technique.

Solve the network and include data from a measurement system (e.g., V-stars data imported from an outstars.txt file) that does not have observations but has points plus uncertainties. This is a method to integrate other 3D metrology into USMN using there estimates for uncertainty.

There are several issues to be aware of when using Point Groups to constraint the network. They include:

1. The control point group can distort station locations to match the Point Group. This effect is not always desired.
2. If the control point group is allowed to move (i.e., it is checked) the network will drift off these nominals.
3. Uncertainty estimates for point groups from other systems need to be verified to be consistent with the SA instrument uncertainty estimates. Inputs for uncertainty in SA are assumed to be one-sigma values.

The control point group does not have to remain fixed. Its relative position can be included in the solution. While its relative position can be solved in USMN for the entire point group is treated as a rigid body. The point group remains consistent relative to itself during the solution.

If the control point group is allowed to move during the solution use a points to points best fit (or another transformation technique, e.g., relationship fitting) to orient the network (points and stations) in object space.

USMN User Interface

The USMN user interface is a basic windows interface for use by a measurement technician. Its primary purpose is to help the user optimize the network confidently and get the best Composite Point Group possible from the instrument stations and measurements. All the instrument stations and points are listed with a mapping indicating which stations measured each point. The combination helps the user get a perspective on the networks of instrument stations relative to their target measurements.

The point list is automatically organized in the interface. Points with the highest ranking are listed first. A high ranking indicates a point with a higher chance of problems. Points with lower ranking suggest the target solution is better.

An auto-solve function automates the solution and outlier trimming process. USMN runs with default options without displaying the main interface dialog. The results are then applied to the instruments in the metrology application automatically.

Initiating the USMN

When the user selects the menu option to begin the USMN they are prompted to select the instruments that are to be included in the network. The user makes this selection by clicking on the graphical representation of the instruments, or by selecting them from a list. Optionally, the user may also select a series of nominal point groups that are to be used in the network analysis.

After making these selections, the USMN interface is shown. It is important to note that the only requirement in the interface is that the user press one of the “solve” buttons, review the results, then exit the dialog creating the Composite Point group. The options available in the interface are for the more advanced users; they are not required for basic use.

Overview USMN User Interface

The USMN main interface is shown in the figure below. There are two data lists given. On the left is the list of instruments and nominal point groups that are included in the network. On the right is a list of all the measured points in the network with their errors and uncertainties.

There are several operational elements in the USMN dialog. The first are the Solve options. There is more than one option because the network may or may not be partially oriented. The primary solution functions are described below.

USMN: Unified Spatial Metrology Network

Weight	Instrument (check if moving)	W...	P...	M...	Max Err	Ranking	Ux	Uy	Uz	Umag
<input type="checkbox"/> 1.0000	0: SA B::0 - SMK Tracker	1.0000	p4	0 1	0.01038	117%				
<input checked="" type="checkbox"/> 1.0000	1: SA B::1 - Leica TPS Total Station (2003.5	1.0000	p7	0 2	0.00379	101%				
<input checked="" type="checkbox"/> 1.0000	2: SA B::2 - Leica Tracker	1.0000	p8	0 2	0.00122	59%				
		1.0000	p3	0 1	0.01212	38%				
		1.0000	p6	0 2	0.00085	36%				
		1.0000	p5	0 1	0.02133	22%				
		1.0000	p2	1 2	0.01247	22%				
		1.0000	p0	1 1						
		1.0000	p1	1 1						
		1.0000	p9	2 2						
		1.0000	p10	2 2						
		1.0000	p11	2 2						
		1.0000	p12	2 2						

Auto Solve, Trim Outliers, and Re-Solve
 ☐ Do this automatically

Uncertainty Field Analysis
 Samples: 300
☒ Time Limit: 4.0 min.

Reporting
☐ Append to Report

Apply Results
☒ Create composite group: USMN Composite
☒ Create point uncertainty fields
☒ Apply instrument and point group transforms in SA
☐ De-Activate measurements weighted to zero

Summary
Point Error: Overall RMS = 0.00461, Average = 0.00169, Max = 0.02133 'p5'

USMN Main User Interface

The Uncertainty Field Analysis controls are used to compute point uncertainty estimates. The estimates are based on individual measurement component uncertainties for each station and the geometric configuration of the network.

Solve

The Solve button on the USMN dialog does what it suggests. It starts the network optimization process from the current configuration. The goal of USMN is to properly combine measurements from multiple stations to produce the best measurement network possible. The method simultaneously combines all of the measurements to produce an optimal composite point group.

Any 3D measurement instrument type and number of stations can be included in the solution. When the measurements are not in SA include the points and their uncertainties estimates as a Point Group.

When solving the network the optimization process moves instrument stations to minimize the residuals in the network's point solutions. The movement is computed to optimize each station's measurements within the network. The solution continues until the algorithm can not find any corrections for each individual station position.

Each measurement component is weighted in the individual point solutions. The weighting mechanism is described in a section below.

After solving the network a summary of the network statistics are presented in the summary area of the dialog. Individual point statistics are updated in the points list. Outlier trimming is also possible after the solution.

Best-Fit and Solve

Best-Fit and Solve combines an Instrument Locate function with the Solve operation. The function Instrument >> Locate >> Best Fit moves individual instruments by solving for the fit between a nominal and measurement point groups. The process performs an Instrument locate for each station in the network then initiates the Solve process. This mechanism is important when the stations locations have not been estimated with other techniques.

Auto Solve

The Auto Solve function combines a series of USMN processes together. It is a macro that does a Best Fit and Solve, Trim Outliers, and then Solve operations sequentially. The trim outlier process is preformed measurements above the 100% ranking threshold after the Best-Fit and Solve step. If no measurements are found to be above the 100% ranking threshold then the process stops. When there are trimmed observations a Solve is initiated automatically. The Solve function starts from the Best-Fit network configuration state.

The automated process is recommended for well understood networks and in applications where the process has to be automated. It is not recommended when first optimizing an unfamiliar network of measurements. Anytime outliers are eliminated from the network the causes and details should be characterized and documented.

Solution Quality Indicators

A number of statistics are provided for USMN networks. Understanding what they mean and how they relate measurement confidence is important. The statistics are derived directly from the measurements and their closure on the Composite Point Group.

Three primary statistics that helps judge the quality of the measured network are the Point Error Overall RMS, Average Error, and Max Error values. Root Mean Square is a statistical measure of the magnitude of a varying quantity. In this case the RMS Error is a good statistic describing the magnitude of the errors between the measurements and points. The Average Error also characterizes the mean error for all the points in the network. The Max Error is important to understand because there maybe some blunders or outliers in the dataset.

There are no hard thresholds for USMN error statistics. The numbers are computed based on network configuration, instruments, and environment. Typically an organization defines specific targets for these statistics to evaluate the success or failure of the network alignment. In many cases these thresholds are determined from the tolerance requirements from the measurement systems or object being observed.

There are two statistics that relate to individual point confidence. The Ranking and Max Error statistics in the Point List provide feedback as too the measurements closure on each point. They are each described below.

Ranking and Max Error

The column labeled Ranking provides an important method to characterize a point's relative solution confidence. A high ranking is bad ... the lower the ranking the closer the individual measurements meet. Generally getting all of the ranking statistics under 200% is considered a typical goal.

Ranking provides an efficient means of determining which points and measurement components are possible outliers. The number in this column shows the percentage of how much of the expected 3-sigma envelope the measurement residuals are consuming. A point with a ranking of 100% is using its entire 3-sigma envelope. A point with a ranking greater than 100% is an indication that it has one or more measurement residuals that are not fitting within 3-sigma of the optimized point. Points with rankings under 100% have all of their individual measurement residuals fitting tighter to the optimized point than the 3-sigma confidence interval.

The ranking percentage value for the point is based on the maximum measurement residual. As an example lets consider a point that has four measurements on it. Three of the four measurement residuals fall within 60% of the 3-sigma envelope and the fourth is outside of the envelope. In this case the point's ranking is based on the relative percentage of the fourth measurement.

Measurement component contributions are computed and shown in the point properties dialog. Double clicking on a point in the list will open up the point properties dialog. Notice that point's ranking is based on the MAX error of the measurements to the point. One of the measurements will have the value shown on the main USMN dialog... it is the most probable culprit.

The next four columns show the uncertainty of the point in terms of the current reference coordinate system. These columns are initially blank until the user performs the uncertainty analysis.

When you hover the mouse over any of the measurements, a window will pop-up showing the point name, the computed coordinates for the current working frame, and the point's weight in the solution. The figure below shows the pop-up circled.

W...	P.	Meas	Max Err	R...	Ux	Uy	Uz	Umag
1.0000	r7	0 1 2 3	0.00197	104%	0.00099	0.00101	0.00045	0.00149
1.0000	P1	0 1 2 3	0.00297	101%	0.00096	0.00090	0.00037	0.00130
1.0000	r7	X=11.3264, Y=11.8594, Z=3.8278 Weight=1.0000			0.00053	0.00053	0.00053	0.00126
1.0000	r4	0 1 2 3	0.00153	81%	0.00089	0.00087	0.00053	0.00135

Point Coordinates Pop-up Window

What is the recommended stat to report the quality of our network? For example, there is RMS; there is an average uncertainty, a maximum uncertainty etc. This is important as it is our success indicator and a baseline for design review or derivation of tolerances. For example, the GUM would use an expanded uncertainty.

USMN Composite Point Group

The USMN Composite Point Group is created at the conclusion of the USMN process. This composite group is a point group without observations. If the user chooses an Uncertainty Field Analysis can be performed i.e., compute the Uncertainty Fields. If the uncertainties analysis is performed the uncertainty clouds are associated with the composite group points. A point with a single observation (assuming laser tracker) has the same coordinates as its measured brother.

After optimizing the network with USMN and creating the USMN Composite Point Group it typically needs to be transformed in to object coordinates. The typical method to accomplish this next step is to use the Analysis >> Best Fit >> Pts to Pts function. The reference group is selected and then the USMN Composite Point Group is selected as the measurement/corresponding point group. After solving the transform (and potentially the scaling component) move all the instruments with the composite point group together and therefore keep the optimal USMN network.

The Composite Points will have different components... if there is more than one observation on it. When more than one station has shot the point USMN computes the optimum points.

Exclude Measurements

The Exclude Measurement function is used when measurements need to be excluded from the solution for naming issues or some other procedure reason. Individual measurement and points can be excluded from the solution with this function.

Trimming Outliers

The interface provides the ability to detect measurement blunders or Outliers. This is accomplished by a listing of a solution statistic called the Ranking. Ranking characterizes point solution errors. Ranking provides an efficient means of determining which points and measurement components have possible outliers. This list may also be sorted by clicking on the column heading. The user can sort by the ranking and determine the points in the measurement job that contain measurement blunders.

In typical job getting the rankings under 200% is a general goal. The high ranking points do not always tie in to the instrument with the high uncertainty. This indicates that this method should not be used to isolate which instrument is not performing as expected. These two issues may not correlate. A high uncertainty instrument may shoot a target accurately and visa-versa. In general trim outliers until all the targets are under 200%. Trim targets only until the general solution stats meet your criteria. If a target or a relatively small percentage of observations are outside the expected amount of variation and a cause can be assigned... then and only then trim them.

To understand what causes an instrument to have high uncertainty analysis statistic is made easier however is there a working limit that USMN can provide. In many cases the environment, operators experience and process play a role. A target up close to the instrument may have a large angular deviation but only have a small distance residual.

The cases of outlier vary. In some cases the targets can be miss-identified when measured from different stations. The stability of the target relative to the instrument is

another source. This can happen when a tracker is kicked between one shot and the next. There are an unlimited number of possible issues that cause outliers. USMN does not differentiate between them it only helps to identify them.

There is a long list of “if” conditions that can cause outliers. Some items to consider include: instrument cal ... reflector cal ... on and on... the Instrument Uncertainty Analysis provides a good look at why particular stations (i.e., instruments) have problems. Please review those details in your search for an understanding on what the potential culprits are.

Instrument List

The instrument list in the upper left of the dialog has check-boxes to the left of each instrument. The check boxes indicate if the instrument is free to move (i.e., the software will optimize its location) during the network solution or if it is to remain fixed. In addition, there is a weight field. Entering a value in this field applies that weight to all measurements from this instrument in addition to any other weighting factors the network is applying.

Fixed vs. Un-Fixed Instruments

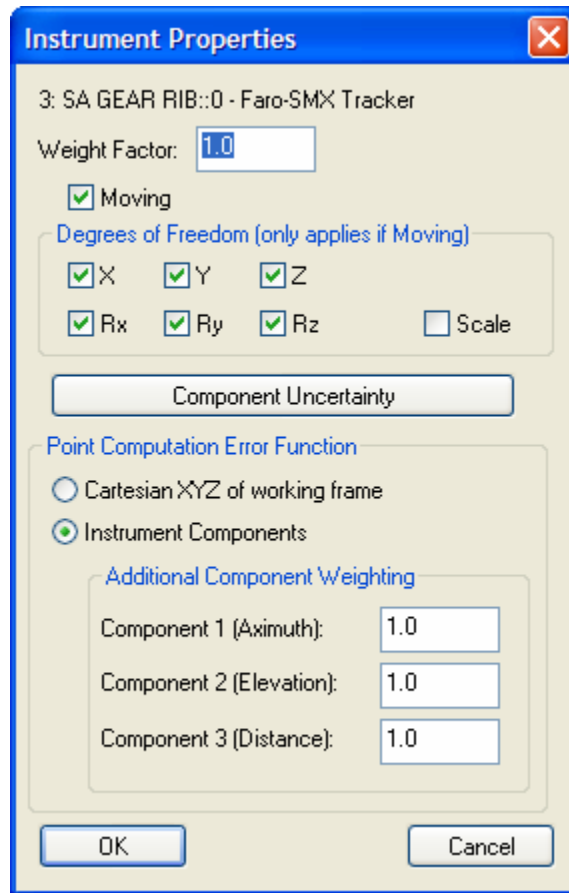
The difference between fixed and floating instruments is controlled by check marks in the instrument list. Fixed instruments are unchecked. They are not allowed to move during the solution or during the Target Uncertainty Field Analysis. Unfixed instruments are allowed to move into a more optimum position during the solution. Generally fix one instrument to solve the network and check them all when performing the optional Target Uncertainty Field Analysis.

USMN will uncheck at least one station when solving. By default it is the first instrument selected. The check boxes allow the user to control which stations are fixed and which ones are solved for.

A commonly used method when solving large networks is to solve smaller grouping of stations separately. After getting smaller sets of stations optimized combine them together. A base grouping of stations is selected to start from. Leave the base stations fixed by un-checking them. The group of stations being added to the base grouping should be checked. Solving the network moves the checked stations to the base group of stations.

Instrument Properties

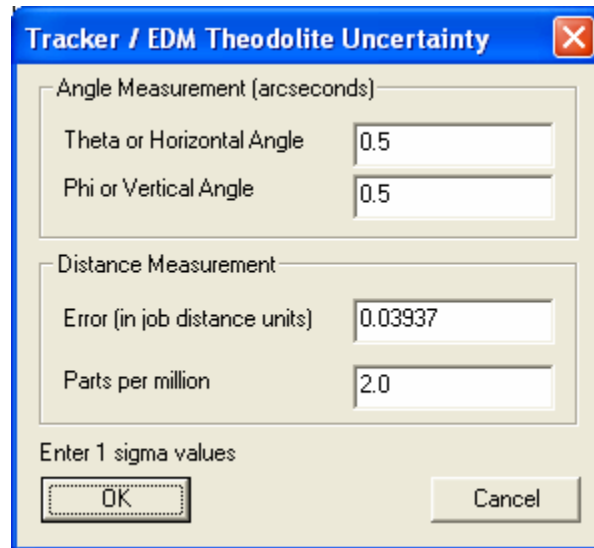
Double-clicking on any of the instruments in the instrument list will bring up the instrument properties dialog.



Instrument Properties Dialog

There is a check-box to indicate whether the instrument is free to move during the solution. There is also a control for scale. If checked, the instrument's scale factor will also be free to move during the solution. The overall weight for the instrument is also accessible in the dialog.

Pressing the "component uncertainty" button brings up the dialog shown below.



The dialog box is titled "Tracker / EDM Theodolite Uncertainty" and features a close button (X) in the top right corner. It is divided into two main sections: "Angle Measurement (arcseconds)" and "Distance Measurement".

Angle Measurement (arcseconds):

- Theta or Horizontal Angle: 0.5
- Phi or Vertical Angle: 0.5

Distance Measurement:

- Error (in job distance units): 0.03937
- Parts per million: 2.0

At the bottom, there is a label "Enter 1 sigma values" and two buttons: "OK" and "Cancel".

Instrument Uncertainty Properties Dialog

This dialog's exact appearance will vary depending on the type of instrument. In this case, the dialog shows the uncertainty settings for a total station or laser tracker. Here, the user may adjust the uncertainty properties for the instrument based on the results of an uncertainty characterization.

It is important to appropriately characterize uncertainty estimates for each instrument in the network. Uncertainty estimates are made for each measured component, e.g., Horizontal Angle, Vertical Angle, Range are each estimated for a laser tracker. The range uncertainty component has two parts. A fixed and range dependent part. The offset component is a fixed quantity. It models the uncertainty due to target offsets, target adapters, etc. The range dependent component is a scalar that is multiplied to each measurement range. For example 2 parts per million (ppm) is used for total station instruments with Electronic Distance Meters to model the environmental variables.

In the instrument properties dialog, shown in the figure, there is also an area for the point computation error function parameters. There is a choice between the working coordinate frame X, Y, Z components and the instrument measurement components. For the case shown in the figure, the instrument uses spherical coordinates corresponding to the instrument's measured components. The selection of the second option, weighting by measurement component uncertainty, is the default since it sets the weights based on the instrument's measurement abilities relative to other instruments in the network. Within this option is also a series of weighting fields where the user can add additional weighting factors to the relative uncertainty weighting.

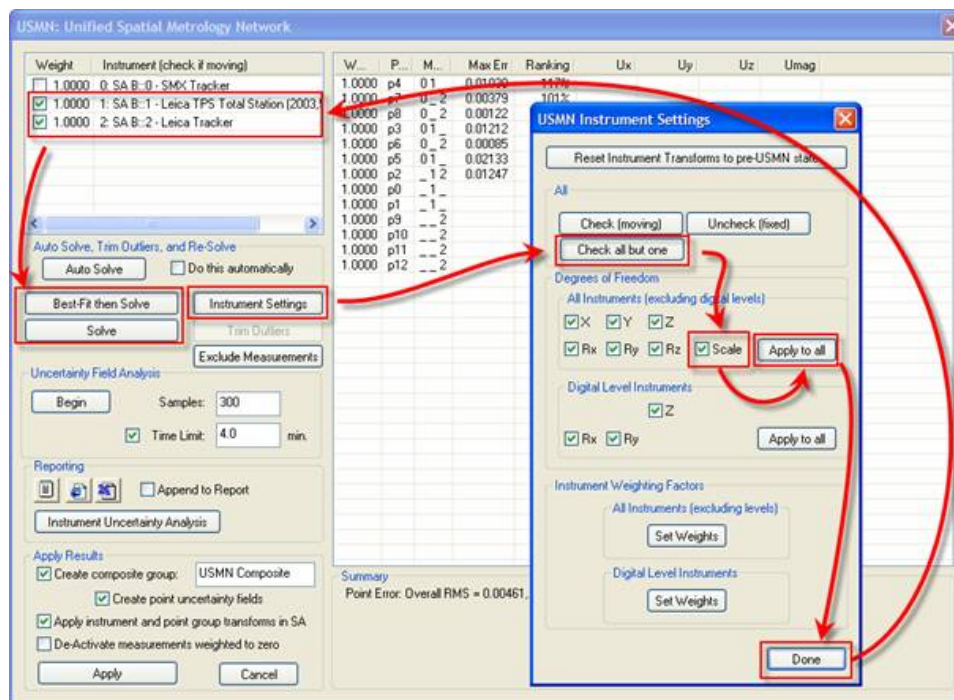
Scaling Instruments in USMN Solution

USMN is able to solve for each instrument's scale factor in its optimization of the network. Scaling an instrument station(s) with a delta temperature and object material process is a common method for establishing a base instrument for the rest of the network to solve too. Another common method is to scale an instrument station(s) with a reference point

group in a best-fit transformation process. The scaled (i.e., temperature compensated) station(s) becomes the base station(s). All other stations can be scaled to match it.

A fixed reference group selected in the USMN with Point Groups function is another method to consider. Specific recommendations on which scaling method is appropriate for an application need to be carefully considered. There are many potential scaling solutions and selecting the one that meet the application requirements needs to be made and then consistently followed by each team of users, engineers, and analysts.

USMN can then scale all of the other stations to match the base station(s). Start USMN in the typical mode. Select the stations that you want USMN solve their scale factor and set the optional scale factor check box for them. The figure below shows the typical process.



Then click the Solve or Best-Fit and Solve button. After accepting the solution each station with its scale factor included will have its scale properties set. Care should be taken to review each station's scale factor property after using this technique. An instrument scale factor is a fundamental property that has significant ramifications to its measured points and within the network.

Point List

Each point in the network is listed in a point list. The weight column allows the user to set the weight of a particular point relative to other points in the network. As with the instrument overall weight factors, these weights are applied to the point after all other weights are applied.

Next to the point name column is another column showing which instruments (from the list on the left) have a measurement to the point. The second point in the list named p4

The next column contains the maximum error for this point. This is the maximum discrepancy between any of the measurements at the point and the optimal point value. If this number is large, then it is worth double-clicking on the point to determine which measurement is causing the error.

Point Properties

Double-clicking on any of the points in the point list brings up the point properties dialog, shown in the figure below.

Point Properties

Point Name:

r7

Overall Weight

1.0

X = 11.3264, Y = 11.8594, Z = 3.8278

Measurements:

Error Statistics: RMS = 0.000890, Average = 0.000701, Max = 0.001970

Weight	Instrument	Err 1	Err 2	Err 3	Mag	Ranking
1.00000	0: SA0 - Leica TPS Total Station (2003,5000,5005)	0.00006	0.00013	0.00197	0.00198	49%
1.00000	1: SA1 - Sokkia Net-2	-0.00065	0.00084	0.00097	0.00144	58%
1.00000	2: SA2 - Leica TC2000, TC2002	0.00014	-0.00045	-0.00149	0.00156	104%
1.00000	3: SA3 - Sokkia Net-2	0.00083	0.00057	0.00031	0.00105	59%

OK

Copy Selections to Clipboard

Cancel

Point Properties Dialog

This interface shows all the instruments with measurements to this particular point. Each individual measurement has a weight value that may be adjusted by the user. In addition, all the measurement's error components are reported individually along with the magnitude. The residuals are shown in length units.

Depending on the settings in the instrument properties dialog (explained later in this section), the error components will be either X, Y, Z, error in the working coordinate frame, or instrument error components such as horizontal angle, vertical angle, and distance.

The last column is the measurement's ranking. A ranking of 100% means the measurement residuals are using the entire 3-sigma uncertainty envelope. The ranking number is computed based on all of the weighted error components. For each component the weighted error is ranked from a ratio of the largest error-component.

If the user hovers the mouse over any of the measurements, a window will pop-up showing the weighting for each component of the error and the relative uncertainty of the components. The figure below shows the pop-up circled.

Weight	Instrument	Err 1	Err 2	Err 3	Mag	R...
1.00000	0: SA0 - Leica TPS Total Station (2003,5000,5005)	0.00006	0.00013	0.00197	0.00198	49%
1.00000	1: SA1 - Sokkia Net-2	-0.00065	0.00084	0.00097	0.00144	58%
1.00000	2: SA2 - Leica TC2000, TC2002	0.00014	-0.00045	-0.00149	0.00156	104%
1.00000	3: SA3 - Sokkia Net-2	0.00003	0.00057	0.00031	0.00105	59%
H, V, D Weights: (6.2338, 6.2338, 0.3874) Uncertainty: (0.00006, 0.00006, 0.00103)						

Observation Weights and Relative Uncertainty Components Pop-up

Uncertainty Field Analysis

This area of the USMN interface allows the user to initiate the repetitive solution of the network. There is an input field for the desired number of samples as well as a solution time limit. Once these are set properly, the user can press the “Begin” button to initiate the solution. The display will periodically update the uncertainty during the solution process. This process can be aborted at any time.

The screenshot shows the USMN: Unified Spatial Metrology Network interface. The left sidebar contains several sections: 'Weight' and 'Instrument (check if moving)' with a list of instruments; 'Auto Solve, Trim Outliers, and Re-Solve' with buttons for 'Auto Solve', 'Best-Fit then Solve', 'Solve', and 'Instrument Settings'; 'Uncertainty Field Analysis' with a 'Begin' button, 'Samples' set to 300, and 'Time Limit' set to 4.0 min.; 'Reporting' with 'Append to Report' checkbox and 'Instrument Uncertainty Analysis' button; and 'Apply Results' with checkboxes for 'Create composite group: USMN Composite', 'Create point uncertainty fields', and 'Apply instrument and point group transforms in SA', along with a 'De-Activate measurements weighted to zero' checkbox and 'Apply'/'Cancel' buttons.

The main window displays a table with columns: W..., P..., M..., Max Err, Ranking, Ux, Uy, Uz, and Umag. The table contains 12 rows of data for various points (p1 to p12) and their associated weights, maximum errors, and uncertainties.

Below the table, a 'Summary' section provides the following information:

- Point Error: Overall RMS = 0.00461, Average = 0.00169, Max = 0.02133 'p5'
- Uncertainty Magnitude: Average = 0.00944, Max = 0.03631 'p0'
- 68.26% Confidence Interval (1.0 sigma), Samples: 300, WCF: A:\WORLD
- Uncertainty Analysis Time: 7.3 sec

Uncertainty Field Analysis

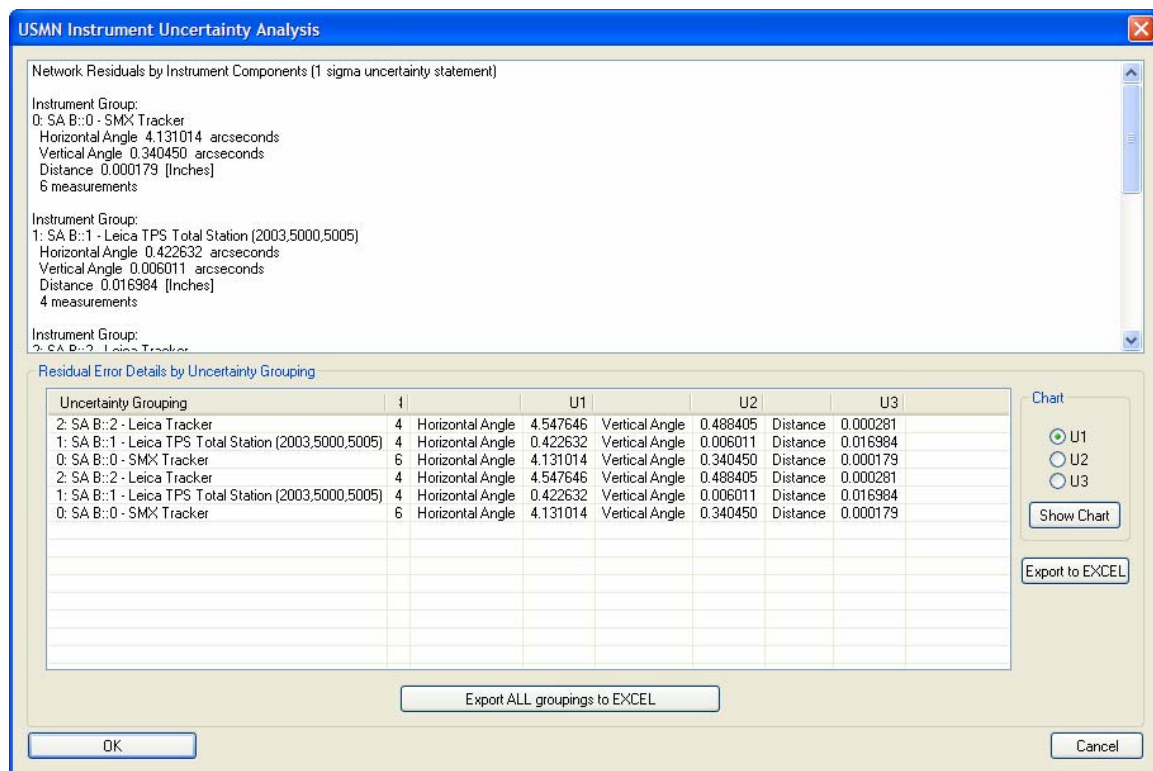
The analysis shown in the figure used 300 iterations to complete. It also ran with all the instruments checked. This configuration is known as a Free-Net solution. After each uncertainty field analysis iteration the uncertainty cloud points are transformed back the Composite Point Group. This mechanism effectively provides an uncertainty field that is not constrained to any specific station. Changing the system's properties by allowing different moving versus fixed instruments and or point groups produces different uncertainty results. One of the primary capabilities of the USMN is the flexibility it provides in this regard.

Reporting

This area of the USMN interface allows users to easily create a text, HTML, or Excel output file containing the results displayed in the points list.

Instrument Uncertainty Analysis

This report is based on the methods for instrument uncertainty characterization presented in Section 0. The figure below shows the Instrument Uncertainty Analysis Results dialog. Each instrument type is analyzed and reported as a group in the first section of the report. Following the instrument group report each individual instrument station is analyzed. The results indicate the performance of the device against it expected values.



Instruments Uncertainty Analysis Results

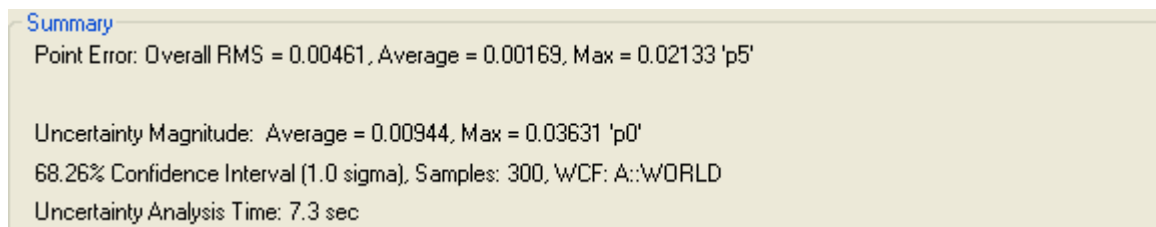
The results from this analysis provide a real process check of the instrument's performance in the network. Specific instrument measurement component performance can also be assessed against its specifications.

Changing Instrument Uncertainty Estimates

Instrument Uncertainty Estimates are generally not changed for the Uncertainty Field Analysis in USMN. There are generally two criteria that should be considered before changing the instrument uncertainty estimates. First the instrument uncertainty analysis shows significantly higher or lower values for the measured component variation. Secondly there are a sufficient number of observations. If only one actual instrument is used to make all the observations ... use the same estimate for each station it was used on. Only one estimate should be used all stations for a specific instrument.

Summary Statistics

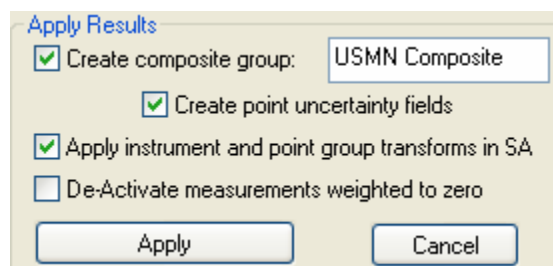
This area contains summary information about the network. This includes statistics on the point computation errors including the maximum error. Statistics are also presented for the uncertainty fields. The figure shows an example of the summary provided. Note that the confidence interval is reported here. This value is set in the Analysis tab on the User Options dialog in Spatial Analyzer.



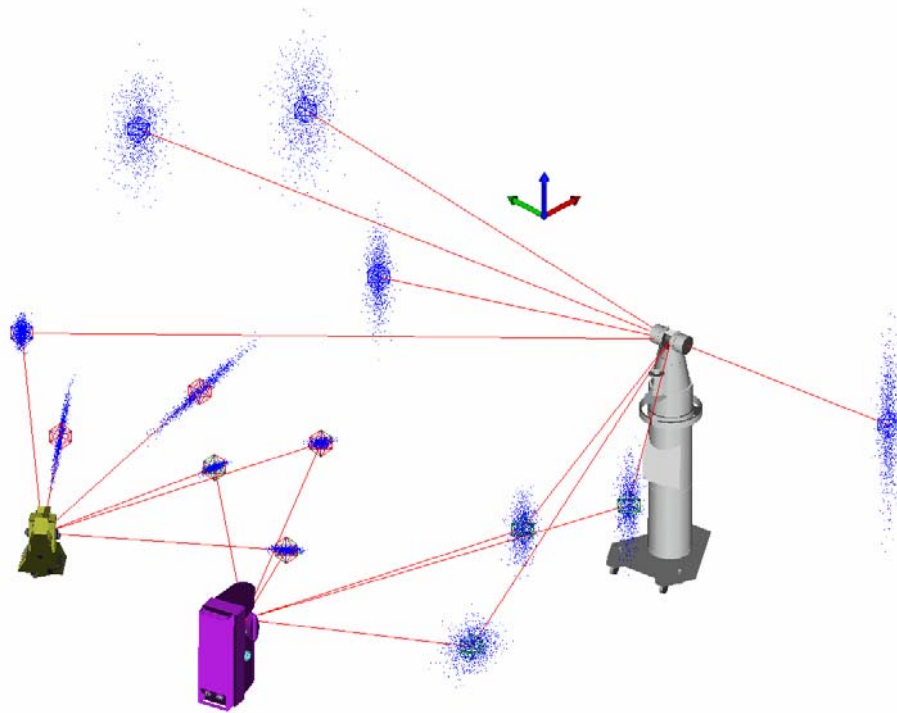
Summary of the Analysis Results

Apply Results

This area controls how the results of the USMN analysis will be integrated with the metrology application. There is an option to create a composite point group. This is a group that will contain the optimal point for each measurement set. There is also an option to include the uncertainty fields (if they have been computed) with the coordinates. Finally, there is an option to apply the instrument transformations determined in the network solution to the actual instruments in the metrology application.



The figure below shows the results in Spatial Analyzer after computing the Uncertainty Fields and selecting the Apply button.

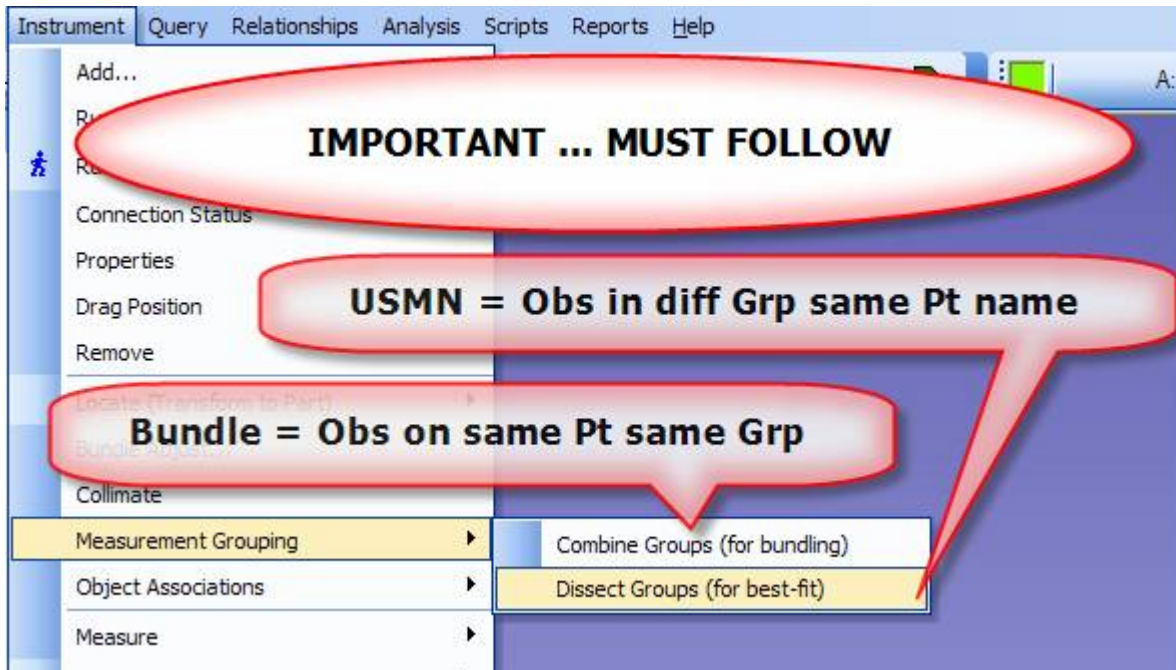


Analysis Results and the created Composite Point Group

Measurement configuration differences between Bundle and USMN

There are important measurement process differences between setups for bundling stations verses using USMN to solve metrology networks. The two types of configurations are outlined below:

1. Bundle = observations from different stations into same point (e.g., same group)
 - a. All groups within active collection are used.
2. USMN = observations from different stations into different points with the same name (e.g., different groups)
 - a. All observations with matching point names are used e.g., two p1's on a sphere will be considered common



There are two functions on the Instrument >> Measurement Grouping menu that help arrange measurements either for bundling or USMN (e.g., Best-Fit). Please see the functions under Instrument >> Measurement Grouping to help manage the observation when needed.

Why USMN?

Accepted standards and best-practices require uncertainty statements for coordinate measurements. This means that the use of measurements from the combination of multiple coordinate acquisition systems requires a statement of the total uncertainty. . It is difficult to take action such as product acceptance, product rework, or product scrapping based on measurements without realistic uncertainty statements.

Often, in the process of inspecting and manufacturing goods, it is necessary to combine measurements from several coordinate acquisition systems in order to build a total representation of the part. Data combination is particularly common with portable coordinate measuring devices including laser trackers, laser scanners, portable CMM arms, total stations, and theodolites. This section will present the motivation behind system combination as well as an overview of the issues it raises.

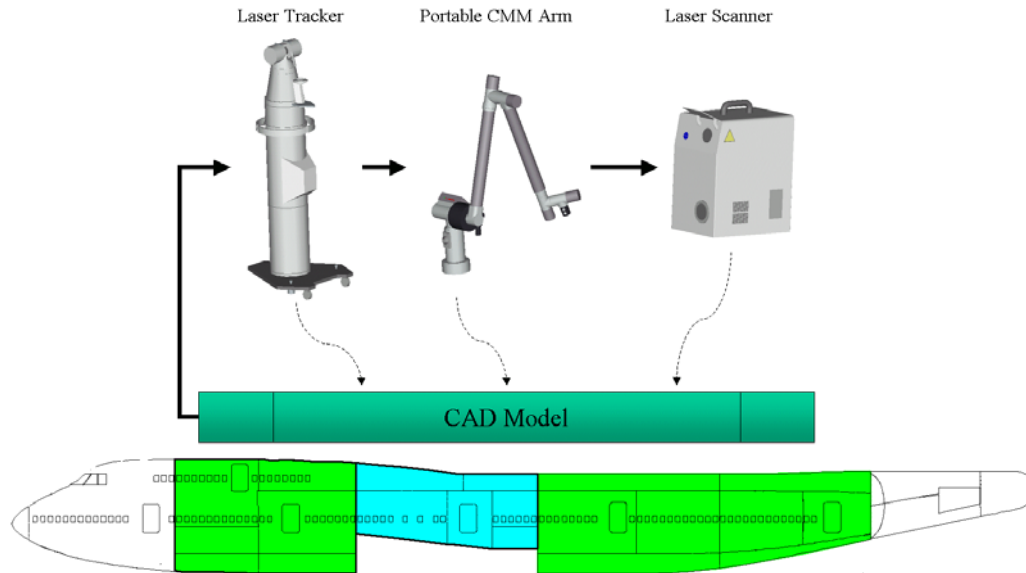
Why is it Necessary to Combine Coordinate Acquisition Systems?

The foremost reason for measurement system combination is physical constraints. Given a large field of targets rich with obstructions, it is not always possible to measure the entire object using one particular measurement system. Multiple types of systems may be required, or a single system may be moved to several locations to cover the entire area. Furthermore, even in the cases where one instrument placement can observe the required regions, it is often necessary to relate these measurements to a design reference such as nominal CAD data. Even in this basic case, uncertainty propagates through a kinematics chain of measurements.

Measurement system combination is required for systems only capable of measuring partial coordinates. Theodolites, for example, must be used in combination to produce complete coordinates. In addition, the combination of full coordinate measurement systems may also be used to increase the potential system accuracy.

Another motivation is the diversity in individual measurement system properties. Systems have varying degrees of accuracy, working volume, measurement speed, setup time, and cost. These limitations and features must be considered when designing a measurement process in order to achieve the highest degree of efficiency.

Consider, for example, a possible measurement scenario for a large aircraft fuselage component. A schematic of this arrangement is shown below. The bold lines show how the systems are coupled and the dashed lines represent the measurements. Given the size of the part, a laser tracker is used to transform into the nominal CAD coordinate system of the airframe and measure points that are visible to the laser tracker. These include points on the outside skin of the airframe. The laser tracker is unable to measure a particular feature around a door, so a portable CMM arm is clamped onto the structure. The portable CMM arm is registered to the airframe through the coordinate system of the laser tracker. This is done by measuring 3 or more common points with both the arm and the tracker then fitting the coordinate systems together. These common points are typically not single point measurements. Instead, each instrument measures many points around a common sphere, determines the best-fit sphere geometry for the measurements, then uses the common sphere center as the common point. The arm is then used to measure various regions of the door. Assume there is an interface part in the door region that has been causing problems during assembly. A dense set of measurement data describing its surface is needed in order to determine the location of the malformation. In this case, the tracker and the CMM are not able to see the part properly, and neither can provide the data density efficiently, so a laser scanner (structured light vision system) with a small workspace is used. The scanner takes a single scan (possibly 300,000 data points) of the part and several tooling balls temporarily located on the structure. The portable CMM measures the tooling balls, the laser scanner extracts the spheres from the raster scan, and the scanner is fit into the portable CMM coordinate system. At this point, we have a complete survey including the skin of the fuselage, points in the door region, and a cloud of points on the troublesome interface part within the structure.



Measurement System Relationship

This data is then compared to the nominal CAD model to determine if the part is assembled properly and, if not, which parts need to be reworked. The question is: What is the uncertainty of the data in the dense raster scan from the laser scanner relative to the nominal CAD geometry for the part? The next section will expand on this issue.

Uncertainty Issues when Combining Measurement Systems

This section presents issues that are raised as a result of measurement system combination.

Uncertainty in the registration to the nominal CAD model: The first operation in a measurement session is usually registration (transformation) or tie-in to the coordinate system of the CAD model. This is done using several methods including the measurement of known points, measurement of a surface contour, or repeatable fixturing. For any of these measurements, the end result is a transformation that is applied to the instrument so that future measurements from the instrument are easily transformed to part coordinates. Any method of registration introduces some degree of uncertainty into the process. This uncertainty must be quantified in order to accurately represent the effects of the registration uncertainty on the final measurements.

Uncertainty fields for the individual coordinate acquisition system measurements: In order to propagate the uncertainty from each coordinate acquisition system to the total measurement set, we must have a geometrical representation of the uncertainty field of each point measurement. This is required because, as the individual systems are tied together, the resulting uncertainty will be dependant on the geometry of the individual uncertainty fields.

Uncertainty field combination for the CAD model and multiple coordinate acquisition systems: Given geometric representations of the uncertainty in the individual

components of a measurement network, an algorithm must be developed to combine these systems while properly representing the uncertainty interactions.

Consider the case where loops are added to the aircraft fuselage measurement chain example presented previously. Suppose another laser tracker was operating inside the fuselage area and was able to see several of the tooling balls measured with the laser scanner. The “inside” laser tracker could measure those points then also measure other points known to other instruments in the measurement system. These could be common points with the “outside” tracker, or known nominal points on the CAD master. In either case, measurements of this sort would close the loop and have an effect on the total uncertainty. Measuring additional CAD nominal points with the portable CMM arm could further complicate this example. The result is a “double-loop” configuration where measurements tie back to the CAD model from 2 sources in the original serial chain. This dissertation will present a method for dealing with these situations and creating realistic uncertainty fields for all the measurements in a network.

Representing Coordinate Uncertainty Fields

The uncertainty in coupled spatial measurement systems is determined through the combination of the uncertainties in the various coordinate acquisition systems that comprise the measurement network. Before dealing with the entire network, we must first address the individual measurement system uncertainty. The individual uncertainties may then be combined with those of other systems. The desire to combine measurements while preserving uncertainty statements imposes several requirements on the uncertainty representation. This chapter will address these requirements, discuss common uncertainty representations, and propose a new method based on a discrete point cloud approach.

Requirements for Uncertainty Field Representation

The uncertainty field representation must be

- An adequate representation of the spatial uncertainty in the measurement system.
- Transformable between coordinate systems without loss of information.
- Combinable to produce a composite uncertainty.
- Intuitive and easily applied to measurement process design.
- Computationally reasonable.
- Applicable to measurement analysis operations

This section will present these requirements in more detail. In addition, commonly used representations will be discussed as they apply to the requirements.

Adequate Representation of the Spatial Uncertainty

If all coordinate measurements devices were capable of producing measurements with uncertainties conforming nicely to X, Y, and Z axis representations, this process would be much simpler. Given the wide array of measurement devices in use today, however, this is not the case. Several different types of sensors are commonly used in

measurement devices, and each type has its own error characteristics. Polar devices such as theodolites and laser trackers provide two angular values and a distance value. Kinematic devices such as portable and fixed CMM arms provide a series of joint values. Laser scanning devices provide a raster grid with range values determined by an offset laser and camera arrangement. There are many others.

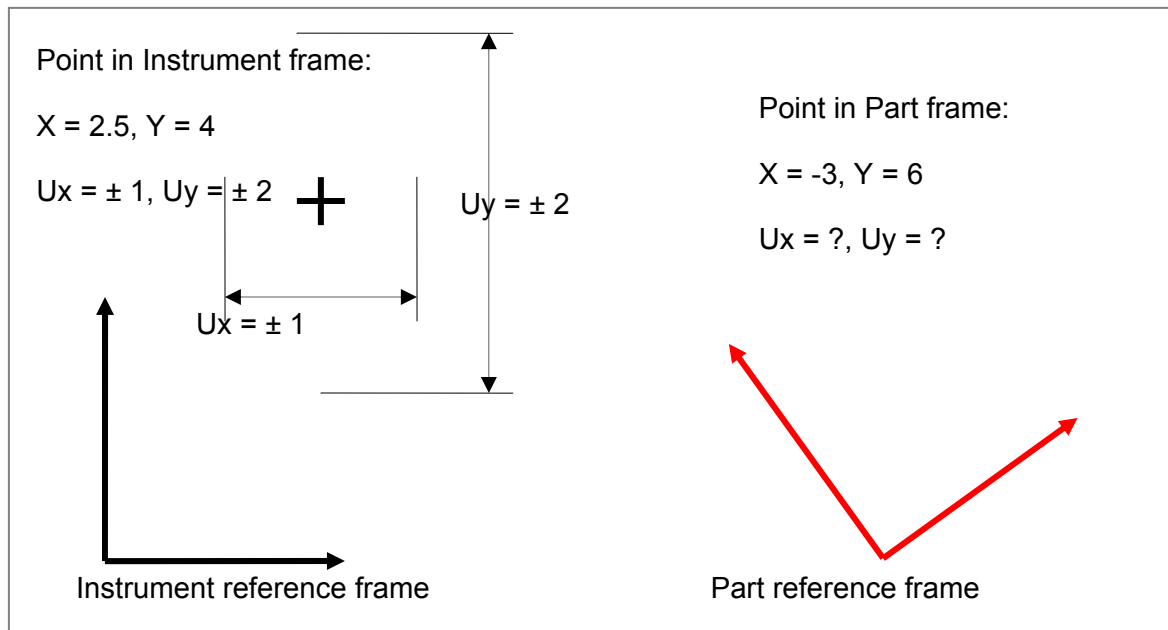
Each system then provides coordinates through a computation which converts the native device values into XYZ coordinates. The resulting coordinates have uncertainties resulting from the uncertainties in the native device values and the model applied in the conversion.

The result is an uncertainty with some shape and density in the coordinate system (usually XYZ Cartesian) of reference. For this reason, simplified representations, such as an uncertainty sphere, are generally insufficient, since they do not accurately represent the geometry and probability distribution of the uncertainty field.

Transformable Between Coordinate Systems

The uncertainty fields must be capable of undergoing a coordinate transformation without loss of information. This is necessary since many coordinate transformation operations are required in a typical measurement session. This requirement becomes even more important when different types of measurement systems are combined and the number of transformation operations performed on the data increases.

Some measurement systems provide numerical XYZ uncertainty values corresponding to their measurements. This numerical representation is only valid in the frame in which it was originally specified. Consider the simple case shown in the figure below. Here, a measured coordinate value is accompanied by an uncertainty statement. Both are stated relative to the instrument reference frame. Suppose, however, the user has arbitrarily placed the instrument relative to the part, so the uncertainty in the instrument frame is not of any use. Instead, the operator is interested in the coordinate value and the uncertainty expressed in the part reference frame. Though the vector directions could be mapped from instrument to part coordinates, the user still cannot determine the uncertainty in part coordinates. This is because the instrument axis uncertainty statements do not contain enough information about the true geometrical shape of the uncertainty.



Coordinate Axis Numerical Uncertainty Statement

Combinable to Produce Composite Uncertainties

When multiple coordinate acquisition systems are used cooperatively in a metrology network, they typically reference each other by measuring common reference points. In addition, when high accuracy is required, multiple measurement systems are often used to measure the same points. This redundancy is used to reduce the uncertainty of the measurements.

In both the reference point measurement task and the redundant measurement task, a point in space is measured with several instruments. Since each instrument has an uncertainty associated with its measurement of the coordinate, there are now multiple independent uncertainty values for the point. Which values should be used both to compute the point location and to represent its uncertainty?

The answers to these questions depend on the process, but in many cases, the answer is that all the measurements and their uncertainties should be used. Given this situation, the uncertainties must be combined in order to provide a proper uncertainty statement for the combined measurement result.

The ellipsoid has often been used as a conceptual representation of coordinate uncertainty. Typically, these ellipsoids are oriented such that the principal axes correspond to the coordinate axes. In some cases, however, they have been used with an orientation that more closely matches the instrument uncertainty profile. In any case, a geometric representation such as the ellipsoid is appealing for the representation of uncertainty. These objects can be mapped between coordinate systems, easily illustrated to the user, and easily derived from instrument performance. This representation is not ideal for combining uncertainties, however. When two ellipsoids

are combined, the resulting shape is most often not an ellipsoid but a more complex shape. This shape cannot be exactly represented with an ellipsoid. It would be possible to determine the bounding ellipsoid for the combined set, but this introduces more approximations into the process. In addition, since the ellipsoid does not contain information about the probability (or density) of the uncertainty field, this important information is lost when the ellipsoid representation is used.

Intuitive and Easily Applied to Measurement Process Design

The end result or product of comprehensive uncertainty analysis should be a graphical and numerical representation of the uncertainty. These results should be easily presented to the measurement system user and the customer of the measurements. This representation must therefore convey the uncertainty in an intuitive manner. Otherwise, the results will be misinterpreted or ignored. For this reason, most representations used in current systems result in a statement to the user of the total coordinate uncertainty as a single number, often a function (such as parts per million) of the distance from the instrument origin. This spherical representation is easy to understand. One of the reasons it is easy to understand, however, is that it ignores the geometrical shape of the true instrument uncertainty. A compromise is needed that is easy to understand, but that accurately describes the complex, spatial descriptions of the true uncertainty field.

In addition, since the uncertainty information may (and should) be used in measurement process design, the uncertainty field should provide information that will help designers minimize uncertainty. If the uncertainty is provided in a useful form, it can easily be integrated into measurement job simulations and overall error analysis. This will aid designers as they determine process capabilities, part fixturing requirements, and a host of other process design factors.

Computationally Reasonable

Computational technology is improving at a rapid rate. For this reason, solutions that are slow, but tolerable, on today's computing platforms will run with negligible burden on the computers of tomorrow. Numerical and iterative methods have, in the past, been avoided for many processes due to computational limitations. These methods are increasing in favor due in part to faster computers, but also due to their flexibility and expandability.

It is important, however, to develop an uncertainty description that is capable of executing on current computer hardware and also able to quickly take advantage of faster platforms. In a practical sense, the uncertainty analysis for a combined measurement job should be performed after all measurements are taken as part of an operator's job reporting process. As manufacturing firms continue to press the application of measurement data to the shop floor and away from the measurement laboratory, the uncertainty analysis must follow this trend. It is not acceptable for the uncertainty analysis to require hours of analysis on a high-power lab workstation.

Applicable to Measurement Analysis Operations

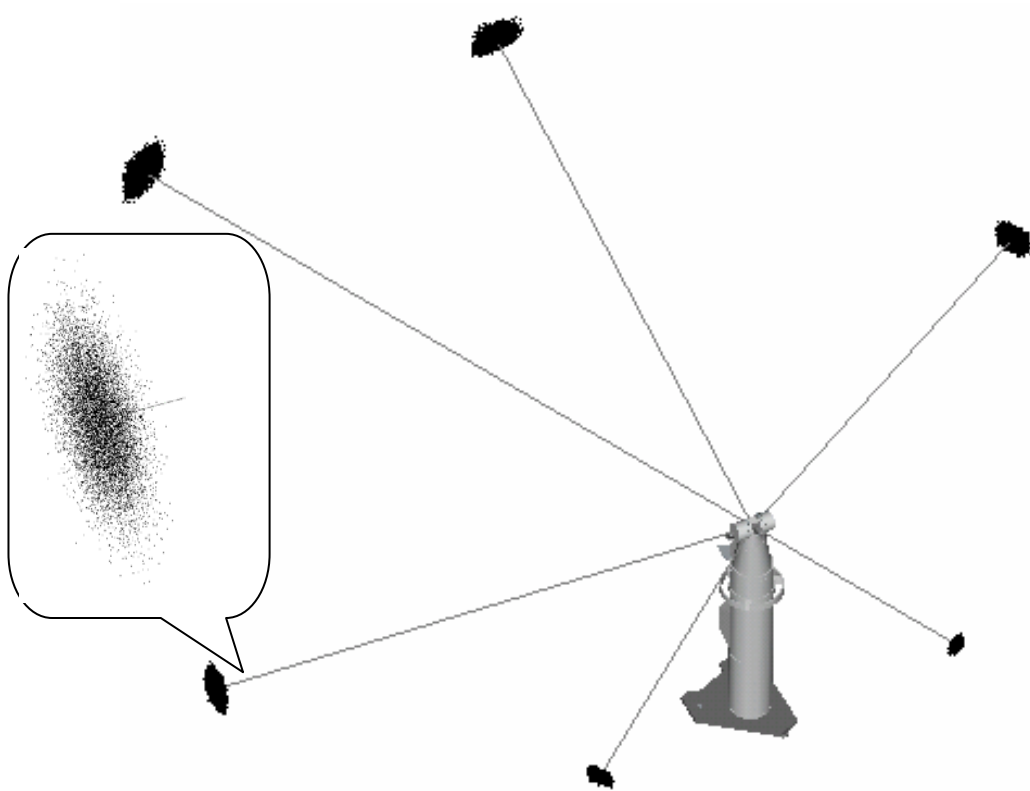
Often, the actual points measured by the instrument are only an intermediate result in the measurement process. This is because those measured points are often used to fit geometric shapes, construct reverse-engineered surfaces, and evaluate geometric

dimensioning and tolerance requirements. The uncertainty field representation must therefore be applicable to these analysis processes. Without this capability, the uncertainty of the actual measurement result is unknown.

Discrete Point Cloud Uncertainty Field Representation

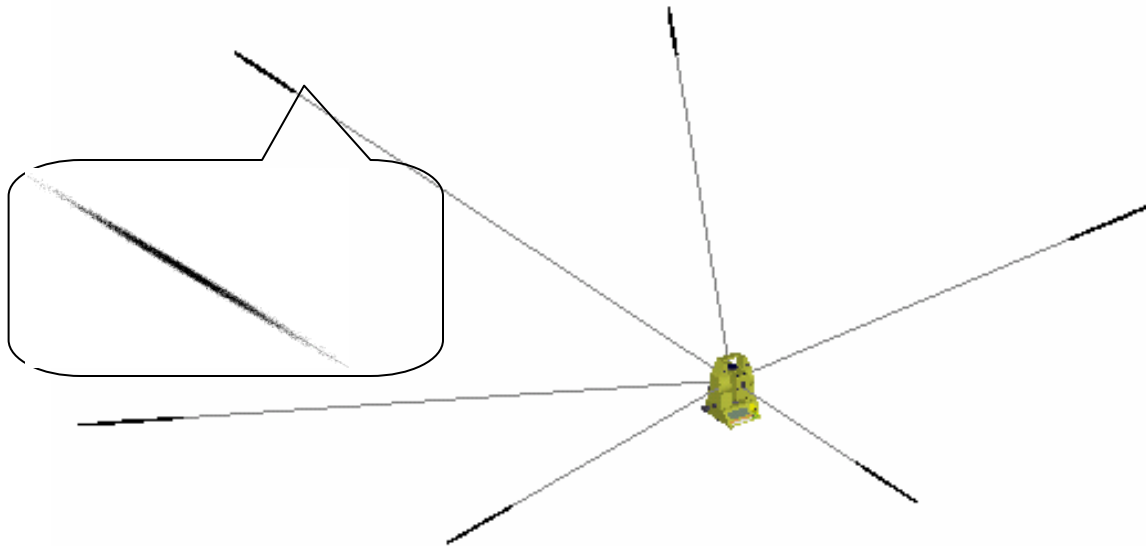
Given the requirements discussed previously, and the currently available uncertainty representations, a new representation, the discrete point cloud method, was developed for the purposes of this research. In this method, the uncertainty field is represented by a series of discrete points that represent the modeled uncertainty due to the measurement device. These discrete point clouds are created by simulating the effects of device errors on a measured coordinate. This simulation is performed by injecting uncertainty into the instrument parameters and determining the effects on the measurements. This process is described in more detail in the next chapter. Here, we will focus on the cloud representation and its advantages over other methods.

The figure shows several uncertainty fields for a common portable measurement device, the laser tracker. The uncertainty fields are scaled for visibility. These clouds have a flattened shape because the distance measurement component is much more accurate than the angular measurement component. Also, notice that as the points move farther away from the instrument, the clouds widen as the angular uncertainty has an increasing effect.



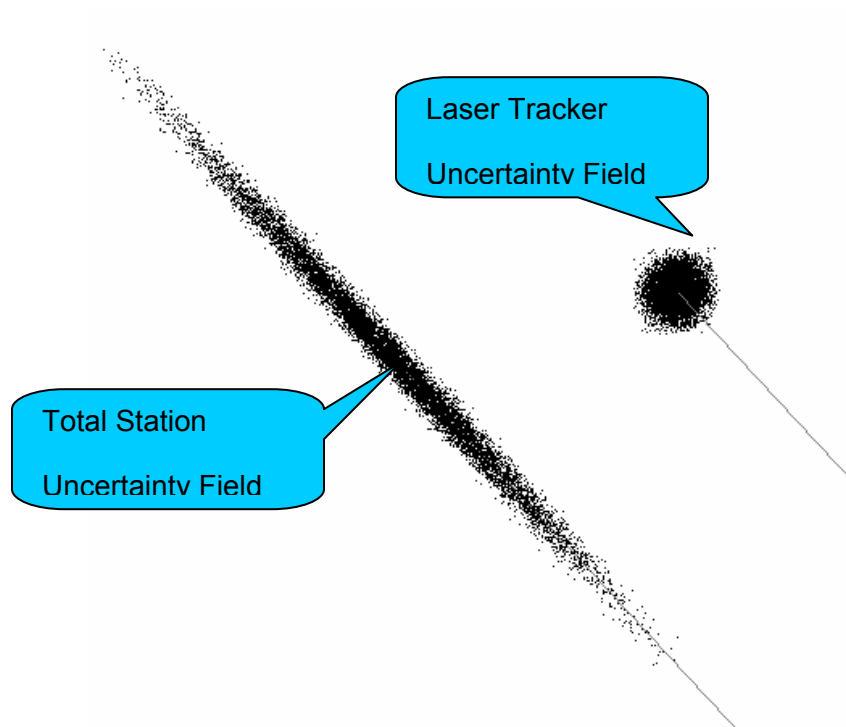
Discrete Point Cloud Uncertainty Fields for a Laser Tracker

Total station theodolites typically have a much different uncertainty profile than that of a laser tracker. This is because the distance measuring component of the total station has much higher uncertainty than the angular components at typical measurement ranges. The figure below shows these fields for the same measurement points used in the laser tracker example.



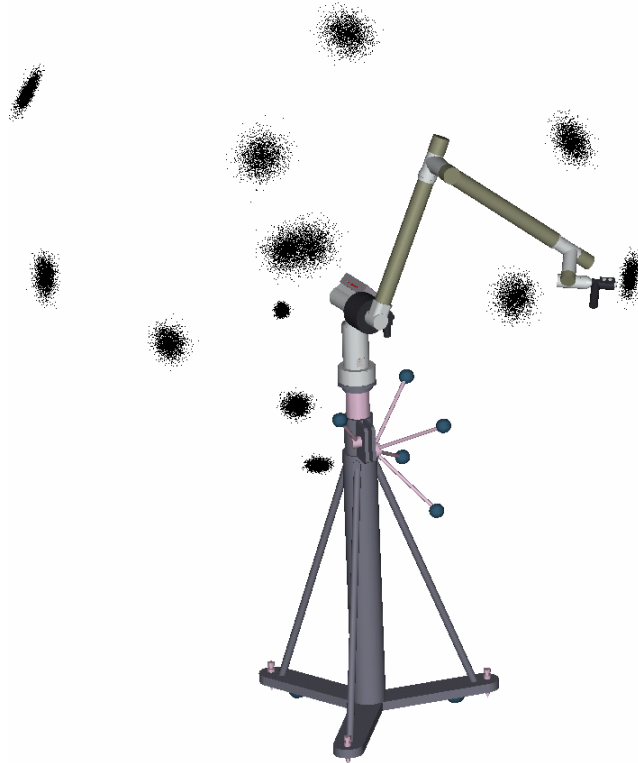
Discrete Point Cloud Uncertainty Fields for a Total Station Theodolite

A comparison of the uncertainty fields is shown in following figure.



Close-up of Corresponding Total Station and Laser Tracker Uncertainty Fields

The uncertainty fields for portable CMM arms result from a more complex kinematics chain than spherical devices such as laser trackers and total station theodolites. Each joint encoder has a degree of uncertainty in its measurement. A series of uncertainty fields for a portable CMM are shown in the figure. Notice the large variability in the size, shape, and orientation of the fields.



Discrete Point Cloud Uncertainty Fields for a Portable CMM Arm

Advantages of the Discrete Point Cloud Uncertainty Field Representation

The point cloud representation meets the requirements set forth in Section 0. First, the discrete uncertainty fields provide an adequate representation of the spatial uncertainty in the measurement system. This is accomplished since the cloud representation has the ability to represent complex uncertainty shapes. Further, this representation also depicts the density of the uncertainty field. This is preferred over depictions that provide only an outer shell bounding the uncertainty.

This uncertainty representation is also transformable between coordinate systems without any loss of information. Such a transformation is performed by multiplying the measured point as well as the uncertainty cloud points by the desired transform. No information is lost, because no assumptions are made in this process.

Combination of discrete uncertainty fields is possible since the two sets of points are easily merged into a composite set. When this is done, all the geometrical properties of each field are preserved. Depending on the measurement scenario, this combination is not simply a concatenation of the point sets. Often, the combination is a result of a best-fit transformation or another operation. In these cases, the best-fit can be performed using each of the corresponding points in the cloud. This provides not only a rigorous combination of the uncertainty fields, but it allows the effect of the uncertain best-fit points to propagate into the points resulting from the best-fit. This process will be discussed in detail in later chapters.

Discrete uncertainty fields are intuitive. They can easily be drawn in graphical measurement software as shown in the preceding figures. When overlaid on a CAD model of the parts being measured, an operator can quickly see how the instrument uncertainties will effect the measurement of the part. For numerical reports, the point clouds are statistically sampled in the coordinate axes chosen for reporting by the user. This provides XYZ uncertainties with the final measurement values. It is important to note that though this numerical representation is not transformable, it is only created in the final reporting phase. The discrete cloud representation should be preserved in case a report of the uncertainty is needed in a different reference frame.

It will be shown in later chapters that the discrete representation is computationally reasonable and therefore capable of providing uncertainty statements on the shop floor immediately after the measurement process. One reason for this is that the discrete representation is computationally scalable. By simply requesting more or fewer points in the cloud representation, the computational burden is easily adjusted to the measurement network complexity and computational platform.

The discrete representation is easily applied to measurement analysis operations. The uncertainty of these analyses may be determined by repeating the measurement analysis for the individual points in the uncertainty fields. The results of the repeated analysis are then statistically processed to determine the uncertainty of the measurement result.

Quantifying Coordinate Acquisition System Uncertainty Fields

Before dealing with coupled uncertainty fields, we must first understand the uncertainty fields of the individual coordinate acquisition systems. Much work has been done in the area of individual measurement instrument uncertainty. Typically, instrument manufacturers state uncertainty as part of the instrument specifications. This uncertainty statement is most often expressed in terms of the individual measurement device technologies present in the device. Given a specification, or experimentally determined values for the instrument uncertainty, the task remains to describe the geometric uncertainty of the individual coordinate values. In addition, methods are needed to characterize the actual shop floor uncertainty of measurement devices under realistic conditions.

This chapter presents an experimental method for characterizing the uncertainty of a coordinate acquisition system's output under any set of conditions. Following this, methods are presented for the conversion of instrument uncertainty values to discrete point cloud uncertainty fields. In addition, the application of these methods to instrument performance evaluation is presented.

Instrument Uncertainty Characterization

Many different methods are used to determine uncertainty specifications for measurement systems. Often, manufacturers seek to provide the best-case uncertainty statements for their instruments. In other words, they provide laboratory results that isolate the instrument from real-world effects such as operator technique and environmental instabilities. This portrayal does not indicate the level of performance an

average operator can expect in a less than pristine environment. As a result, users often have difficulty evaluating or even estimating the performance of their instrument under realistic conditions. Most of the instrument specifications do not quantify uncertainty resulting from factors such as operator technique, environmental properties, and targeting issues. All of these issues must be considered in evaluating the performance of a coordinate acquisition system.

Current evaluation methods typically make use of known artifacts such as scale bars, tetrahedrons, and linear interferometer scales. These artifacts are expensive and difficult to maintain and transport. The method proposed here uses a series of fixed, but unknown targets. The performance of the instrument is evaluated based on the fact that the points remain fixed throughout the measurement process. The degree to which the instrument shows the points to move when they are measured from different instrument locations is directly related to the uncertainty of the system. If the system had zero uncertainty, the relative measurements of the points would be identical for all instrument locations.

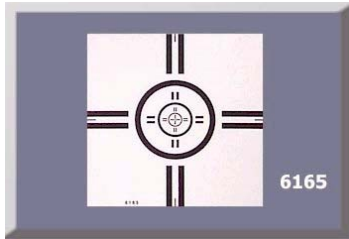
This approach measures the of the instrument's measurements. This measure is then used to make an estimate of the uncertainty. The term in metrology represents the variation in measurement results carried out under different conditions. , on the other hand, represents the measurement variation under identical conditions of measurement. (ANSI, 1997; NCSL, 1995; Taylor and Kuyatt, 1994; ISO, 1993)

The proposed approach is summarized as follows. Establish a field of unknown yet fixed points. Measure these points from several locations of the instrument. Perform an optimization on all the measurements to determine the transformations of the instruments that minimize the residual discrepancies in the redundant point measurements. Statistically process the residual errors to determine uncertainty values for the individual instrument components. Though the laser tracker will be used as the primary example in the following description, this method is applicable to any coordinate measurement device.

This approach is a "Type A" uncertainty evaluation. This is because it is based on a statistical analysis of a series of observations (ANSI, 1997; NCSL, 1995; Taylor and Kuyatt, 1994; ISO, 1993). In addition, it is assumed that all possible corrections have been applied for systematic effects. This is the case for modern metrology instruments since they undergo periodic compensation and calibration in order to refine their kinematics model parameters.

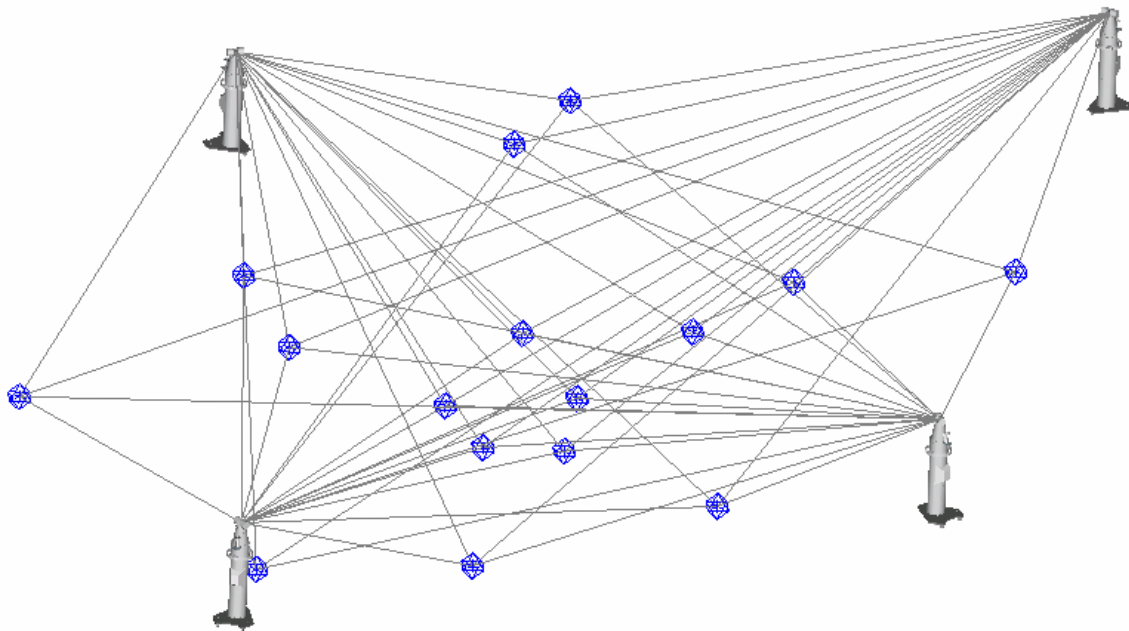
Measurement Process

First, a fixed target field is established. Depending on the instrumentation, this may involve fastening targets to a concrete floor or another rigid structure. There are many options for measurement system targeting. Some of these are shown in the figure below. The coordinate values (locations) of these points do not need to be known. It is recommended, however, that they be distributed throughout the measurement volume of interest.



Measurement System Targeting Methods

During the target field layout process, consideration is given to the desired instrument locations. Since the instrument will be moved throughout the workspace, the majority of the target field should be visible from all instrument locations. An example layout is shown in the figure below. where four instrument positions are distributed around a field of 16 fixed targets. For each of the four positions shown, there are actually two instrument positions at varying heights and orientations.



Sample Target Field Layout

Once the layout process is complete, a series of measurements are performed. Care should be taken during this process to use the instrument in the same manner as it will be used on the shop-floor. In addition, this measurement process should take place in a realistic environment, not a temperature controlled laboratory.

The instrument is placed in the first position, and all the targets are measured. For each measurement, the system will typically provide an XYZ coordinate value. This is not sufficient, however, unless dealing with a prismatic conventional coordinate measuring machine. The information required for each point is a unique point name and the fully compensated internal measurement data set. For a laser tracker, this consists of three values: horizontal angle, vertical angle, and range measurement. In the case of a portable CMM, the data consists of the joint values of the kinematics chain.

The instrument is then moved to the next location and the process is repeated. The time over which all the measurements occur should represent a typical measurement job. This will incorporate the drift in instrument calibration and other factors into the instrument uncertainty statement. In addition, if additional factors are being studied, introduce those variables into the process in order to quantify their effect on the uncertainty. An example would be determining the additional uncertainty due to placing the instrument on scaffolding.

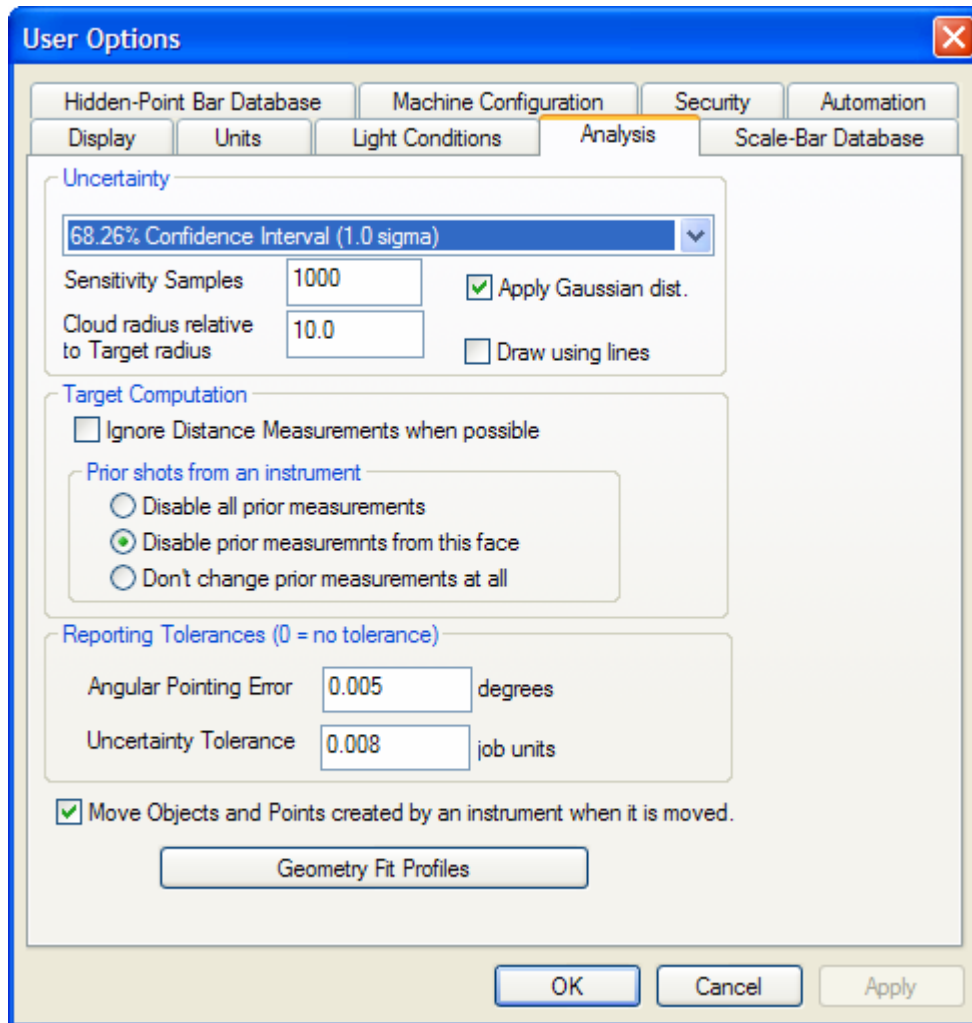
Data Analysis and Optimization

Given the measurements of each point in the target field from multiple instrument locations, the next task is to determine the transformations of the instruments that minimize the discrepancy between the observations. Once this is accomplished, the remaining residual errors will be analyzed to determine the uncertainty of the instrument.

Spatial Analyzer User Options

Options for uncertainty calculation and display in SpatialAnalyzer will help to present the results of the USMN analysis. The figure shows the Analysis tab within the User Options dialog in SpatialAnalyzer. The section on Uncertainty contains a number of parameters that control computation results and how the results are portrayed.

Selecting the confidence interval with the drop down selection control in the Uncertainty section sets the scalar value that is multiplied to the 1-sigma computed and stored uncertainty components. Uncertainty values that have a multiplier greater than one are considered expanded uncertainties. SpatialAnalyzer applies the confidence interval scalar to uncertainty values it reports. A comprehensive set of choices is available in the drop down selection control. They start at a 50% Confidence Interval (0.6745 sigma) and go to a 99.9999% Confidence Interval (5.0 sigma). The typical choices between the two are available.



User Options Dialog Analysis Tab

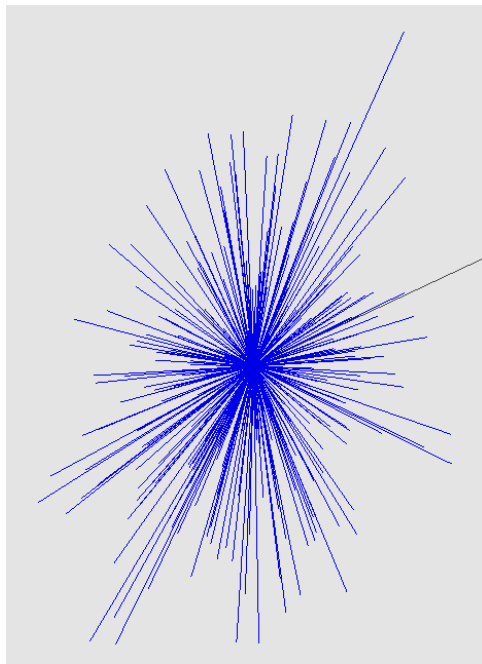
The value of Sensitivity Samples is used to provide a default number of samples when computing uncertainty fields in SpatialAnalyzer . By changing this value you can increase the number of samples and as such generate a larger number of statistical samples to draw inferences from. Computation time is improved by decreasing the sample count but the statistical conclusions are not as confident. Studies presented in chapter 7 showed that when the sample count is above 300 the results are generally within $\pm 5\%$ of an established true value.

The Cloud radius relative to Target radius value helps to control the scale of uncertainty clouds. Relative effects can be objectively evaluated in SpatialAnalyzer 's graphics view by adjusting the Cloud radius relative to Target radius parameter. This value generally defaults to 1.0 but for some networks a different value can be helpful to easily visualize and communicate the relative uncertainty expected throughout the network.

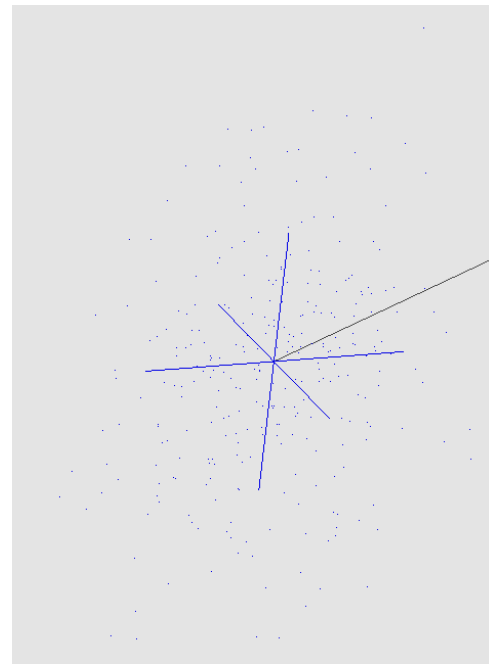
The Apply Gaussian option is on by default. When the Apply Gaussian option is off, SpatialAnalyzer will assume a Uniform distribution when developing samples for

analysis. In general for most metrology applications a Gaussian distribution more closely represents the distribution of real-world data.

Draw using lines applies to way uncertainty clouds are displayed graphically. This option is off by default. When Draw using lines is on uncertainty clouds are shown as in the figures shows the field in the default mode.



Uncertainty Field drawn with lines



Uncertainty Field drawn with pixels

Reporting for USMN

SpatialAnalyzer provides a wide array of reporting options for metrology data. This section will focus on three reporting mechanism that are applicable to uncertainty analysis. Coordinates and uncertainty values are reported relative to the current working frame in SpatialAnalyzer. Changing the working frame updates all of the reported coordinate and uncertainty values.

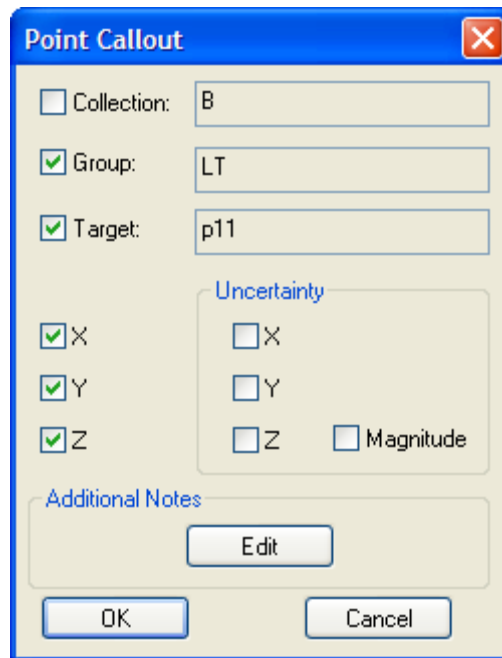
Callout View Reporting of Uncertainty

Callout Views in SpatialAnalyzer support options to display the computed uncertainties for points. To create or return to a Callout View click on the Callout View toolbar icon



in SpatialAnalyzer. Add a Point Coordinates callout with the button on the Callout View dialog. Select the option to create a Point Coordinates callout. Select a point with an uncertainty field associated to it and then an anchor point for the callout within the graphics view. After SpatialAnalyzer creates the callout, right-click on it and select the callout's Properties option. The figure shows an example Callout View Properties dialog.

Select the point attributes needed for the callout. The available attributes include Group name, Target name, X, Y, Z coordinates plus the X, Y, Z and Magnitude uncertainty estimates for the point.

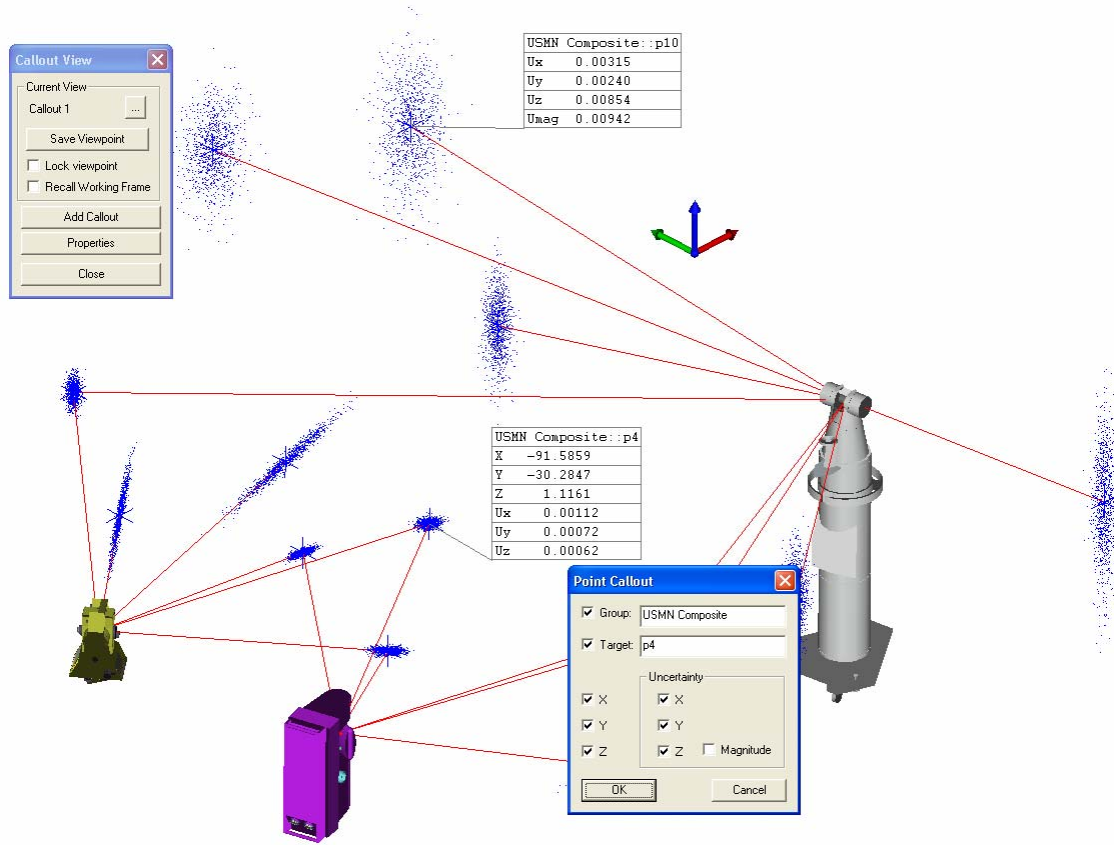


The image shows a 'Point Callout' dialog box with a blue title bar and a close button. It contains several input fields and checkboxes. The 'Collection' field is set to 'B'. The 'Group' field is set to 'LT' and has a green checkmark. The 'Target' field is set to 'p11' and has a green checkmark. There are three checkboxes for coordinates: 'X', 'Y', and 'Z', all of which are checked with green checkmarks. To the right of these is an 'Uncertainty' section with checkboxes for 'X', 'Y', 'Z', and 'Magnitude', all of which are unchecked. Below the coordinate checkboxes is an 'Additional Notes' section with an 'Edit' button. At the bottom are 'OK' and 'Cancel' buttons.

Attribute	Value	Checked
Collection	B	<input type="checkbox"/>
Group	LT	<input checked="" type="checkbox"/>
Target	p11	<input checked="" type="checkbox"/>
X		<input checked="" type="checkbox"/>
Y		<input checked="" type="checkbox"/>
Z		<input checked="" type="checkbox"/>
X (Uncertainty)		<input type="checkbox"/>
Y (Uncertainty)		<input type="checkbox"/>
Z (Uncertainty)		<input type="checkbox"/>
Magnitude (Uncertainty)		<input type="checkbox"/>

Callout View Properties Dialog

The figure shows Point Uncertainty estimates in Callout Views.

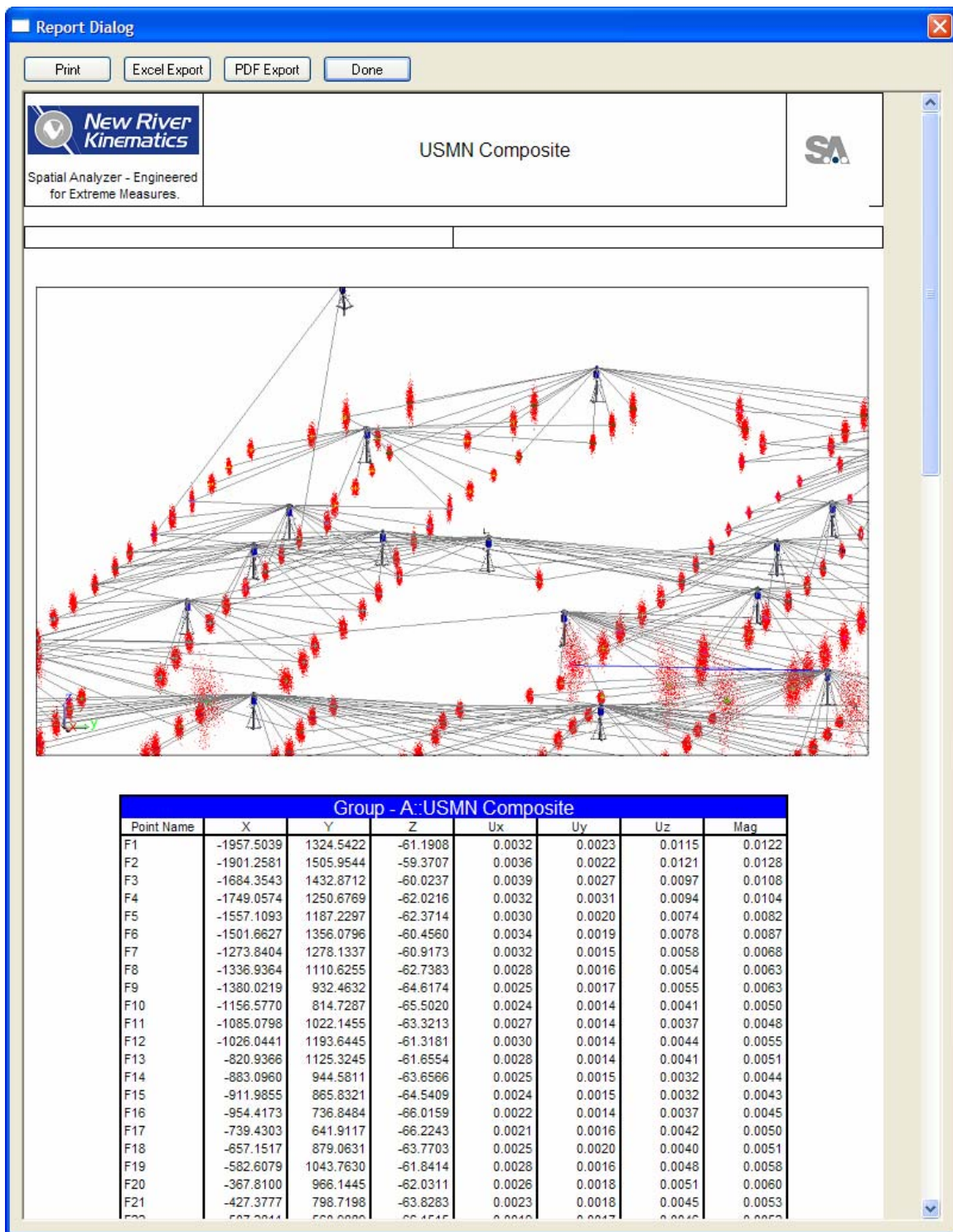


Point Callout Views with Uncertainty Reports

The reported values are relative to current working frame. Changing working frames will update the values to be consistent with the new working frame.

Quick Report with Uncertainties

A Quick Report of a point group with uncertainty fields documents the coordinates and target uncertainties. The report is accessible by right-clicking on the point group and selecting the Quick Report option. The uncertainty fields are reported with the active confidence interval. Set the confidence interval on the User Options >> Analysis tab.

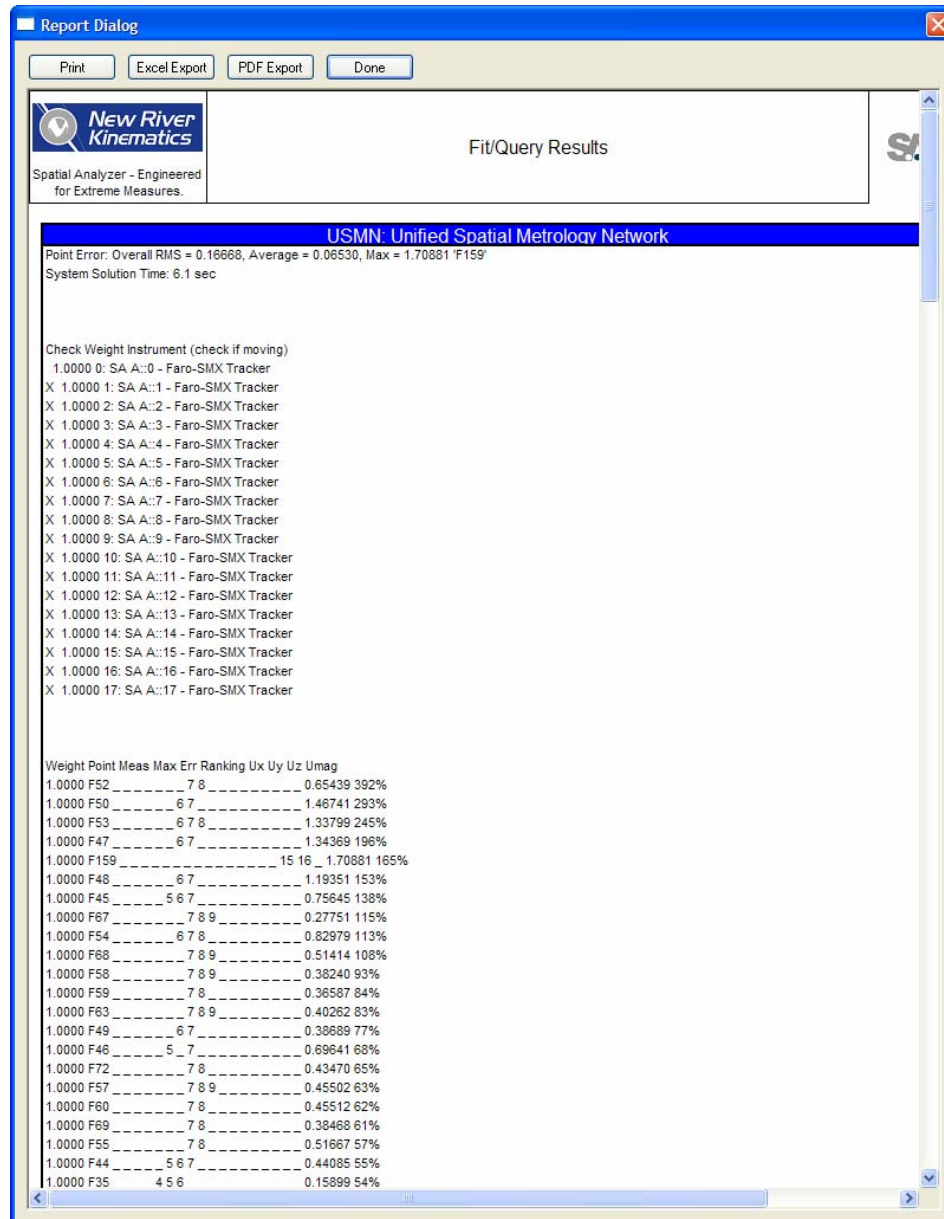


Coordinates Report with Uncertainties

Composite groups from USMN with uncertainty fields that appear in these Quick Reports. Report views in SpatialAnalyzer can be sent directly to a printer or into an Excel file.

Fit and Query Reports with Uncertainty Analysis Results

By selecting the Append to Report option on the USMN dialog a comprehensive synopsis of the analysis is retained in SpatialAnalyzer's database. To access this information select the [Reports >> Fit/Query Results] menu option. The figure below shows example USMN results in the Fit/Query Results report.



Fit and Query Report

Uncertainties of Individual Network Components

The goal of the Unified Spatial Metrology Network (USMN) is to properly combine nominal data and measurements from multiple coordinate acquisition systems to produce a measurement network complete with a realistic statement of the

measurement uncertainty. Previous sections described a method for simultaneously combining all of the coordinate acquisition systems and nominal data that compose a measurement network. This method will be extended here to incorporate the uncertainties of the individual network components in the combination process. This chapter will outline the concepts of the USMN and the next chapter will present the salient software implementation details.

Design Model Uncertainty

In cases where the goal of the measurement task is to relate the physical, measured object to the nominal CAD model, it is necessary to tie the measurements into the CAD coordinate system. In these cases, the model is added as a component in the USMN.

If there is a known uncertainty in the nominal tie-in points, this uncertainty may be represented by point uncertainty fields. These fields along with the rest of the CAD geometry are treated in the same manner as the coordinate acquisition systems included in the network. To provide a reference, however, the CAD model can be treated as the “ground” object or the one to which all uncertainty is referenced. This makes sense, because the primary goal of most measurement systems is to compare measured point locations to the product design model. It is important to note, however, that the choice of reference system is arbitrary and left to the user.

If the nominal model uncertainty is not available or if it is not to be considered, the model is still treated as a component of the network. This is because there will be uncertainty in the tie-in of the measurement systems to that model. In such cases, the nominal points will not have uncertainty fields, but the corresponding points from the measurement systems will have uncertainty fields that then result in uncertain tie-ins to the nominal coordinates.

Nominal data may also be used to provide a reference link between individual measurement systems or collections of measurement systems. In such cases, the nominal information not only provides a useful reference system, but it also serves as a critical link in the measurement network allowing the connection of separate measurement systems. This is another example where the nominal data is treated identically to the other measurement systems in the network. It is also important to note that multiple sets of nominal data may be used just as multiple coordinate acquisition systems are used in a network.

Solving the Unified Spatial Metrology Network

The optimization process used to solve the measurement network is an algorithm that computes optimal instrument movements. Individual instrument-station movement is computed to minimize measurement closure on common points. Previous discussions raised the issue of the weighting factors used in the point computation solution. The individual measurements are relatively weighed by their estimate uncertainty. This weighting scheme, applied to the point computation, is also applied to the objective function used in the total network solution.

Given reasonable uncertainty statements for the measurement components of an instrument, it is possible to determine the uncertainty field for a given measurement using the methods presented in Chapter 0. In this case, however, the goal is not to

determine the uncertainty field, but instead to describe the uncertainty of this measurement relative to the other measurements in the network. It is not necessary therefore, to compute an entire uncertainty field, but instead to represent the effect the instrument measurement component uncertainties will have on the optimization objective function components. To do this, we simply determine the worst-case effect (at 1 sigma) that the uncertainty values will have on the measurement and then weight the measurement accordingly.

Suppose, for example, a point in the network is measured by both a laser tracker, and a total station. Each instrument provides three measurement components consisting of two angular values and a distance. Given the differences in the instruments, however, the laser tracker provides distance measurements that are far more certain than those from the total station theodolite. This is due to the different technology employed in the instruments. In addition, the relative difference in this uncertainty is a function of where the measurement is in the workspace of the instrument. For this reason, it is necessary to determine individual weighting values for every measurement in the Unified Spatial Metrology Network relative to all other measurements. A suggested method for accomplishing this follows.

First, it is necessary to determine a representation for the discrepancies in the measurements. As discussed in previous sections, representing the error in terms of the instrument's measurement values is a logical choice. This representation was applied to the instrument uncertainty characterization process. In this case, the error is represented in terms of the instrument measurement values, but converted to linear distance as opposed to a mix of angular and linear values. This provides for an objective function homogenous in units. Also, a linear distance expression of error is often the most intuitive for the measurement technician.

Once the component errors are determined, the task remains to determine the weighting factor that should be applied to those components before placing them into the optimization objective function. Obviously, these weights must correspond to the representation of the residual error representation. In this case, the instrument measurement components are used to represent the error and state the instrument uncertainty so that requirement is met. To determine the weighting factors, we represent the component uncertainty of the instrument as a linear distance error term at the location of the measurement. For the laser tracker, total station, or other spherical measurement device, these uncertainty components are expressed as linear distances by

$$D_{\theta} = \ell \sin(U_{\theta})$$

$$D_{\phi} = \ell \sin(U_{\phi})$$

$$D_{\ell} = U_{\ell} + \ell U_{\ell_{PPM}}$$

where,

D is the linear distance representation of the component uncertainty.

ℓ is the distance from the instrument origin to the measured point.

U is the uncertainty of the instrument measurement components.

The $U_{\ell \text{ ppm}}$ term represents the parts per million (ppm) uncertainty component of the distance measurement technology in the device. Though this term is not commonly solved for in uncertainty extraction methods described in Chapter 0, it is possible to extract it from the measurements, or to simply use the manufacturer-supplied value. This requires large variations in measurement distance that are beyond the normal scope of industrial metrology. The ppm term is used in the equation for compatibility with conventional instrument manufacturer specifications.

Given the uncertainty of a particular measurement component in linear form, the next step is to determine the relative uncertainty of these components considering all other measurements in the network. This requires applying the equations or an equivalent function to all the measurements in the network. After determining all of these individual uncertainty values, a statistical metric is chosen to represent the overall uncertainty. In this case, the average, \bar{D} , is used. This yields the average linear distance uncertainty for all the measurements in the network based on the locations of the measurements and the instrument uncertainty characteristics.

This overall average uncertainty value is then used to revisit all of the individual measurements and create a weighting ratio for each component. For the spherical devices used in this example, the weighting factors are

$$W_{\theta} = \frac{\bar{D}}{D_{\theta}}$$
$$W_{\phi} = \frac{\bar{D}}{D_{\phi}}$$
$$W_{\ell} = \frac{\bar{D}}{D_{\ell}}$$

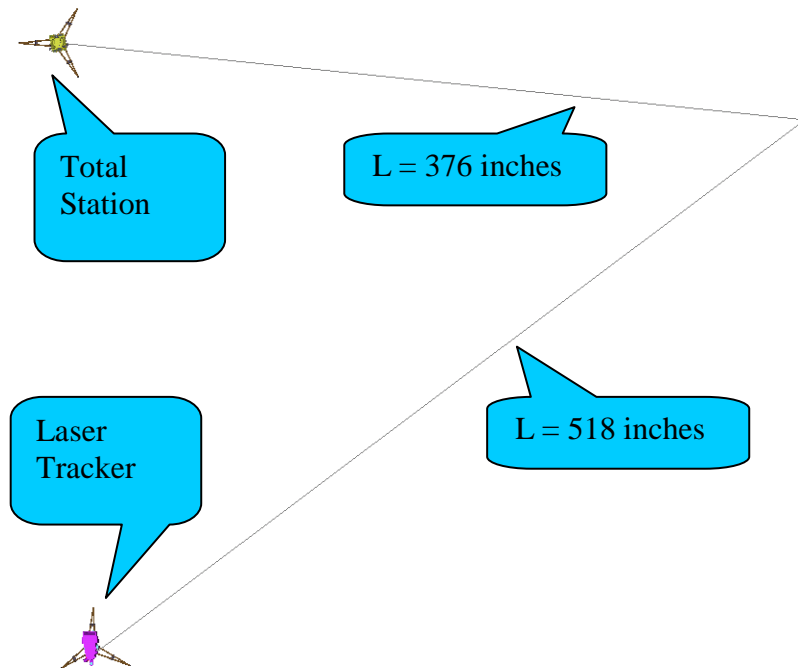
where,

W is the weighting factor for each component of the measurement.

The following section provides an example of this weighting process.

Example: Weighting by Relative Uncertainty

Consider the case mentioned previously where a common point is measured by both a total station and a laser tracker. This section provides an example of the weighting scheme applied to the geometry shown below.



Point Measured by Two Instruments

The uncertainty values for the measurement components of each instrument are listed in the Instrument Uncertainty Values table. These values are an input to the USMN and may be experimentally determined using the methods in Chapter 0 or obtained from manufacturer specifications.

		Total Station	Laser Tracker
Horizontal Angle	(arcseconds)	0.5	1.3
Vertical Angle	(arcseconds)	0.5	1.3
Distance	(inches)	0.03937	0.0006
	(parts per million)	2	0

Instrument Uncertainty Values

The common point measurement shown in the figure is the result of a complete set of measurement values from each instrument. These are shown in the Instrument Measurement Values for the Common Point table.

		Total Station	Laser Tracker
Horizontal Angle	(degrees)	306.418	54.737
Vertical Angle	(degrees)	100.192	123.621
Distance	(inches)	376.2864	518.2100

Instrument Measurement Values for the Common Point

Given the uncertainty model and the measurement values, specifically the measurement length, distance weighting equation is used to determine the effect of the component uncertainties in distance units. The results for the case are shown in Uncertainty at the Measured Point in Length Units table.

		Total Station	Laser Tracker
Horizontal Angle	(inches)	0.00091	0.00330
Vertical Angle	(inches)	0.00091	0.00330
Distance	(inches)	0.04012	0.00060

Uncertainty at the Measured Point in Length Units

Before proceeding with the actual weight computation equation, \bar{D} must be determined. In this example, the measurement shown in the figure is part of a much larger measurement network. For all of the measurements in that network, \bar{D} is 0.0098 inches. Applying the weighting equation to the uncertainties given in Uncertainty at the Measured Point in Length Units table yields the weighting factors in Measurement Component Weights Based on the Effects of Instrument Uncertainty table.

	Total Station	Laser Tracker
Horizontal Angle Weight	10.77	2.97
Vertical Angle Weight	10.77	2.97
Distance Weight	0.24	16.33

Measurement Component Weights Based on the Effects of Instrument Uncertainty

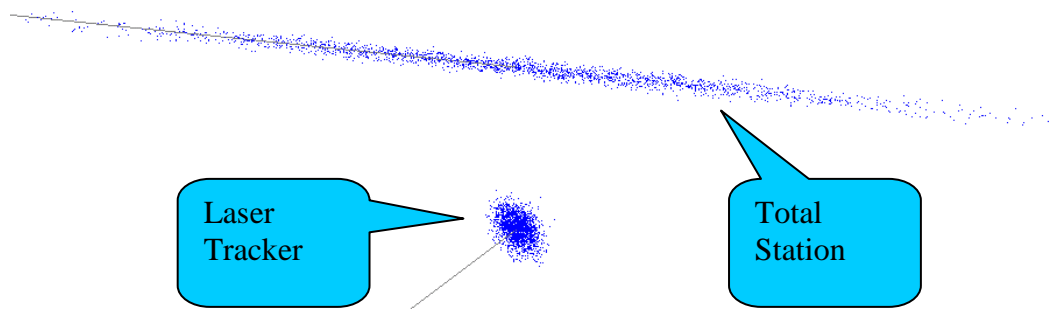
The next step is to apply these weight factors to the individual component errors that describe the residual error in the point computation problem. The residuals for this case are shown in Relative Uncertainty Weighting Applied to Residual Errors table along with their objective function contributions based on the component weights.

Total Station			
	Residual (inches)	Weight	Objective Contribution
Horizontal Angle	-0.00055	10.77	-0.00592
Vertical Angle	-0.00061	10.77	-0.00657
Distance	0.00825	0.24	0.00198
Laser Tracker			
	Residual (inches)	Weight	Objective Contribution
Horizontal Angle	0.00409	2.97	0.01215
Vertical Angle	0.00970	2.97	0.02881
Distance	0.00003	16.33	0.00049

Relative Uncertainty Weighting Applied to Residual Errors

The weighted residual errors are used directly in the objective function for the point computation optimization. This is the same objective function used for the total system solution since the errors in the overall system are the point residuals.

Visualization of the weighting scheme is best accomplished by viewing the uncertainty fields for the independent point measurements. For this case, the uncertainty fields are shown in the Uncertainty Fields for the Independent Point Measurements figure. Though the uncertainty fields are not used directly in the weighting scheme, they are another representation of the component uncertainties that are the basis for the weighting values.



Uncertainty Fields for the Independent Point Measurements

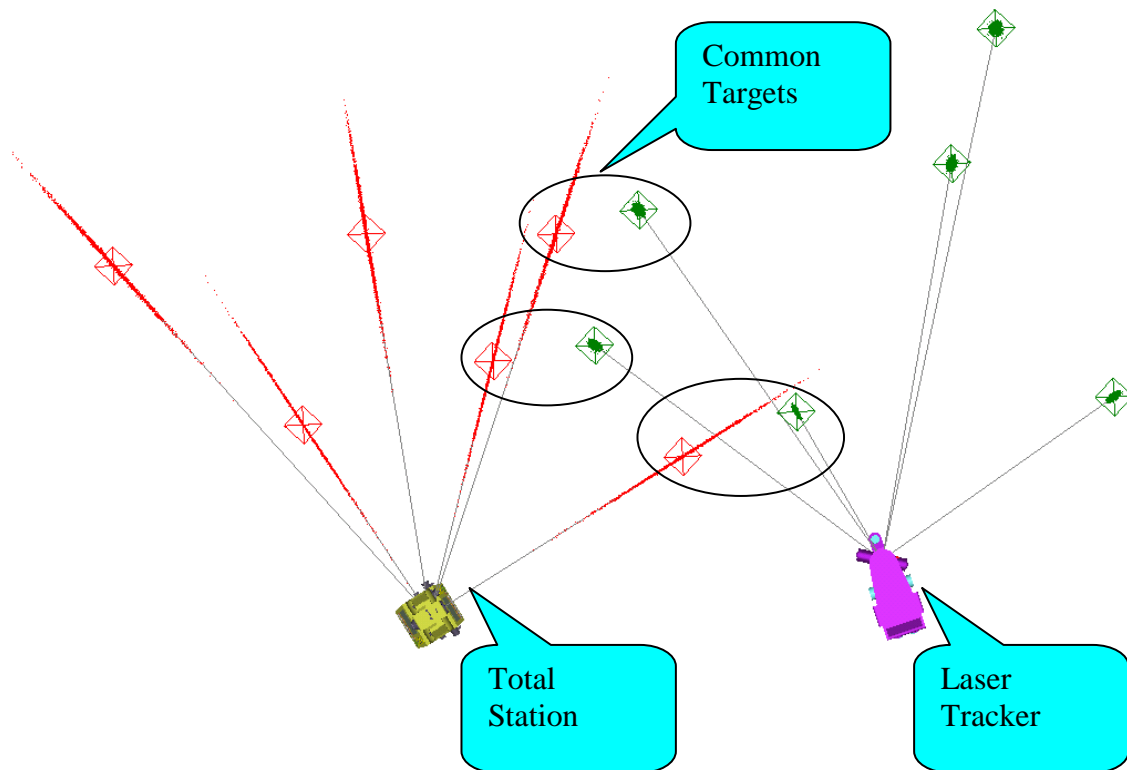
There are other weighting schemes that could be applied to the point computation problem. The key factor in selecting a weighting method is to include the instrument uncertainty in some manner. This is especially important in cases where the instrumentation is mixed within a network.

Forming Combined Coordinate Acquisition System Uncertainty Fields

The USMN is comprised of coordinate acquisition systems where each system contains point measurements in some instrument-based reference coordinate system and the corresponding uncertainty fields. This section will present a method for combining these

components and determining the uncertainty fields for each point resulting from the uncertainty interactions.

First, the actual measured network is solved simultaneously as described in the previous section. The actual measured values are used, ignoring the corresponding uncertainty fields. Consider the simple example of two coordinate acquisition systems measuring 3 common points. The figure below highlights the common measured points between a total-station theodolite and a laser tracker. It also shows the uncertainty fields for the individual measurements. By applying a simultaneous transformation optimization to the common points, they are brought together in a least-squares sense.

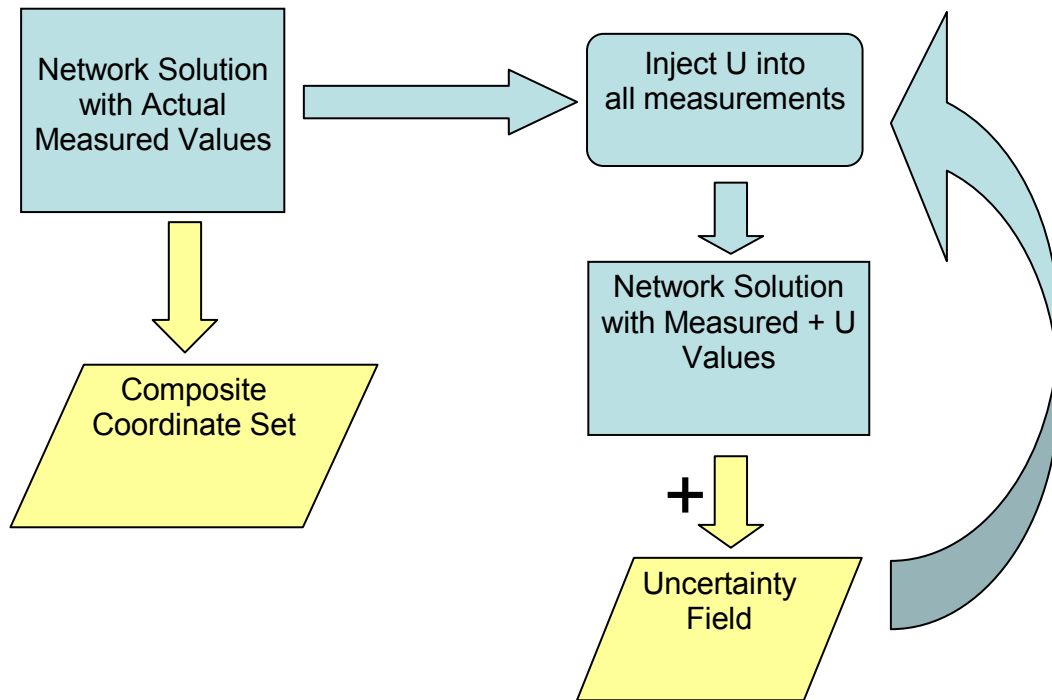


Coordinate Acquisition Systems with Individual Uncertainty Fields

After the transformation, the common points move together to produce a total coordinate model of the network where the coordinates are all referenced in a common coordinate system. The missing information is the uncertainty of the measured points relative to the other elements of the network. Notice the difference in the size and shape of the individual uncertainty fields. The ranging theodolite is much less accurate in range (distance) measurements than the laser tracker.

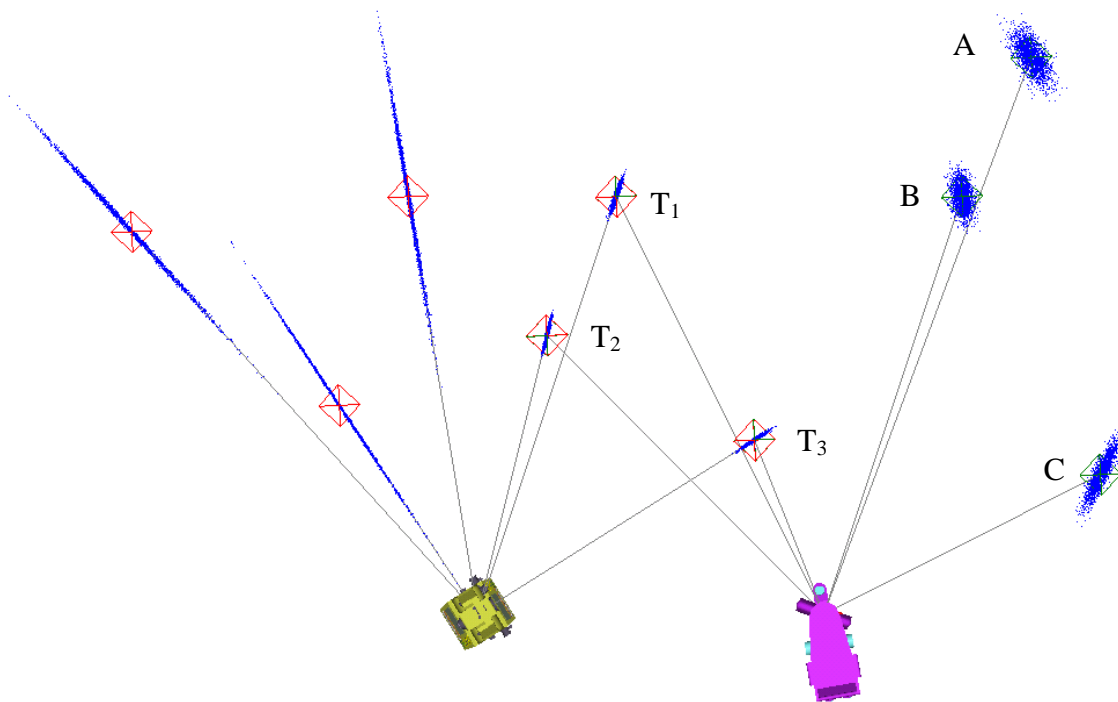
The next step is to repeat the simultaneous transformation process for the common targets using a different uncertainty field point for each iteration. In the first iteration, the first point in each target's uncertainty field is treated as if it were the measured target. The complete process occurs and the results are logged. On the next iteration, the second point in each target's uncertainty field is used and so on. Each iteration builds

another candidate value for the measured coordinate. This process is depicted in figure below. When all the points in the uncertainty fields are exhausted, the iteration stops and we are left with an uncertainty field representing the combination of the individual coordinate acquisition systems' uncertainties.



USMN Uncertainty Analysis Process

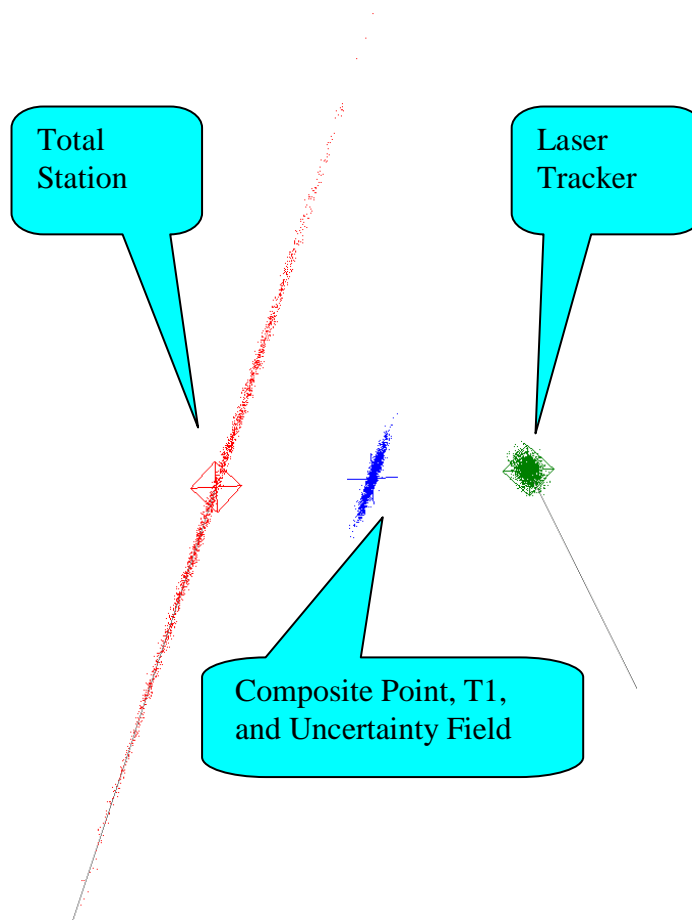
Combined Uncertainty Fields figure shows the results of this uncertainty analysis for the total station and laser tracker combination. The analysis includes 2000 uncertainty field points for each measurement.



Combined Uncertainty Fields

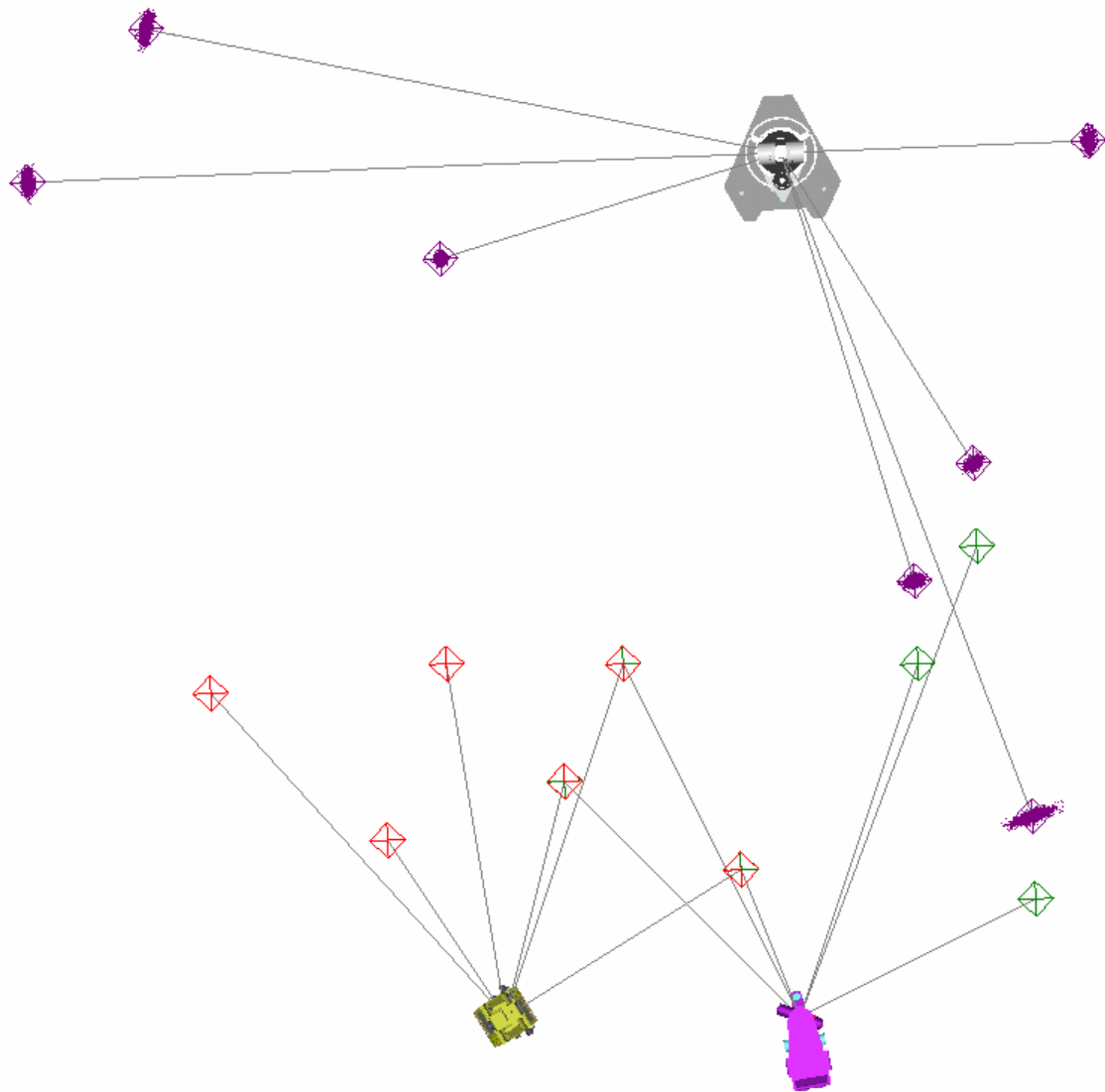
Notice the increase in the size of the uncertainty fields for the A, B, and C points that were measured with the laser tracker. Their clouds grew much larger due to the uncertainty in their tie-in to the theodolite measurements. Also notice the shape and density of the uncertainty fields. If traditional methods were used to predict the uncertainty of the laser tracker measurements, A, B, and C, without considering the interactions of the theodolite measurements' uncertainties in the common points, the actual uncertainty would be grossly underestimated.

Also notice the change in the shape of the uncertainty fields for the common points. In this case, these points are T_1 , T_2 , and T_3 . The optimal point is computed. There are many options for the weighting of the solution and the representation of the measurements. The result is, however, a single, optimal coordinate value. This value varies as the individual points in the measurement uncertainty fields are used to recompute its location. The result is an uncertainty field for the optimal point that is some combination of the fields from the associated measurements. The Effects of Combined Uncertainty Field Analysis on the Common Points figure depicts the results of this combination showing both the individual uncertainty fields and the combined uncertainty field for common target T_1 .



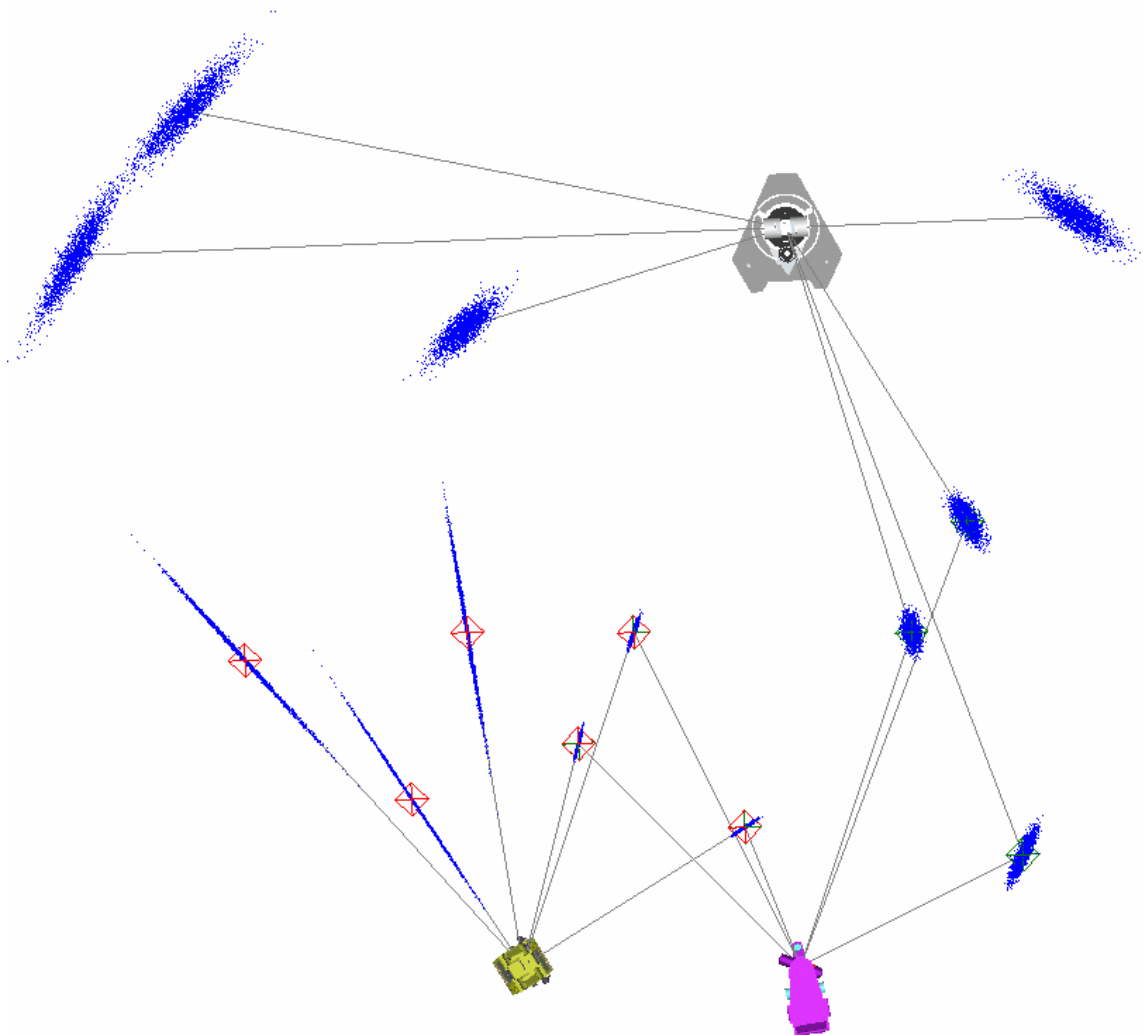
Effects of Combined Uncertainty Field Analysis on the Common Points

Suppose another coordinate acquisition system is added to this particular Unified Spatial Metrology Network. It must measure at least 3 points already defined in the model in order to tie its measurements to the network. For this example, another laser tracker is added and used to continue the measurement chain. The figure below shows this case before the solution. The uncertainty fields in the figure are those from only the new instrument.



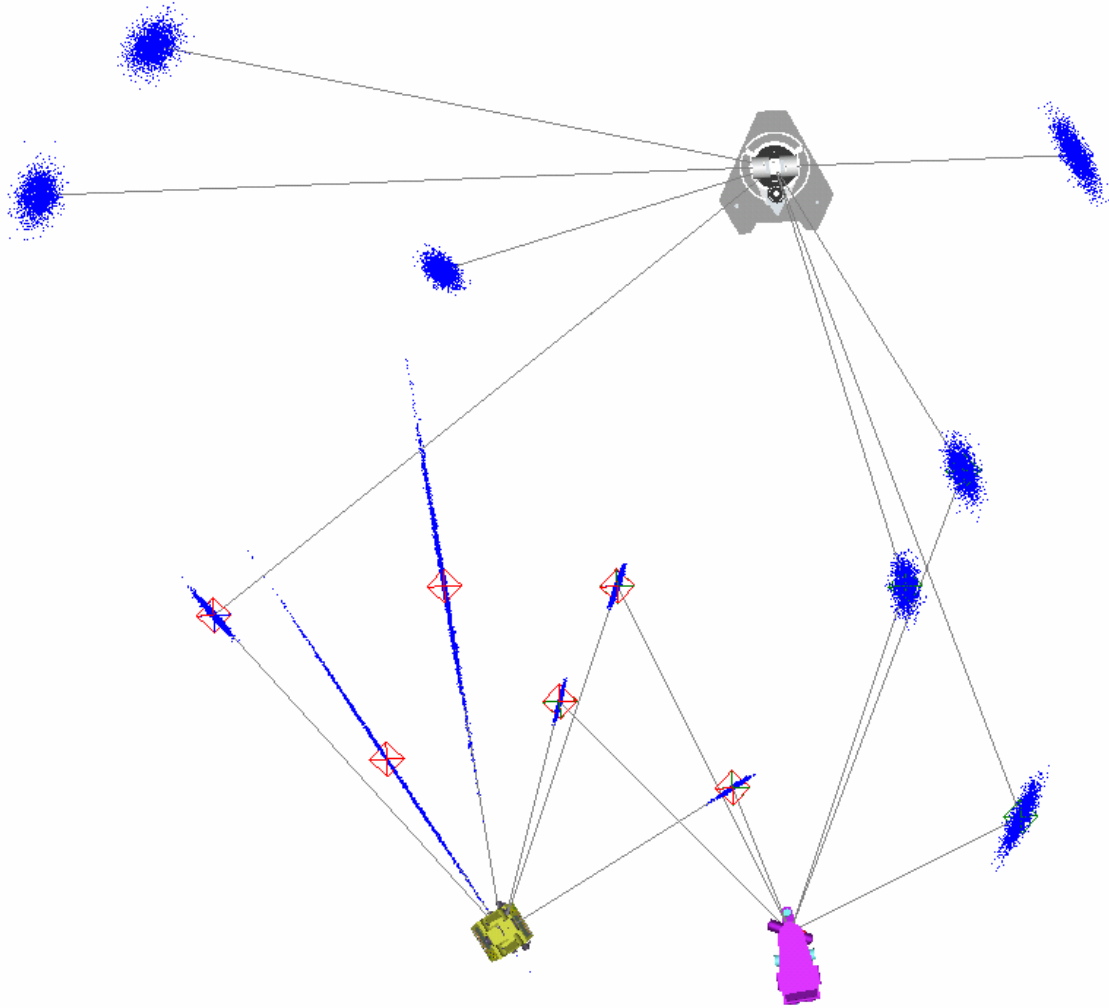
Additional Instrument Added to Measurement Chain

The results of this analysis are shown in the figure below. Notice the size and shape of the uncertainty fields for the measurements at the end of the chain. The uncertainty fields are much larger than those shown in the figure above because those were from only the last instrument in the chain.



Combined Uncertainty with Additional Instrument Added to Measurement Chain

If the network is modified by adding a single measurement that closes the chain, the uncertainty may be greatly reduced. This is because both ends of the chain are now referenced to ground. This looped configuration is shown in the figure below. The addition of more measurements to close the loop will continue to reduce the uncertainty.



Uncertainty Reduction by Closing the Measurement Loop

A summary of the USMN solution process is as follows. First, the network is solved using the simultaneous optimization process. Then, the solution is repeated for each of the uncertainty field points from each coordinate acquisition system. For each solution, the results are logged. Once all of the points in the clouds have been processed, the result is a new set of uncertainty fields for the points. These uncertainty fields represent the point uncertainties due to each instrument's uncertainty and the interactions of the uncertainties with other instruments in the combination process. The details of the software implementation of this approach will be presented in the next chapter.

Unified Spatial Metrology Network Results

Upon completion of the combined network uncertainty analysis, the USMN contains a new set of composite points representing the optimal location for all the points in the network. Each of these points is accompanied by an uncertainty field. This composite set is the output of the USMN.

The results may be displayed graphically as in the figures above or processed statistically to produce a summary. This statistical process yields an uncertainty statement for each measured point relative to any coordinate frame.

In addition, the variation in the instrument transformations is also available. For each iteration in the uncertainty field computation, the resulting instrument transformations may be logged to provide a transformation set at the conclusion of the process. By statistically processing this array of transformations, a statement can be made as to the uncertainty in the instrument location given the current constraints of the network. Though difficult to depict graphically, this information can be valuable to measurement planners.

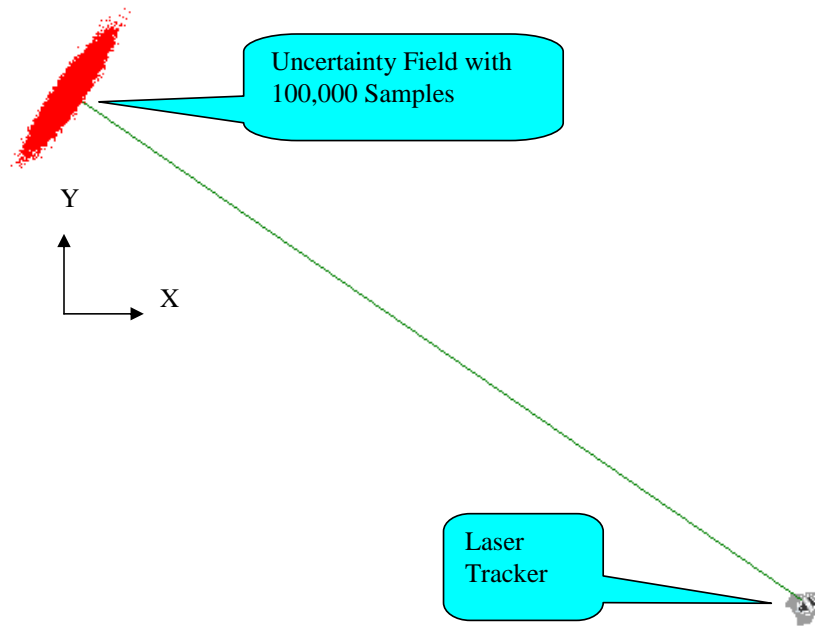
As with all numerical simulation algorithms, the question of sample size must be addressed. If computation burden was not an issue, all uncertainty fields could be populated with millions of points. Computation considerations are important in this case, however, since the goal is to make the uncertainty analysis readily available to those actually performing the measurements on the shop floor.

Uncertainty Field Density

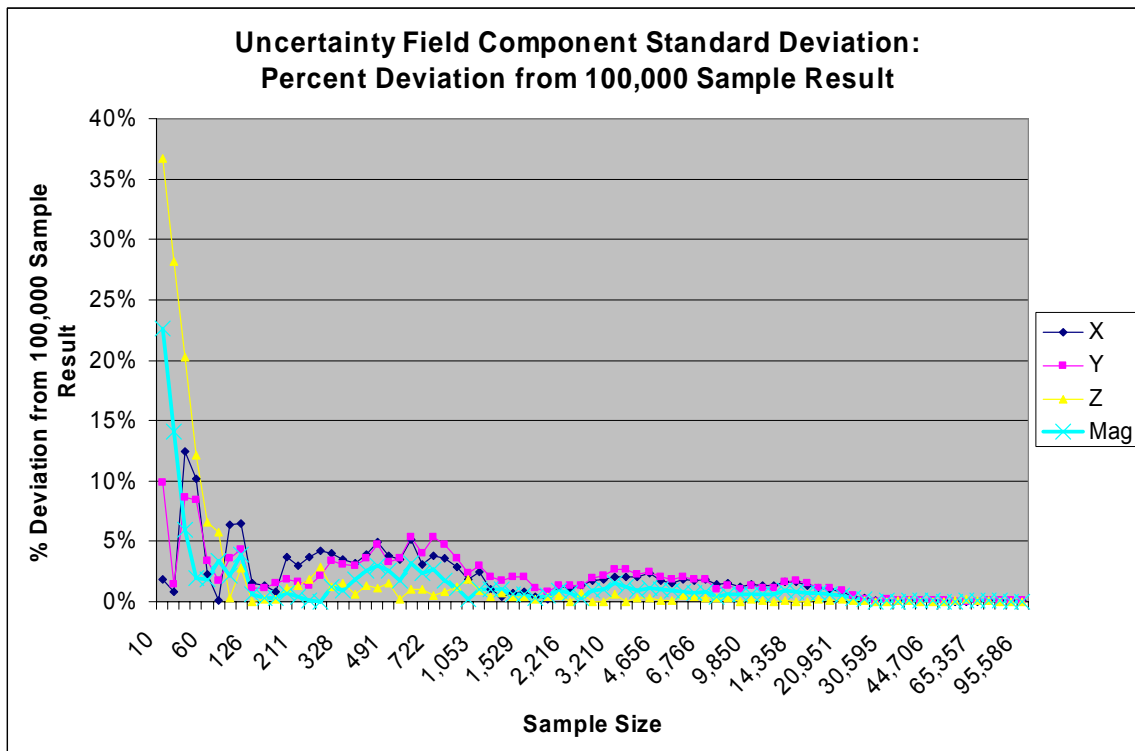
This section will address the effect of uncertainty field density on the statistical results of discrete field analysis. Field density also affects the usefulness of the graphical field representation, but that effect is not addressed in detail here as it is largely subjective and easily observed. In order to determine the effects of field size, the following procedure is used:

- Create a dense uncertainty field for a point.
- Compute the coordinate axis uncertainty values relative to the frame of reference.
- Store these values as the representative “population” statistic.
- Repeat this process using a subset of the field points and compare the results to the population statistic.
- Progressively grow the field and repeat the sub-analysis (10 points, 20 points, 40 points, etc).
- Graph the results to observe the convergence rate of the analysis to the “population” statistic.

For this test, the measurement geometry shown in the figure below was used to create an uncertainty field with 100,000 points. This field was then sub sampled as described above to provide uncertainty statements of the coordinate components at varying densities. The algorithm starts with the first 10 points from the field, and then adds 110% more points to the field so the next set contains 21 points. This process continues with each subsequent set containing the points from the previous set and the additional points from the “population”. The resulting uncertainty statements are plotted in the chart below as a percentage of the uncertainty for the “population” point field.

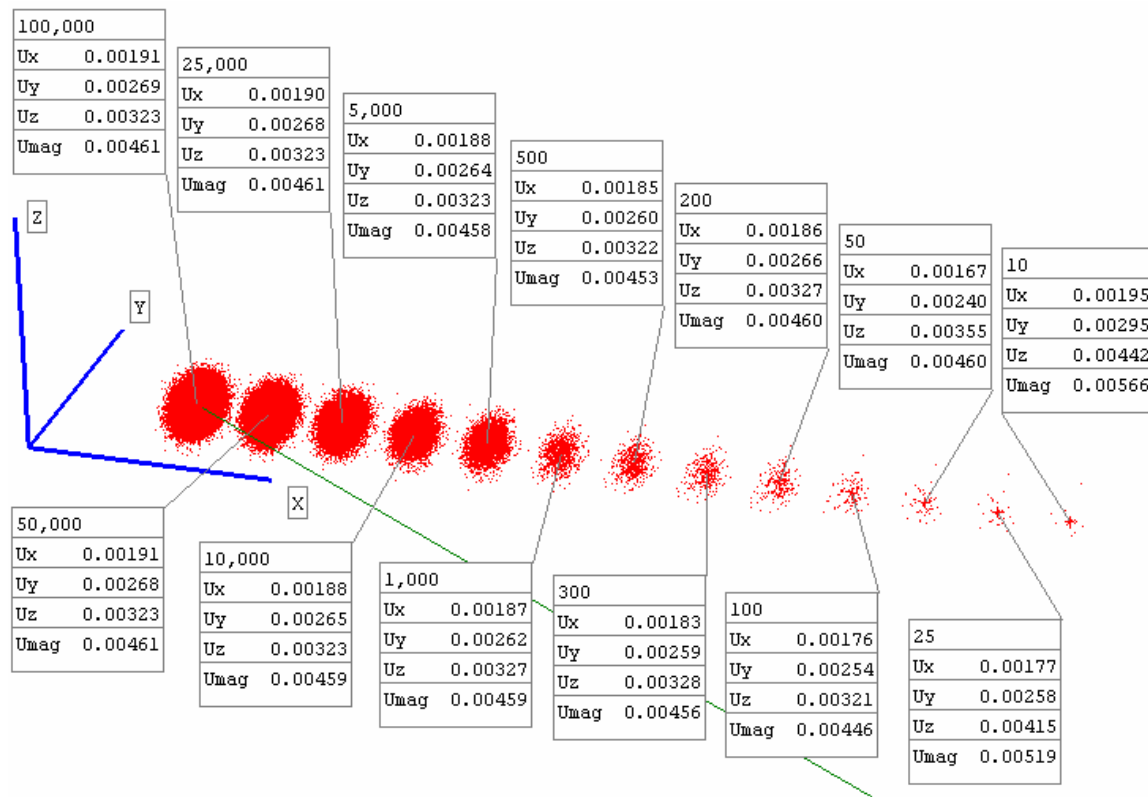


Uncertainty Field Density Test Geometry



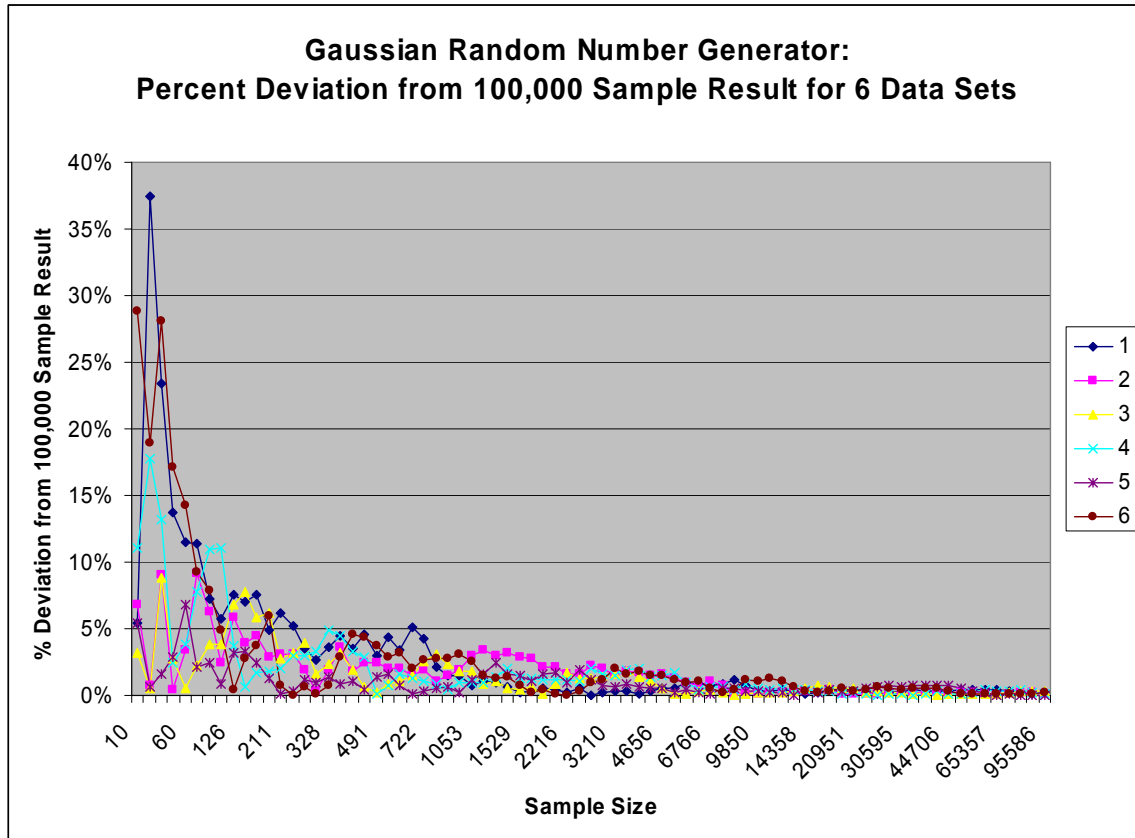
Uncertainty Field Component Standard Deviations as a Function of Sample Size

Though it requires many samples to converge to the dense cloud uncertainty result, after approximately 200 – 300 iterations, the results are within approximately 5% of the value. This “uncertainty of the uncertainty” is well within the needs of the measurement users. In addition, since the uncertainty statements are based on a simulation of the component errors, it is optimistic to expect the results to match the actual instrument uncertainty to within 5%. To provide a visual reference for the effects of density on both the graphical depiction of the fields and the numerical statements, the figure below shows multiple copies of the uncertainty field at varying densities. To produce the sub sampled clouds, the sample was taken starting at the beginning of the cloud point set and proceeding to the desired number of points then discarding the remaining points. This is the same sub-sampling strategy used in the previous analysis with slightly different sample sizes. This strategy is used since it most closely matches the issue we are trying to address. That is the question of when to stop sampling and rely on the statistics. By re-using the initial cloud points, the process is simulating the results obtained at various stages of the simulation.



Uncertainty Field with Varying Density

The rate of convergence of the sample statistics shown in the chart to the final value is heavily influenced by the random component of the uncertainty simulation. This was done by creating 6 sets of 100,000 Gaussian random numbers. For each set, the standard deviation was computed using an identical sampling strategy to that used for the uncertainty field points. The results are shown in figure below.



Gaussian Random Number Generator Performance as a Function of Sample Size

The convergence characteristics of the random generator are similar to the characteristics of the uncertainty field sampling shown in the chart. Based on the results of this uncertainty field density test, the network case studies presented in the follow sections typically used 300 points per uncertainty field.

Measurement Analysis within the Unified Spatial Metrology Network

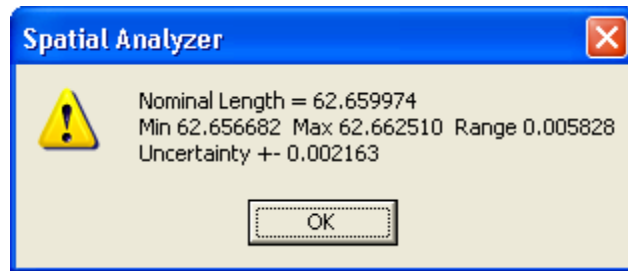
This section addresses applications of coordinate uncertainty fields beyond the uncertainty in a measured coordinate value. Often, the measurement task is not only to determine XYZ coordinate values of points as measured by the instrument, but also to evaluate the true, as-built, shape of an object. In these cases, the coordinate values measured by the instrument are only an intermediate result in the metrology job.

This means that in order to determine the uncertainty of the actual geometrical analysis operation, other methods must be used. The discrete point cloud uncertainty fields are well-suited for such an analysis.

Distance Uncertainty between Points

The uncertainty between two points can be computed with a function within SpatialAnalyzer . Use the menu selection Query >> Sensitivity Cloud to >> Sensitivity

Cloud. Select any two points that have uncertainty fields associated with them to complete the process. SpatialAnalyzer computes the geometric uncertainty between the points. Example results is shown below.



Point to Point Distance Uncertainty

The analysis includes the nominal length, minimum and maximum distances between the points. A range between the max and min is also provided in the dialog. A nominal uncertainty is reported based on the standard deviations of all the distances pairs between the two uncertainty fields.

Geometric Object Fitting

The fitting of measured data to geometrical shapes is quite common in measurement analysis. One of the most common is the sphere fit. Tooling ball measurement is common since it is often used to corresponding common measurements or transform into nominal coordinate systems. The measurement device may measure several hundred points on the tooling ball, perform a best-fit sphere analysis, and return the center of the sphere as the XYZ value. In this case, a statement of the uncertainty in the sphere fit should accompany the XYZ center point of the sphere.

Using the discrete point cloud representation of the uncertainty field, it is a straightforward task to determine the effect of the measurement uncertainty on the resulting geometrical fit. A description of this process follows.

First, a geometrical fitting operation is performed on the actual measurement values, or the optimal common points as determined in the simultaneous system combination portion of the USMN solution. This result is stored as the baseline for the geometrical properties. In the case of a sphere-fit, the results are a center point (X, Y, and Z), and a radius.

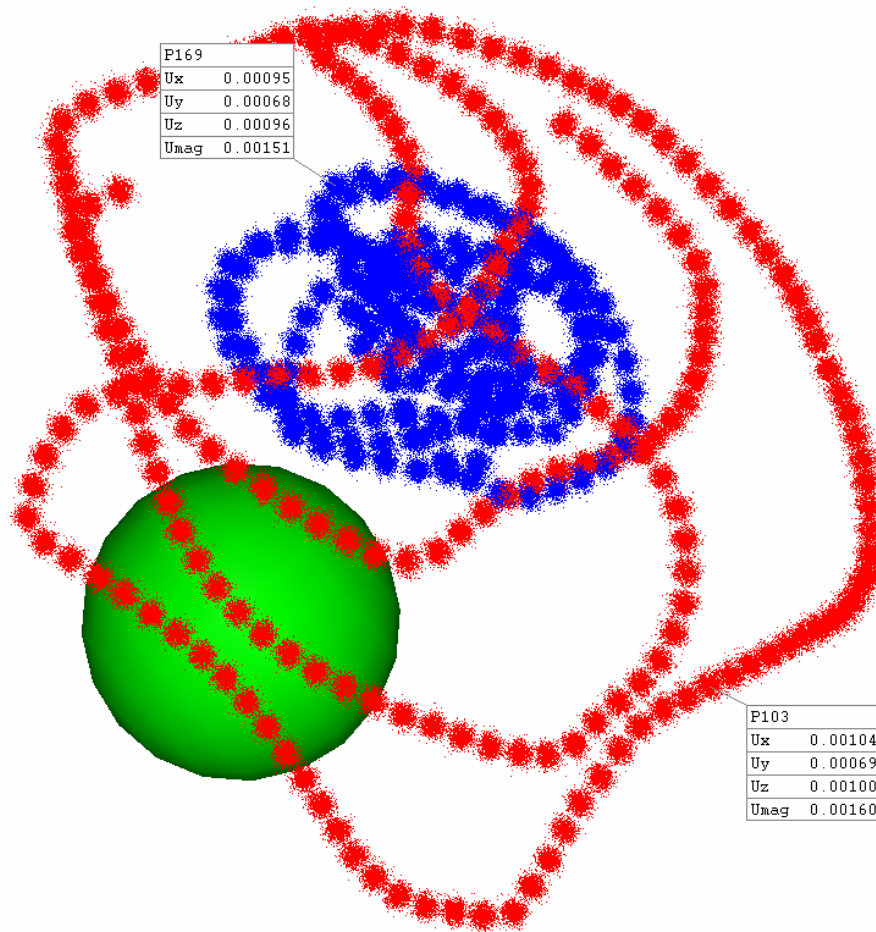
Next, the geometric fit operation is repeated for all the points in the uncertainty fields. The result is a list of geometrical parameters representing the "uncertainty field" of the geometry. These sets of parameter values may be evaluated using methods similar to the uncertainty field numerical evaluations. By computing the standard deviation of each of the variable lists, and applying the desired confidence interval, a fit uncertainty statement may be given.

If the uncertainty fields for the points in the fit contain a different number of uncertainty field points, the largest quantity is used in the analysis. When the uncertainty field index exceeds the total number of points in the smaller clouds, they just loop back to the first point and reuse the cloud.

The following geometry fitting examples demonstrate the geometry fitting analysis capabilities of the USMN.

Sphere Fit Uncertainty Example

The measurement of a tooling ball with a laser tracker is accomplished by sweeping the laser tracker target across the sphere to obtain sufficient coverage for a sphere fit. Typically, laser tracker targets are 1.5 inch spheres with a center-mounted retro-reflector. Tooling Ball Measurement using a Retro-reflector Target figure shows two sets of measurement data on a ½ inch tooling ball. One set shows typical coverage of the sphere while the other shows inadequate coverage.



Tooling Ball Measurement using a Retro-reflector Target

Notice that the data is offset from the actual sphere. This target offset is one of the factors that must be properly handled in all measurement analysis. In the case of a sphere fit, this is accomplished by adjusting the objective function so that the optimal solution is one where the measured points are ½ of the target diameter away from the best-fit sphere.

As discussed previously, the analysis begins with a computation of the best-fit sphere for all the actual measured points. This is stored as the nominal value to which all subsequent fits are compared. Then, the first point from each cloud is used and another fit is performed. The results are tabulated. This process is repeated for the second point, third, and so on until all the points in the cloud are exhausted. The result is a tabulation of sphere fit results. By applying basic statistics to these results, a statement of their uncertainty is made.

In this particular case, the entire process was repeated twice, once for the small data set, and then again for the larger data set. The results of this analysis are shown in Sphere Measurement Coverage Comparison table.

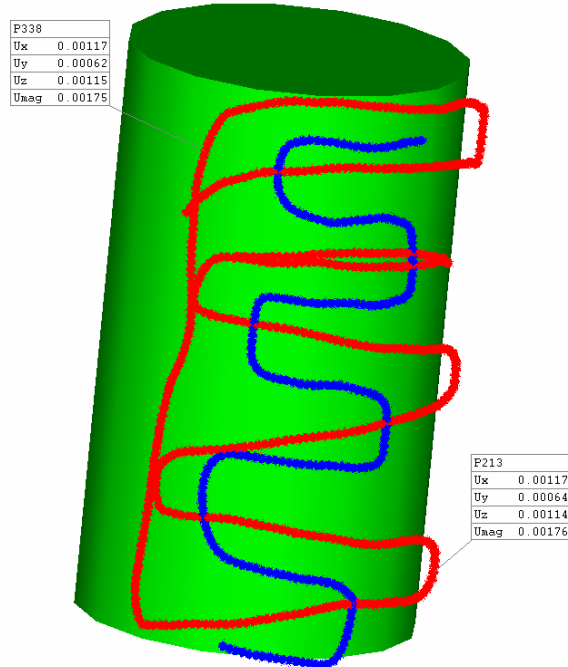
	Small Coverage	Large Coverage
Sphere Center:		
Ux	0.0018	0.0003
Uy	0.0027	0.0004
Uz	0.0012	0.0002
Umag	0.0035	0.0005
Udia	0.0067	0.0008
Analysis time (P-4 1.3 GHz)	3.3 sec.	3.7 sec.
Points in Fit	186	222
Field Points	1000	1000

Sphere Measurement Coverage Comparison

This example not only underscores the importance of adequate measurement coverage when fitting a sphere, but it also demonstrates the need for uncertainty analysis. Depending on the measurement instrument and the shape of the uncertainty fields, the uncertainties in geometrical fits can be significant.

Cylinder Fit Uncertainty

Applying these methods to other geometrical shapes is a straightforward process. In this example, two sets of cylinder measurement data are investigated. The first maintains a thin area of coverage on the surface of the cylinder, while the second covers a larger portion of the surface. The analysis methods are identical to those in the previous example with the exception of the parameters that are tabulated. The figure below shows the actual measurement data sets used in this example and the resulting best-fit cylinder. Cylinder Fit Uncertainty Results table shows the numerical results of the analysis and provides a comparison of the two measurement geometries.



Cylinder Fit Measurement Geometry

	Small Coverage	Large Coverage
Cylinder:		
U Axis (deg)	0.0021	0.0009
U Diameter (inches)	0.0027	0.0005
Analysis time (P-4 1.3 GHz)	14.7 sec.	32.3 sec.
Points in Fit	217	495
Field Points	1000	1000

Cylinder Fit Uncertainty Results

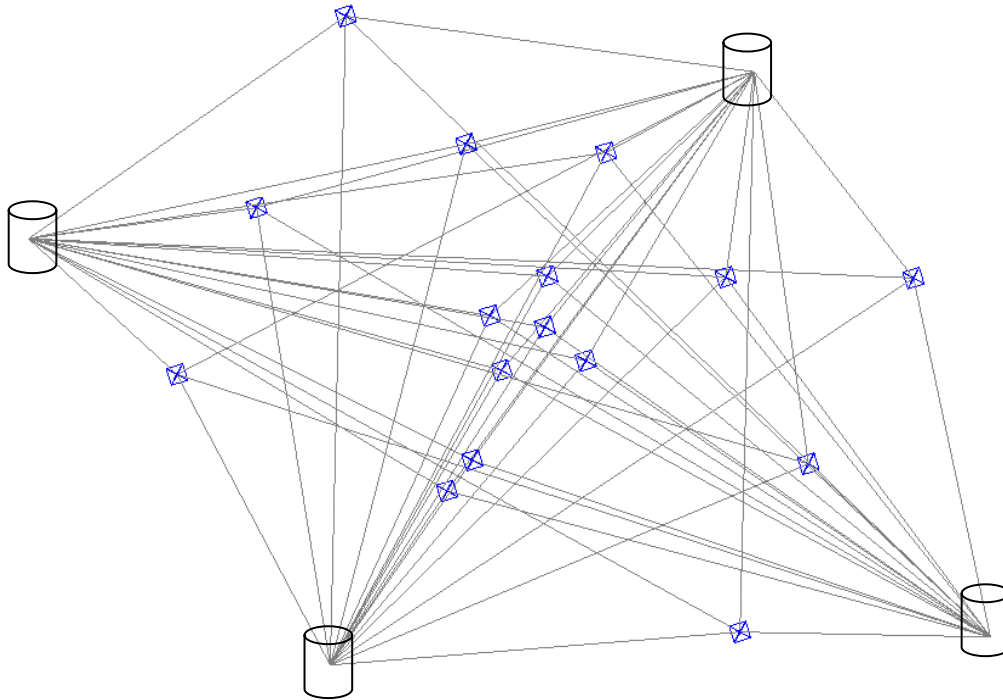
Notice the reduction in the uncertainty of both the axis of the cylinder and the diameter in the case of the large coverage measurements.

Case Study: Laser Tracker Performance Survey

The following case study is provided as an example of the instrument uncertainty characterization process. In this example, the goal is to survey the performance of three brands of laser tracker measurement systems and provide insight into the performance of these systems under realistic conditions. The manufacturers are identified as A, B, and C for anonymity. The detailed analysis for one particular set of the data from instrument B is presented while the other analyses are summarized. For all measurements, only the interferometer range measurement was used for the distance component as opposed to the less accurate, albeit more convenient, absolute distance meter (ADM).

The measurements for this process were taken in different locations by different technicians from a variety of corporations. This data was acquired by participants in the standards committee, . The goal of this body is to develop shop floor uncertainty statements and system health checks for laser trackers and other optical devices.

The target field geometry for one particular set of data acquired with instrument B is shown in the Point Field Geometry figure. At each instrument location, target data has been acquired from an upper and lower position. This vertical shift was accomplished by adjusting the height of the instrument stands.



Point Field Geometry

In this measurement scenario, the instrument in question was moved to 8 locations (including the upper and lower stand positions) throughout the workspace. At each location, the 16 fixed points in the field were measured. The instrument transformation optimization process, described in Section 0, was performed on the data. The system size was:

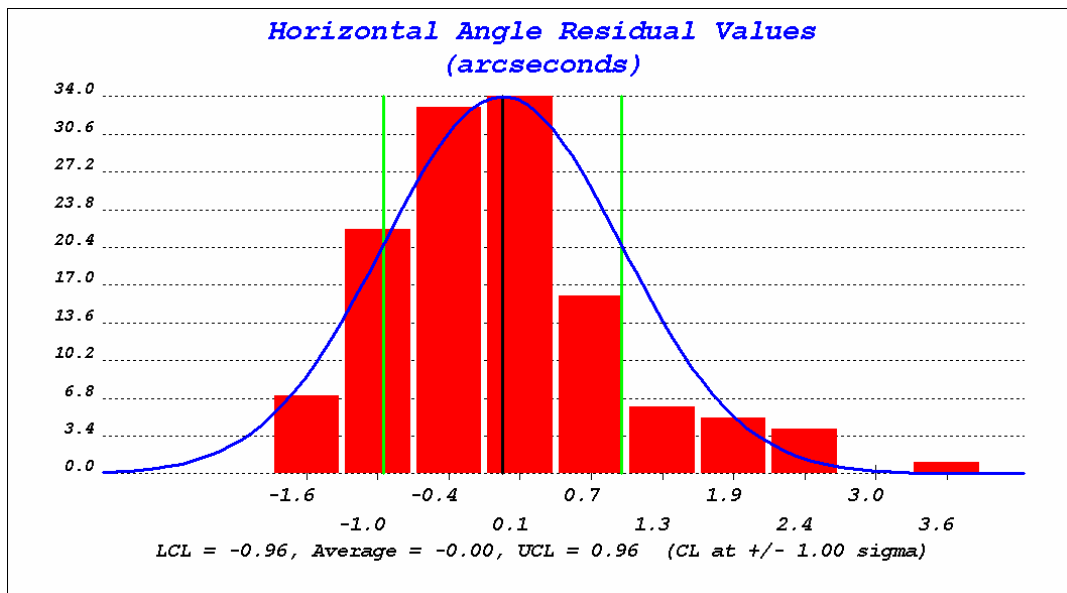
$$I = 16$$

$$K = 8$$

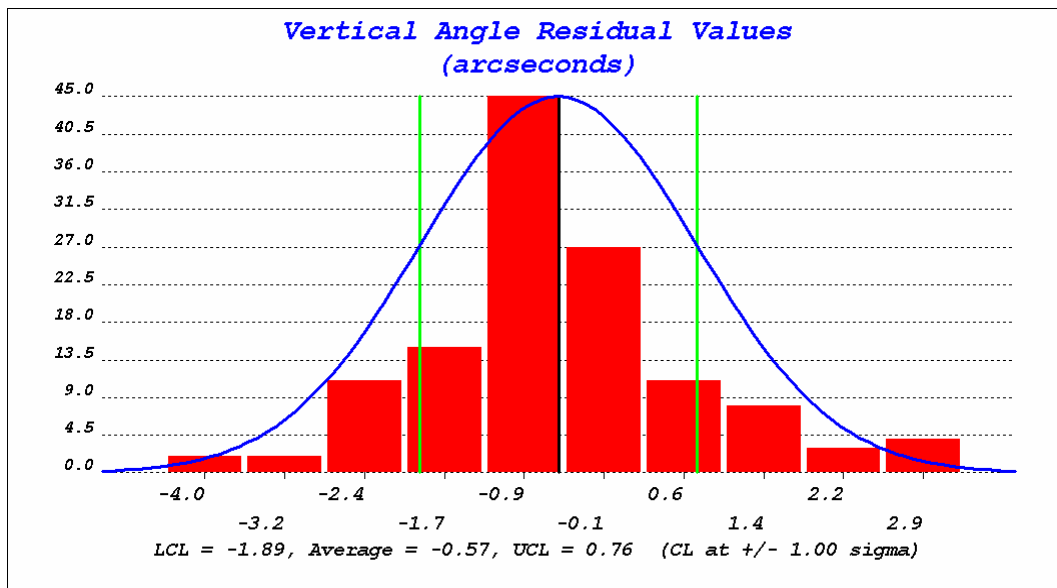
$$\text{Variable count} = 6(K - 1) = 42$$

$$\text{Equation count} = \tau I K = 384$$

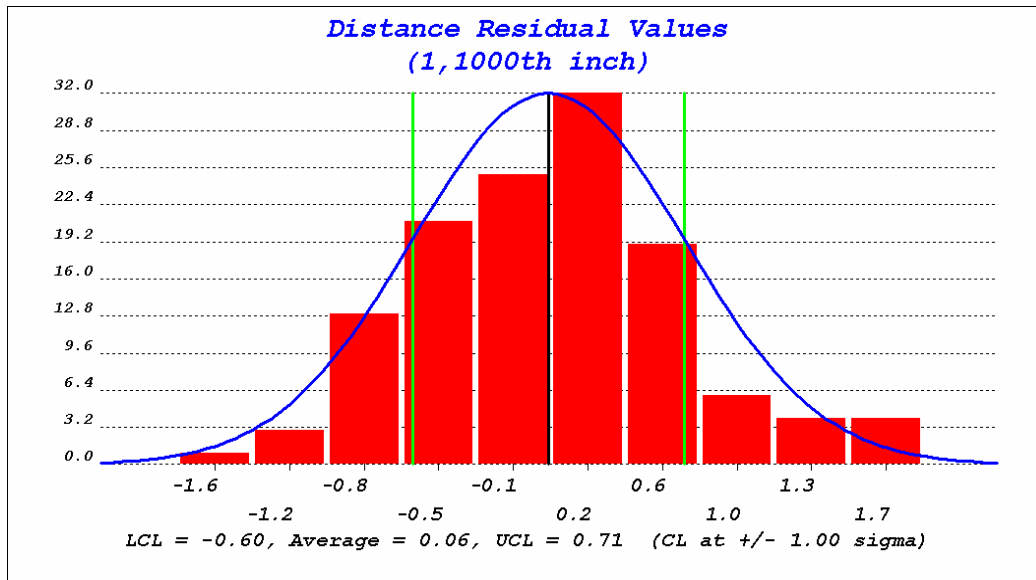
This means that the optimization must solve a system matrix of dimension 384 x 42. In this case, a basic weighting scheme was used for the purpose of simplicity. A weight value of 0.5 was applied to the angular values while a value of 1.0 was applied to the distance measurements. More advanced weighting options could also be used in this process (Section 0). At the completion of the optimization, 384 residual errors remain. By dividing these residuals into horizontal angle, vertical angle, and distance, 128 residuals are available to represent the uncertainty of each component. This data is shown in the figures below.



Optimization Residuals: Horizontal Angle



Optimization Residuals: Vertical Angle



Optimization Residuals: Distance

Calculating the standard deviation, as described in, for each component yields the uncertainty values shown in the table.

Component	Uncertainty
Horizontal Angle	0.96 arcseconds
Vertical Angle	1.44 arcseconds
Distance	0.000658 inches

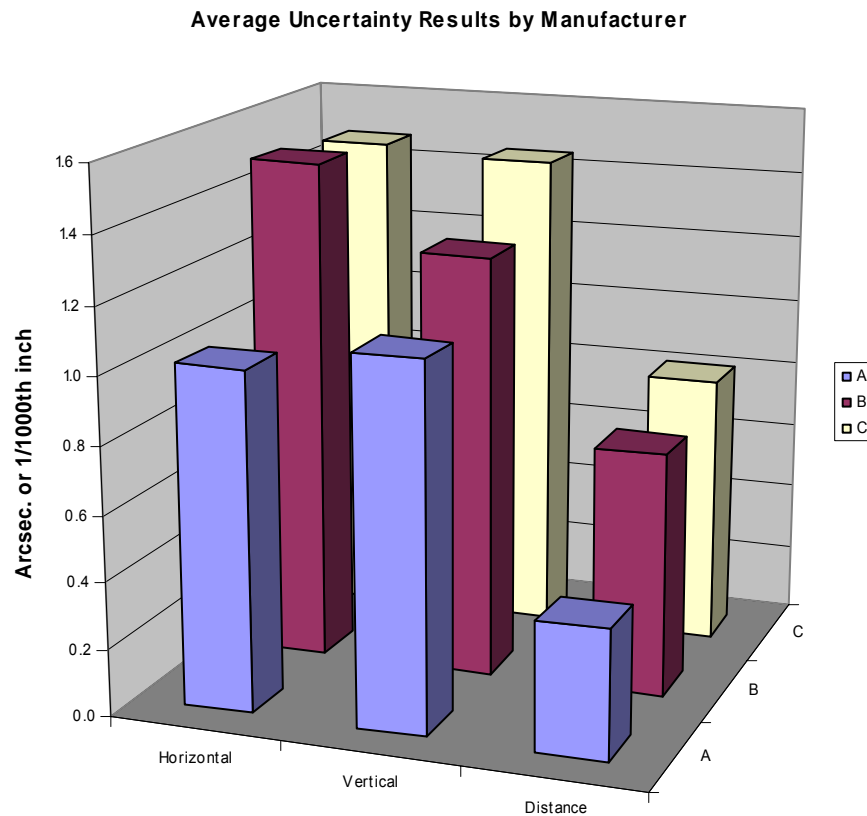
Instrument Component Uncertainty for Instrument "B" table

This process was repeated with several measurement operators in different locations with different instruments even within the particular brand. All the data was analyzed using the same algorithms and in fact, the exact same software package. The results of each measurement session and analysis are shown in Uncertainty Analysis Results by Measurement Session table. The index numbers for measurement location represent the grouping of measurement sets within each instrument. Location 1 for instrument A is not the same as location 1 for instrument B.

Manufacturer	Measurement	Horizontal	Vertical	Distance
	Location	(arcseconds)	(arcseconds)	(1,000th inch)
A	1	0.6	0.4	0.2
	1	0.6	0.4	0.2
	2	1.1	1.5	0.5
	3	1.4	1.5	0.7
	4	1.3	1.7	0.3
	Average	1.0	1.1	0.4
B	1	1.8	2.9	1.5
	1	0.9	0.8	0.1
	2	1.0	1.4	0.7
	2	1.1	0.5	0.4
	3	2.2	1.3	0.5
	3	1.8	1.0	1.0
	3	1.8	1.0	1.0
	Average	1.5	1.3	0.7
C	1	1.8	1.9	1.0
	1	1.9	1.8	0.4
	2	1.0	0.9	0.4
	3	1.4	1.3	1.3
	4	1.2	1.3	0.7
	5	1.3	1.4	0.6
	6	1.7	1.5	1.4
	Average	1.5	1.5	0.8
A, B, C	Average	1.3	1.3	0.6

Uncertainty Analysis Results by Measurement Session

It is important to note that these measurements were taken in different regions of the country, using different instruments, different point field geometries, different instrument locations, various environmental conditions, and various states of instrument calibration/compensation. To perform a more rigorous comparison between individual makes and models of laser trackers, it is necessary to test them in similar conditions with the same instrument and point geometry. The average results, grouped by manufacturer, are shown graphically in the figure below.



Average Uncertainty Results by Manufacturer

The next section will present a method for converting the measurement device output value uncertainties to coordinate uncertainties.

Application of Uncertainty Characterization to Instrument Performance Evaluation

Given the instrument uncertainty characterization methods presented in this chapter, it is possible to monitor the performance of an instrument on a periodic basis. Instrument operational checks are common in the measurement industry, but these typically do not cover a wide range of the instrument workspace and only check a subset of the performance.

Evaluation of performance requires a touchstone for comparison. This means the first step is to perform the uncertainty characterization on a well-calibrated instrument. This provides a baseline uncertainty value for the instrument's performance. The instrument is then placed into service. This usually includes packing, shipping, measuring, and enduring a variety of environments. In this scenario, the normal instrument operational checks are performed periodically. The instrument may still be within the valid calibration timeline recommended by the manufacturer. In addition, suppose the operational checks for the device are providing slightly higher error values, but are still

within the specifications for these values. Nothing would indicate the need to re-calibrate or certify the device at this point.

If a more detailed analysis of the instrument's current state of operation is desired, a user could perform an uncertainty characterization. Since the purpose of the analysis is performance evaluation, not complete uncertainty characterization, a large set of points and instrument locations is not required. Instead, the same process is implemented with fewer points and locations than a typical uncertainty characterization. After performing the network optimization, the residuals are analyzed in an identical manner to the total uncertainty characterization process. The result is an uncertainty statement for the measurement values.

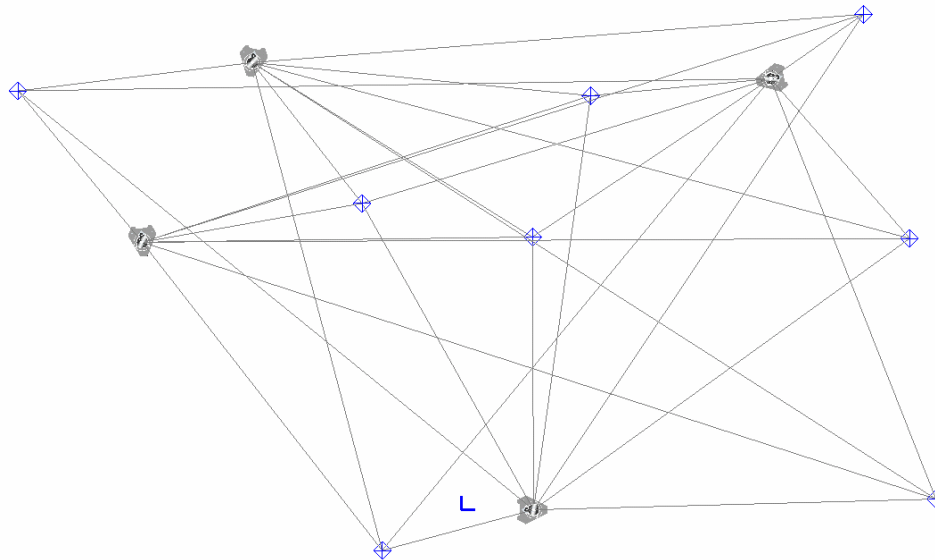
By comparing this result to the touchstone measurements, the user can easily determine if the performance has been degraded significantly. In addition, by taking additional data and repeating the operation with a more thorough data set, a revised uncertainty statement may be issued. This is important because the uncertainty characterization some time after calibration may be more realistic of true instrument performance than the idealized results obtained immediately after calibration.

The following example presents the measured uncertainty results before and after an instrument compensation process.

Instrument Performance Evaluation Example: Before and After Instrument Compensation

This section will briefly present the results of uncertainty characterization measurements on an instrument before and after partial instrument compensation. The purpose is to demonstrate the drastic effects of instrument calibration drift and the ability of the uncertainty characterization method to detect these deviations.

In this case, a laser tracker was used for the test. Four instrument locations were selected and eight points measured from each location. The geometry is shown below.



Instrument Performance Verification Test Geometry

After collecting the measurements, the analysis process described in Section 0 is used to determine the uncertainty for the instrument measurement components. For this test, the results are shown in the table below. These results are significantly worse than the typical uncertainty values for a laser tracker. This indicates the instrument performance may be suspect.

Component	1 Sigma Uncertainty	Typical Performance
Horizontal Angle	3.47 arcseconds	1.3 arcseconds
Vertical Angle	11.45 arcseconds	1.3 arcseconds
Distance	0.0087 inches	0.0006 inches
Total measurements	32	

Instrument Uncertainty Before Compensation Table

For this particular device, the instrument manufacturer provides a compensation procedure that is a subset of the total calibration. The intent is to determine compensation values for only those parameters that are most likely to change between full calibrations runs. Based on instrument performance, the manufacturer has determined this minimal set and provided a procedure for computing the values. Given

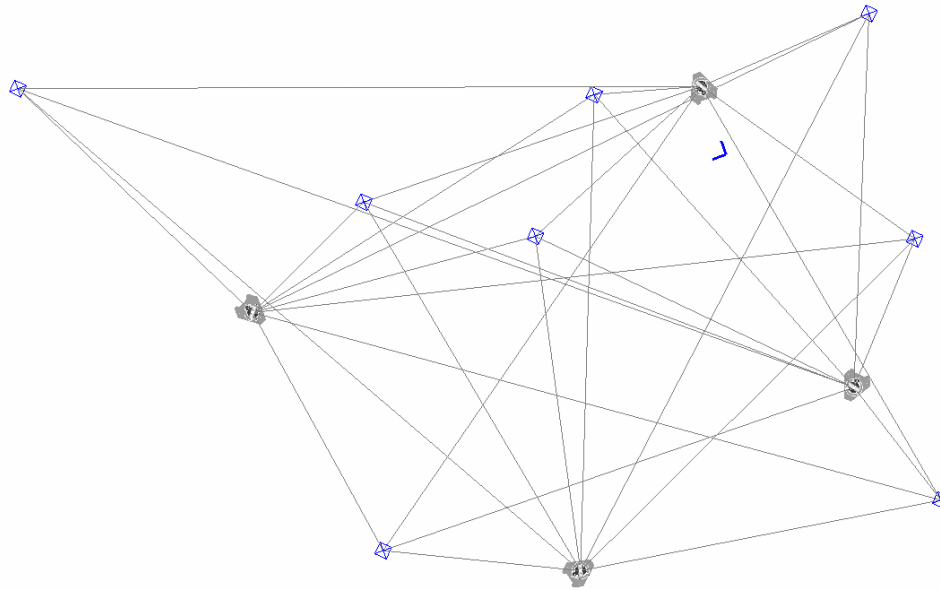
the poor performance shown in the table, a partial compensation was performed on the instrument.

This particular instrument's partial compensation routine uses a series of measurements to determine the vertical index error, the axis tilt, and the mirror tilt. This procedure and the selection of the parameters most likely to require adjustment is based on manufacturer experience with the device and its performance in the field. The procedure is described in detail by Markendorf (1999). There is also a separate but straightforward compensation procedure for the distance measurement component. It determines the distance from the kinematic center of the device to the home location of the instrument. Using the compensation procedures and computational routines provided by the manufacturer, the compensation results were as shown in the table.

Parameter	Before Compensation	After Compensation	Delta
Vertical index (arcseconds)	17.820	20.736	2.916
Axis tilt (arcseconds)	27.864	25.272	-2.592
Mirror tilt (arcseconds)	-10.692	-10.692	0.000
Home distance (inches)	6.09528	6.09661	0.00134

Instrument Partial Compensation Results Table

After applying the compensation results to the instrument's internal kinematic model, another performance test was performed using the same 8 fixed target positions and 4 instrument locations. Note that the instrument locations used in this case were not identical to those used in the initial performance evaluation. It is not required that the stations be similar, only that they provide a sufficient range of operation for the instrument. The geometry for the post-compensation evaluation is shown in the figure below.



Instrument Performance Evaluation Geometry After Compensation

The uncertainty analysis was performed in an identical manner to the pervious analysis. The results are shown in Instrument Uncertainty Before and After Compensation Table.

Component	1 Sigma Uncertainty Before Compensation	1 Sigma Uncertainty After Compensation	Typical Performance
Horizontal Angle	3.47 arcseconds	0.91 arcseconds	1.3 arcseconds
Vertical Angle	11.45 arcseconds	1.18 arcseconds	1.3 arcseconds
Distance	0.0087 inches	0.000598 inches	0.0006 inches
Total measurements	32	32	

Instrument Uncertainty Before and After Compensation Table

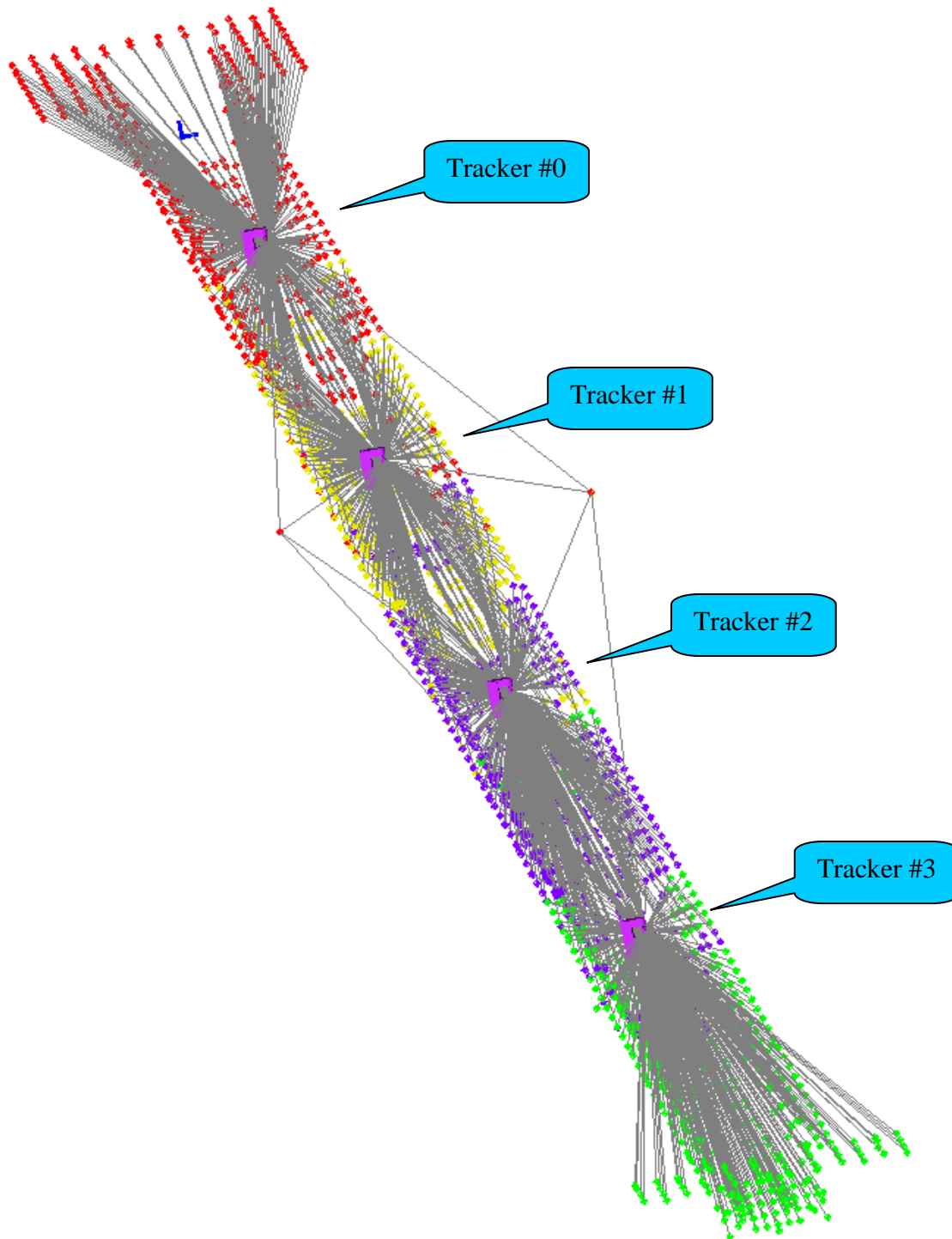
The significant reduction in the component uncertainties is reassuring since it provides the user with confidence in the manufacturer's compensation process. In addition, it highlights the ability of the uncertainty characterization methods presented here to detect degradations in instrument performance and notify the operator of the effect on the uncertainty of subsequent measurements.

It is possible that the operator may not wish to run any further compensations on the device, but instead proceed knowing the instrument is not performing to its fullest potential. In this case, the user could use the uncertainty values determined in the performance evaluation as the input for the uncertainty field analysis methods. The result would be a statement of the uncertainty in points measured with the instrument in the degraded condition. If this uncertainty is well within acceptable limits for the measurement job at hand, no action need be taken.

Aircraft Carrier Catapult Alignment

During the various fabrication stages of an aircraft carrier, one of the metrology tasks is the alignment of the steam catapults used to launch aircraft. Northrop Grumman Newport News Shipbuilding constructed the Ronald Reagan, CVN 76. In the construction process, it is necessary to quantify the alignment of the key support points along the catapults for the purposes of determining the shim thickness at various locations needed to provide optimal alignment. This alignment is critical since it reduces the frequency of catapult component maintenance.

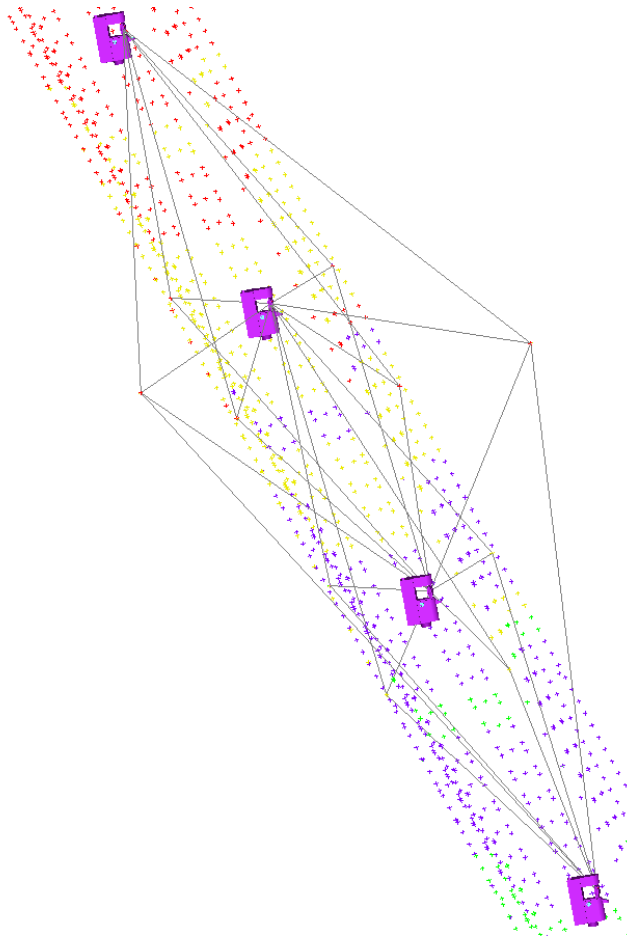
Given the geometry of the catapult, the number of points that must be measured, and the desired accuracy, a laser tracker was chosen as the best type of instrument for the application. Due to the large quantity of measurements required and the physical constraints of the measurement environment, it was necessary to use four laser trackers spaced along the catapult trough. The figure below shows the measured points along with the locations of the instrument stations. Note that the measurements from each instrument are color coded to indicate their grouping and measurement rays have been drawn from each instrument to each point it measured.



Catapult Measurement Geometry

The traditional analysis method used for this measurement job was to combine all of the stations into a common reference frame using sequential best-fit transformations as described in a previous section. This works reasonably well since there is a good

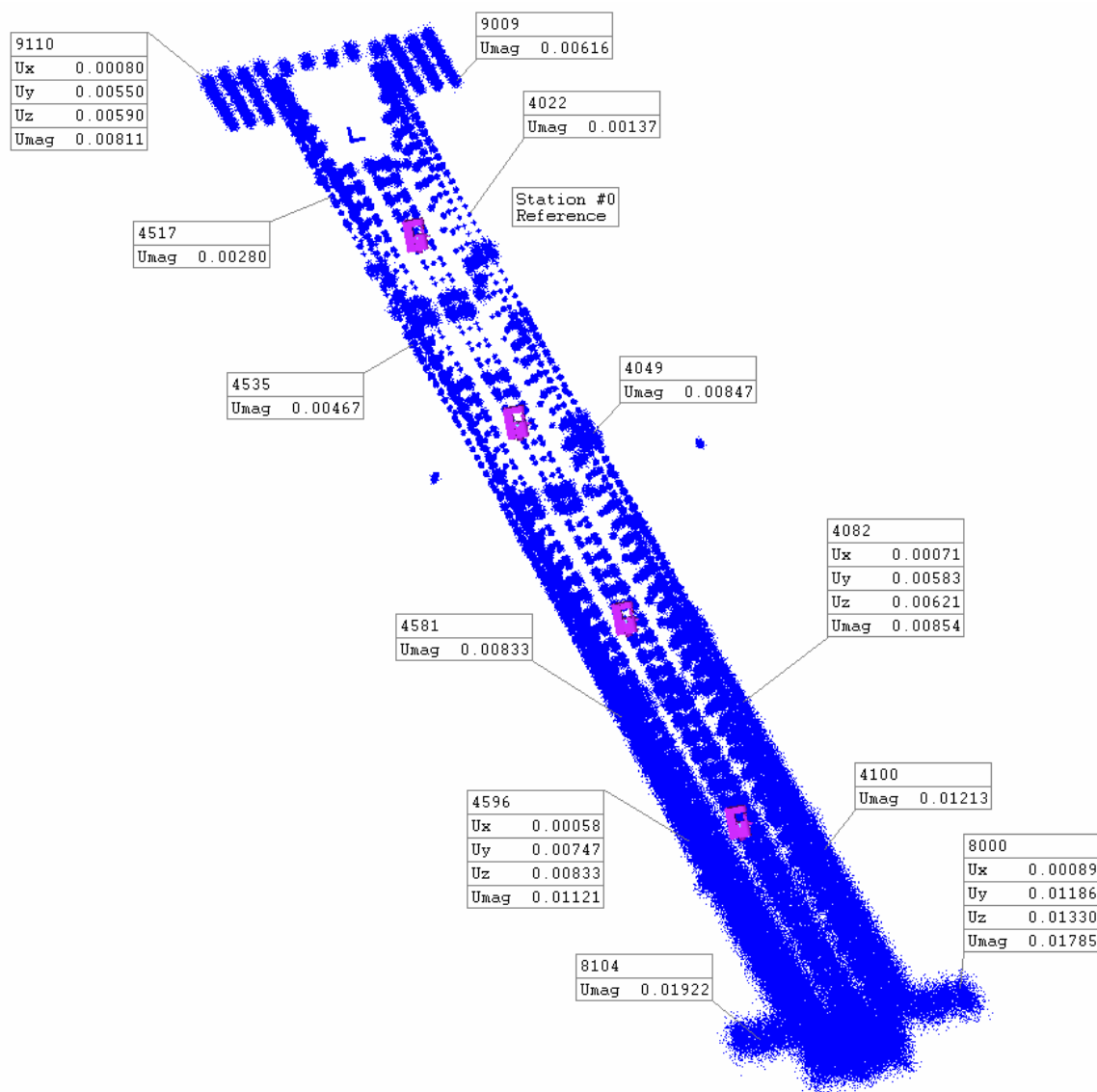
degree of overlap between the measurement stations. This method does not, however, make full use of the available measurement data since each successive transformation only considers the points in common with the two stations being transformed at that time. Although it is difficult to see in the figure above, there are two points near the middle of the trough that are measured by all four of the stations. There are in fact, a total of ten other points in the network that are measured by more than two stations. These points provide additional information that may be used to further tighten the measurement network and reduce the overall uncertainty in the measurements. The figure below shows these points by hiding the other measurements.



Additional Network Connectivity Used with USMN Method

Using the USMN, the measurement network was solved making use of the full network connectivity. In addition, the relative uncertainty weighting scheme was used to further exploit the instruments accuracy. Given a solution for the network, the next step is to solve the network repeatedly in order to determine the uncertainty fields in the network. This was done using 300 samples, and a laser tracker uncertainty model with horizontal and vertical angle uncertainty of 1.3 arcseconds and a range uncertainty of 0.0006 inches. The results are shown graphically in the USMN Uncertainty Field Results: Station #0 as Reference figure. The figure also shows the total magnitude uncertainty values for selected points in the network and full coordinate component uncertainties for

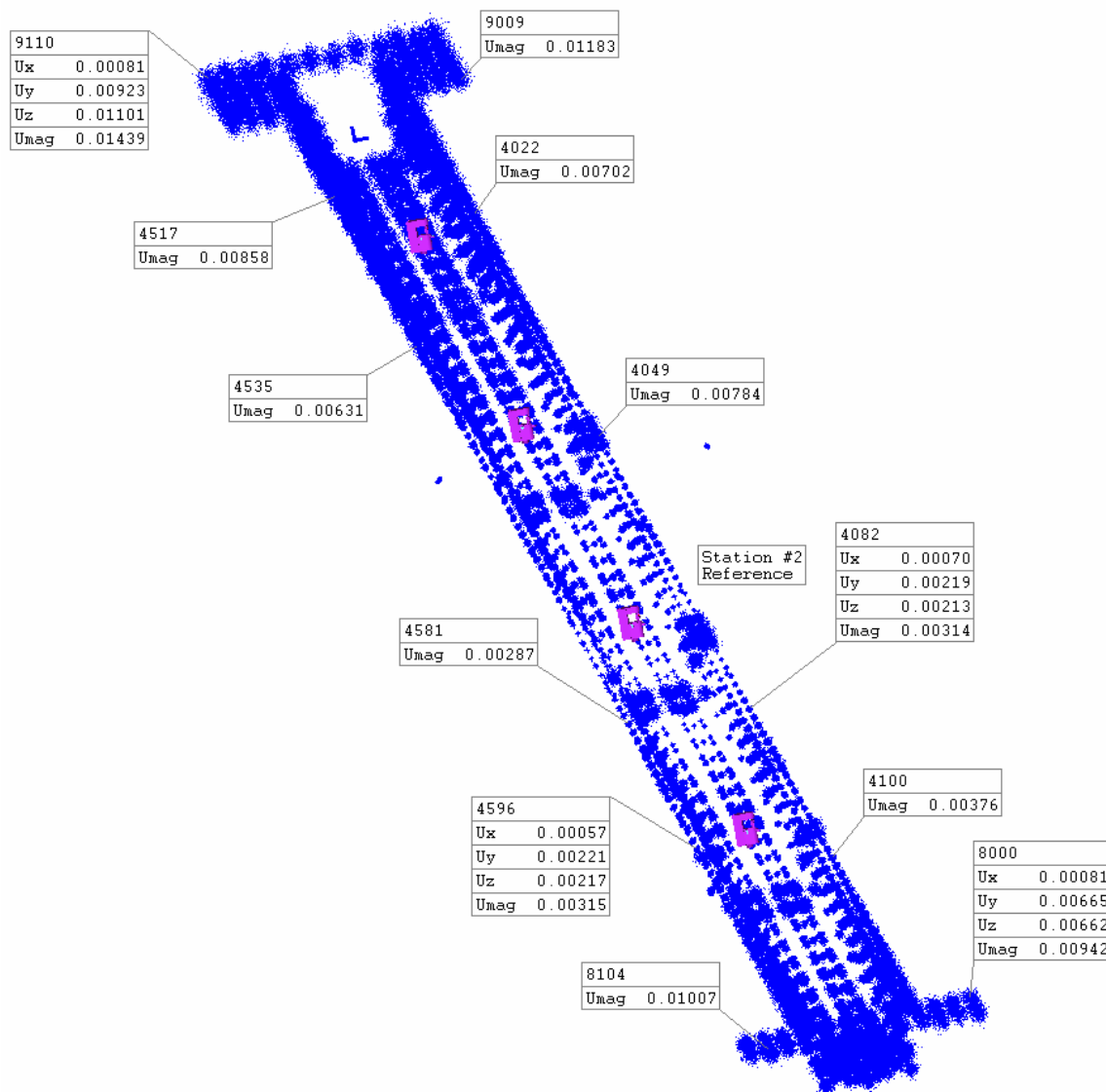
three of the points. There are several items worth noting in the figure. First, as expected, the uncertainty of the coordinate increases as the measurements progress down the chain. In addition, the XYZ components of the point uncertainties follow common notions about laser tracker performance on this type of geometry. The coordinate system for the catapult trough has the X axis along the trough, the Y axis transverse to the trough, and the Z axis vertical up from the carrier deck. The uncertainty for point 9110, for example, is low in the X axis direction relative to its uncertainty in Y and Z. This is due to the measurement geometry. The laser tracker's range measurement component is primarily responsible for the X axis coordinate measurements. Since this is the most accurate component of the tracker, the uncertainties in the X axis are low. The point with the maximum overall uncertainty is point 8104 with a value of 0.0192 inches. As expected, this point is at the end of the measurement chain.



USMN Uncertainty Field Results: Station #0 as Reference

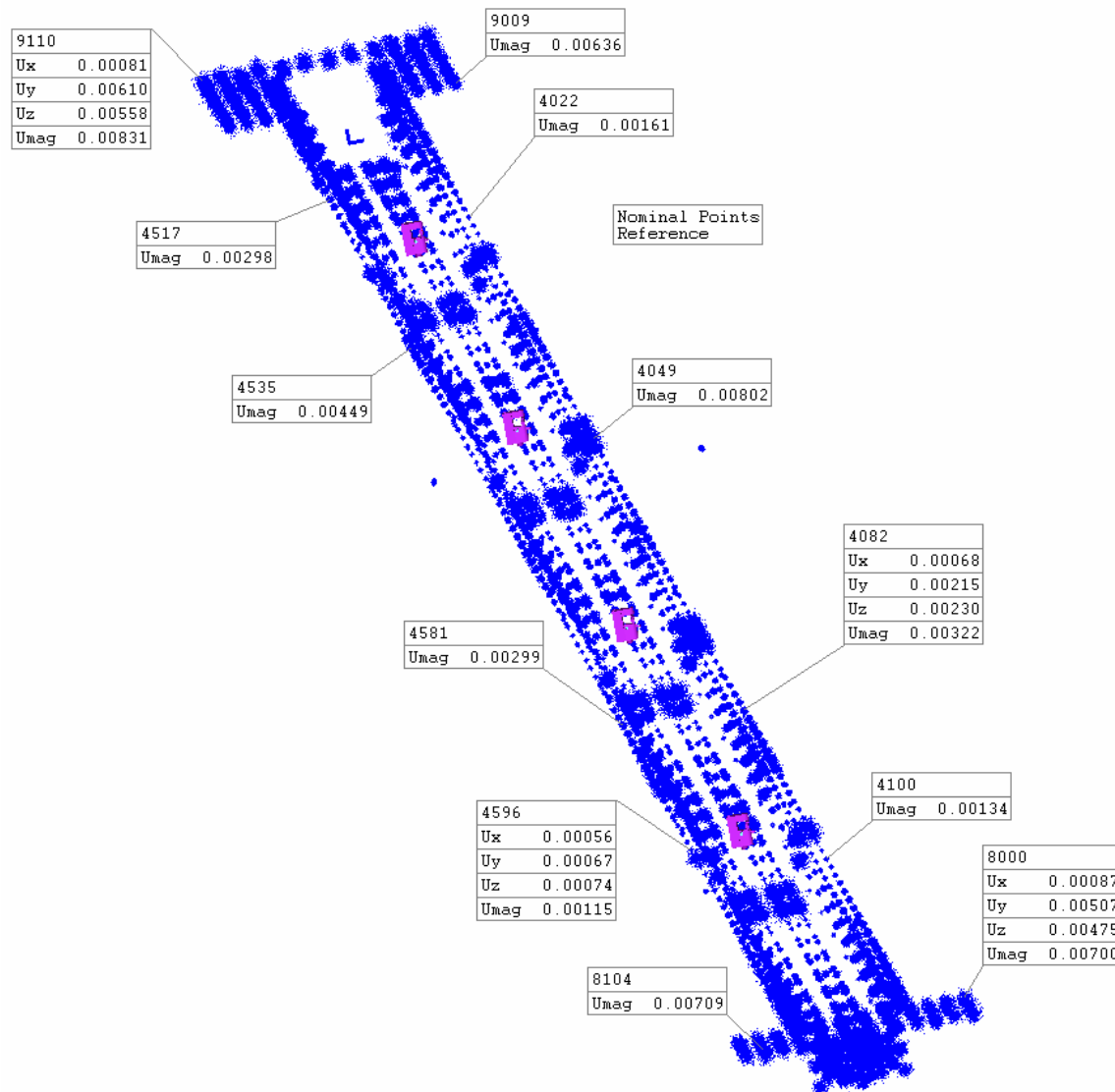
There were a total of 1,797 measured points in this network. Of these points, 60 had measurements from more than one instrument. Of these, only 10 had measurements from more than 2 instruments.

It is important to note that the uncertainty results presented here are based on the first laser tracker station as the reference. This is an arbitrary choice. It means that all the uncertainty values in this analysis are based on the first station remaining fixed in the computation. It would be just as valid to leave another station in the network fixed instead. If, for example, the analysis is repeated with the third station, station 2, fixed the results are based on a different reference. The next figure shows the results of this analysis along with the numerical uncertainty values for selected points. The maximum uncertainty magnitude is 0.0145 inches. As expected, the lowest uncertainties are in closest proximity to the reference instrument.



USMN Uncertainty Analysis: Station #2 as the Reference

In addition to measurement data, the USMN is also capable of incorporating nominal reference data into the network. In this case, there is a set of nominal points along each side of the trough. These points span the entire measurement area. By including the nominal data in the network and designating it as the reference entity, the uncertainty results will yield the uncertainty of the measured points relative to the nominal values. In this case, this will substantially reduce the uncertainty in the measured points because there is effectively a strong ground link that is added to the network and measured by each instrument. To demonstrate this effect, the sample data was analyzed in this manner. The results are shown in the figure below. The maximum uncertainty magnitude is 0.0089 inches.



USMN Uncertainty Analysis: Nominal Data as the Reference

The inclusion of the nominal coordinate data in the network raises several issues. First, if the nominal data is measured by several of the stations, the nominals will have the

same effect as another set of measurements. This behavior may be beneficial in many cases, but it is important to note that if the physical object that is being measured does not match the nominal model reasonably well, the results will not truly represent the measurements from the instruments. Instead, the measurement stations will be adjusted so that they fit the nominals in addition to the common measured points. This will compromise some of the measured points while improving the fit to the nominal data. If the part does not represent the nominals then this is an undesirable result. If, however, only one of the instruments measured points in the nominal data set, the nominals simply provide a different uncertainty reference. They do not impact the instruments in the measurement network. In the catapult measurement example presented in this chapter, this would be analogous to the first instrument being the only one to measure points in the nominal data set.

The measurement system user must decide which entities should be included in the USMN. If nominals are included, the ramifications of this decision must be understood. The USMN does provide for weighting by instrument or by the nominal data set. This means that the nominals could be included, but given a relatively small weight. This would cause them to have less of a negative effect on the measurement sets as they are combined while still providing a measure of the system uncertainty relative to the nominal reference frame.

To conclude, the USMN provides critical information for large complex measurement processes such as aircraft carrier catapult alignment. The knowledge of the uncertainty of the final measurement results is essential to properly understanding the capabilities of the alignment process and the stability of the measurement process. In addition, the ability to go beyond the traditional, sequential best-fit transformation treatment of combined measurement processes allows for a much tighter network solution. By making full use of the redundant, cross-station data, the optimization can more accurately combine the measurement systems. Also, the relative uncertainty component weighting algorithm makes full use of the instrument's measurement components. This is especially important given the scale of the catapult measurement job. Uncertainty weighting is in sharp contrast to a traditional best-fit where all components of a measured coordinate are treated equally in XYZ coordinate space. Future catapult alignment processes can take full advantage of these powerful analysis tools by incorporating as many common network points as possible. The payoff in uncertainty reduction is well worth the additional measurement effort. In addition, the location of additional network points may be analyzed in a measurement simulation using the USMN. This will allow the measurement session designers to determine the optimal placement for additional measurements to maximize their effect on reducing the overall uncertainty.

Disney Concert Hall

The unified spatial metrology network was applied to the Walt Disney concert hall currently being constructed in Los Angeles, California. This concert hall will have an exterior superstructure that will be clad in architectural stainless steel. A model of the structure is shown in the picture of the Concert Hall Scale Model. This model shows the reflective quality of the stainless steel and highlights the need for proper alignment to maintain the visual properties. To achieve this appearance, it is necessary to accurately

adjust the structural supports for the steel sheets during construction using coordinate metrology.



Concert Hall Scale Model

In preparation for the actual construction, a detailed analysis of the measurement process and instrumentation was performed. This analysis included the creation of a unified spatial metrology network to represent the complex interdependencies of the multiple measurement devices and determine the effect of the uncertainties on the overall measurement process.

The goal of the measurement analysis was to determine in advance what type of instruments to use, where to place them, and the shape and size of the uncertainty fields the system would create. The shape of the uncertainty fields was particularly important because it was necessary to hold a tight tolerance on the “in and out” direction normal to the skin surface. This was necessary in order to achieve the smooth appearance specified in the design.

The simulation was performed using uncertainty specifications based on the Leica TDA5005 total station theodolite. This device has angular encoders and a distance-measuring ranger. The corresponding uncertainties used in the simulation were:

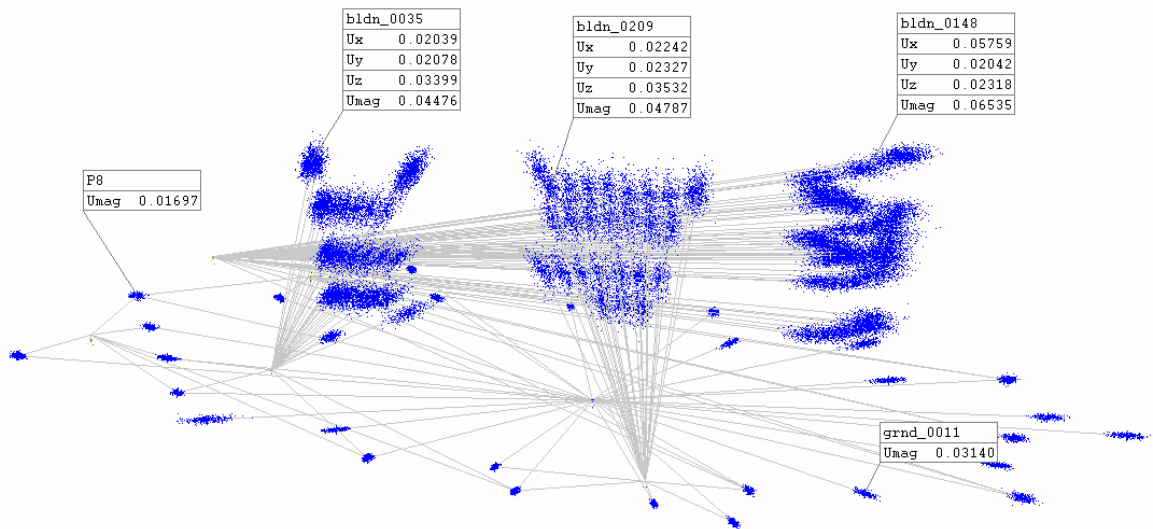
Horizontal and vertical angles = 0.5 arcseconds (1 Sigma)

Distance = 0.039 in. + 2.0 PPM (1 Sigma)

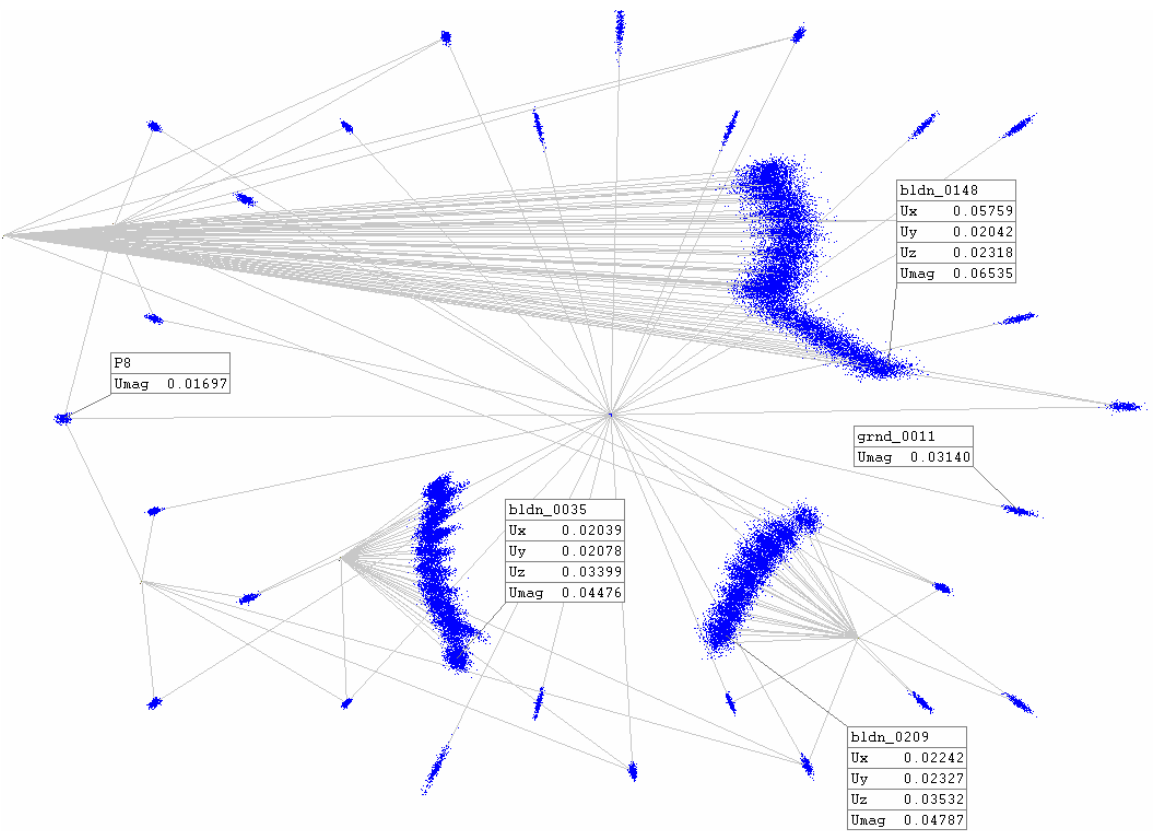
The contractor tasked with erecting the supports for the structure provided a file containing coordinates for some of the points that were to be measured as well as the desired instrument locations based on the physical constraints of the measurement job. There were a total of 7 instrument locations and 285 measured points. Some of these points were not on the structure itself but were instead laid out around the site to provide additional rigidity to the network.

The Unified Spatial Metrology Network was formed and the uncertainty fields for each point were determined. In this case, 300 cloud points were solved for each of the 285

measured points. The coordinate uncertainties resulting from this analysis are shown in the figures below.



Concert Hall Uncertainty Analysis: Oblique View



Concert Hall Uncertainty Analysis: Top View

The uncertainties for the network control points around the base of the structure are relatively low when compared to those of the structural measurements. This is primarily because multiple instruments measured the network points, while the structural points were measured by a single instrument. In addition, notice the difference between the uncertainties in point bldn_0035 and bldn_0148. The latter has a higher uncertainty because it is much farther from the instrument and therefore subject to the parts per million error term of the device. In addition, the instrument measuring bldn_0148 is referenced to the control network differently than the instrument used for the other point. All the uncertainties in this analysis are stated relative to the #0 instrument in the job which in this case is in the center of the network.

The results from this uncertainty analysis were used to validate the proposed measurement process. The measurement contractor has used the measurement geometry to not only verify the structural fabrication, but also to set the components of the structure during assembly as shown in the picture below.



Disney Concert Hall Structural Supports

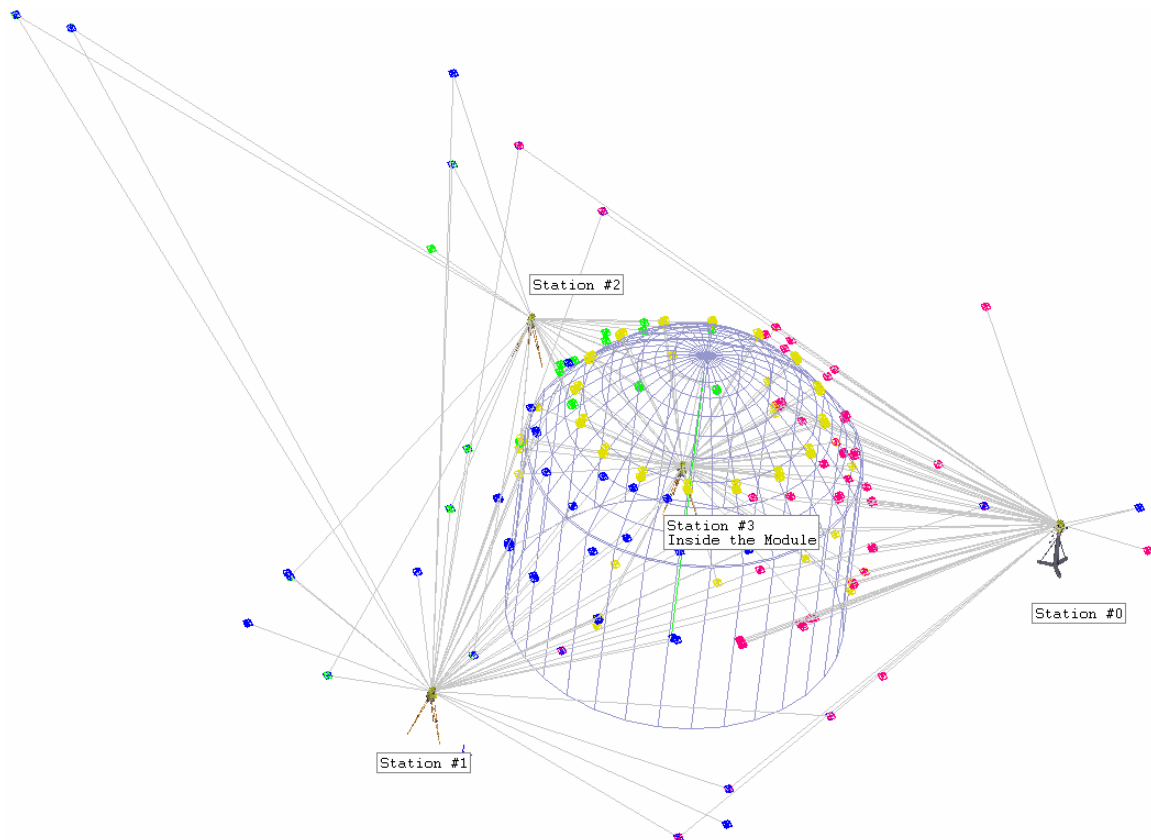
Submarine Section Assembly

Modern submarine fabrication is a modular process with many stages. Currently, General Dynamics Electric Boat and Northrop Grumman Newport News Shipbuilding are teamed on the fabrication of the Virginia SSN 774, an attack submarine. The modular nature of the submarine design allows each shipyard to fabricate individual modules and transport them to the final assembly area where they are mated with the submarine. As each module is assembled and fitted, a series of measurements must be performed to verify geometry and determine the physical interface with the next stage of the process or another module. It is common to measure the as-built characteristics of a module then transfer those measurements to another module at another location. This allows holes to be drilled and interfaces to be machined before the components are ready for final assembly. Each measurement job typically requires multiple instruments due to the size and geometry of the submarine as well as physical constraints in the measurement environment. This section presents two measurement jobs at two different stages in the life of a submarine module.

Hull Fabrication: Vertical Configuration

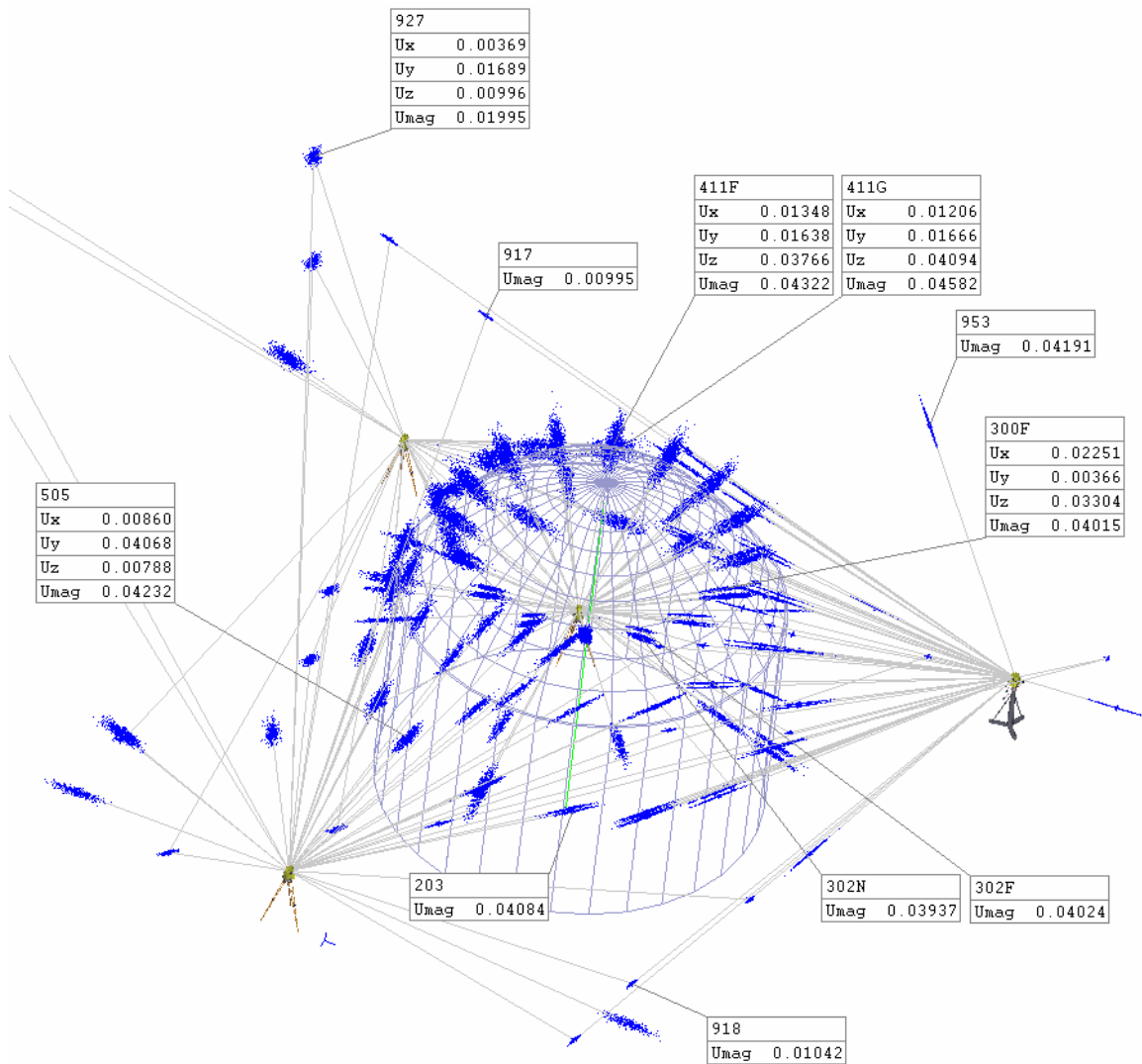
The first stage has the section in the vertical configuration. This is the orientation of the module during the hull assembly process. The sections of the hull are welded one to another and measured at each step in the process. The figure below shows the

measurement of a partially completed hull module. At the time of the measurements, the segment shown on the top of the module was not yet in position. The purpose of the measurement job was to measure the interface between the currently assembled components and the final module hull section before assembly. The job required the use three total station theodolites placed around the outside of the module and one station inside the module. The goal of the measurement job is to combine all of the measurements into a single coordinate set for subsequent analysis. The results of the analysis are then used to determine where to cut the section for proper fit-up with the next component.



Submarine Module Measurement in Vertical Configuration

Using the USMN, it is possible to form the combined coordinate set in a single operation and provide the user with uncertainty statements for all of the measurements considering the combination methods. There were a total of 196 points in the network. The results are shown in the figure below of this analysis with the first instrument, station 0, as the reference.



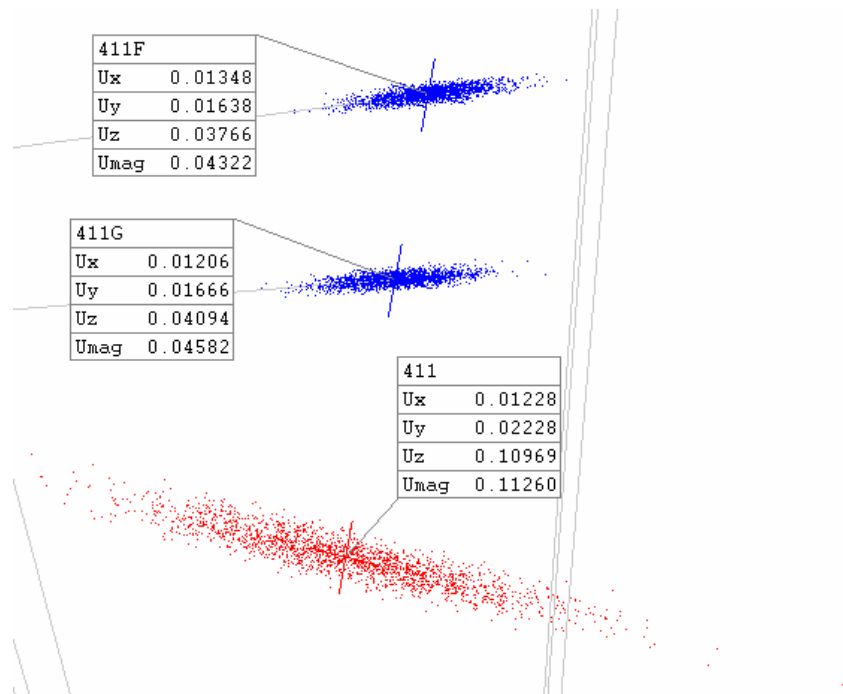
Submarine Module Measurements with USMN Uncertainty Fields

The X coordinate axis is along the axis of the submarine, which is vertical in this case. Notice the large reduction in uncertainty for the points that were measured by more than one total station. Point 917 for example, has an uncertainty of 0.00995 inches with two measurements while many of the points measured with only one instrument have uncertainties in the 0.040 inch range. Also notice that the uncertainty fields for measurements from only instrument 0 are narrower than those for instruments down the chain. This is because their uncertainty fields incorporate not only the instrument uncertainty, but also the uncertainty in the orientation of the multiple measurement systems.

Also notice the 411F and 411G points. They each have an uncertainty of approximately 0.043 inches. These points are named F and G to indicate that they are measurements on a hidden point bar. A hidden point bar is a device used to measure a point which is not visible to the instruments. The points of interest to those assembling the hull are not the 411F and 411G points, but instead point 411 which is the end of the hidden point

bar. By measuring the two points on the bar, the third point is computed by extending along the direction of the vector between the two measurements by a known length. This practice is quite common in coordinate metrology and therefore the uncertainty of this measurement process is of interest to the USMN.

Following the methods used to determine the uncertainties in geometrical fitting operations (Section 0), it is possible to determine the uncertainty field of the end of the hidden point rod. This is done by re-computing the end of the bar for each set of points in the uncertainty fields of the measurements. Performing this analysis on the 411F and 411G points with a 25 inch hidden point bar length yields the results shown in the figure below. These results are based on 2000 uncertainty field points. This number of points is not normally necessary in order to determine the uncertainty, but it is helpful to have a dense distribution when viewing the results graphically at close range.

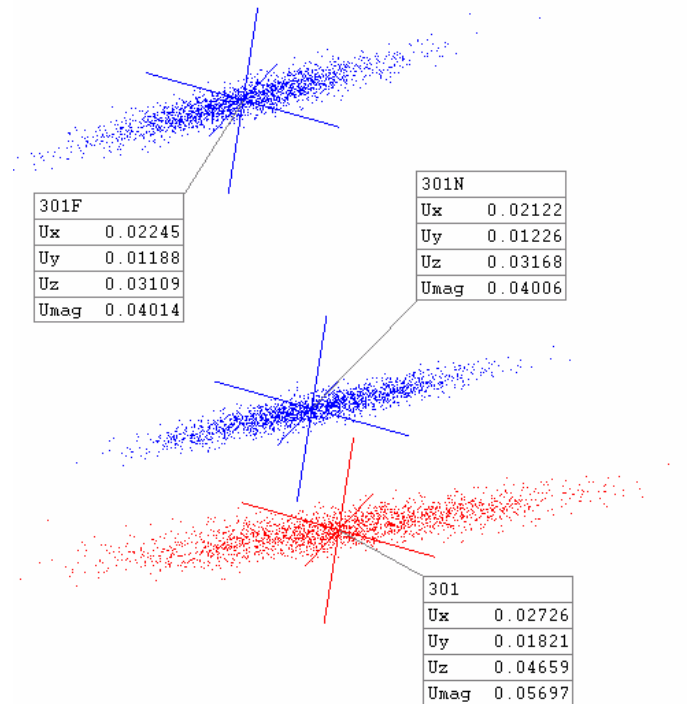


Hidden Point Bar Uncertainty for a 25 inch Bar

Though the total uncertainty in the end point, 0.114 inches, is fairly large given the instrument uncertainties, it is most likely not a hindrance to this particular measurement application. The goal of this measurement is to determine the location along the hull axis where the section should be machined in order to properly mate with the next component. This means the X axis is of the most concern. In this case, the X axis uncertainty of the hidden point is only 0.013 inches, well below the required measurement uncertainty.

The 25 inch point bar measurements were spaced by approximately 10 inches. This means the angular effects of the point uncertainty on the bar direction were magnified by the distance from the last point (411G) to the end point due to the extrapolation of the hidden point. This is one of the primary reasons for the large uncertainty in the end point. To demonstrate the effects of using a shorter hidden point bar (one with less

extrapolation), the analysis was repeated on a different set of points in the same measurement job. This set was spaced by approximately 6 inches, on an 8.5 inch hidden point bar. The results of this analysis are shown in the figure below.



Hidden Point Bar Uncertainty for an 8.5 inch Bar

The uncertainty magnitude of the hidden point in this case is 0.057 inches, much lower than that of a 25 inch bar. The real issue is not only the length of the bar, but the percentage of the bar length that is extrapolated from the measurement direction. Obviously reducing this percentage reduces the hidden point uncertainty. The extrapolation percentage is computed by

$$E_{\%} = \frac{100(\ell - d_{AB})}{\ell}$$

where,

$E_{\%}$ is the extrapolation percentage

ℓ is the length from the first target, A, to the hidden point

d_{AB} is the distance between the measured points on the bar, A and B

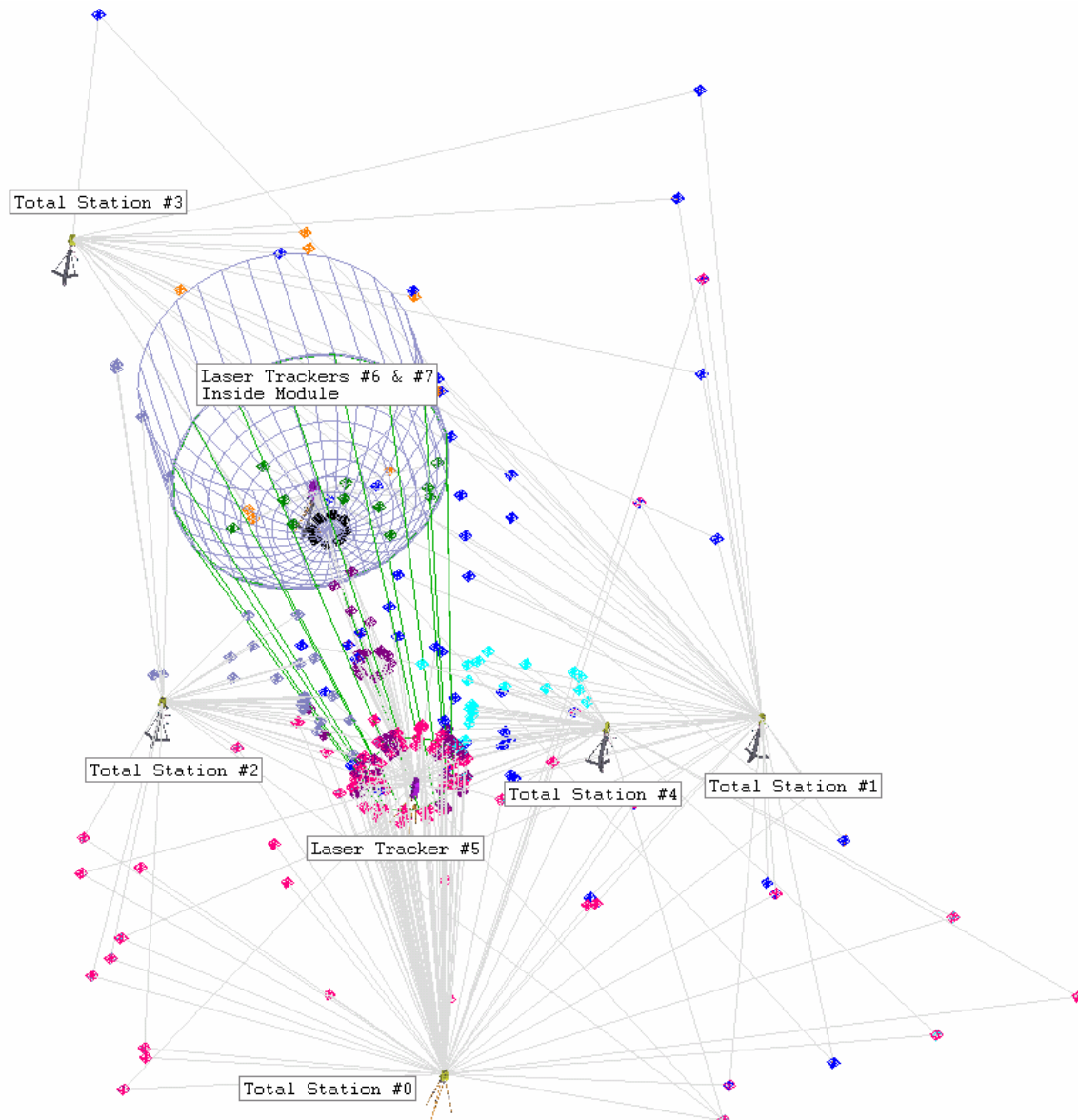
For the 8.5 inch bar, the extrapolation percentage is 29% and for the 25 inch bar, it is 60%. Note that the error for the longer bar is also approximately twice that of the shorter bar. This approach provides a heuristic method that may be used to determine the

proper hidden point bar for a measurement process. Ideally, a simulation of the measurement job beforehand would highlight potential uncertainty problems.

As mentioned previously, since the X component is of primary concern, the large uncertainty in the hidden point analysis does not adversely affect the manufacturing process. If, however, the measurement technician had oriented the instruments differently relative to the object, the uncertainty of the X direction could degrade substantially. In addition, the orientation of the hidden point bar relative to the uncertainty fields of the coordinates may also have a drastic effect on the results. These possibilities underscore the need for the USMN uncertainty field information in industrial measurement processes. The hidden point computation component of the metrology application is augmented to create uncertainty fields for the end-point of the bar based on the uncertainty fields of the measured targets. This allows users to quickly check their hidden point measurement geometry and understand the uncertainty of the measurements.

Module Alignment: Horizontal Configuration

Once the shell of the module is welded together and the internal supports are in place, the module is flipped into its horizontal orientation. More outfitting is then performed, and eventually the module is completed to the point where it is ready to be joined to another section. This particular module is the aft end of the submarine. This means the drive system components must be aligned with the interface on the module during the assembly process. The measurements required for this process cover the outside and inside of the module and the fixturing aft of the module where the drive systems will attach. The measurement job geometry is shown below.



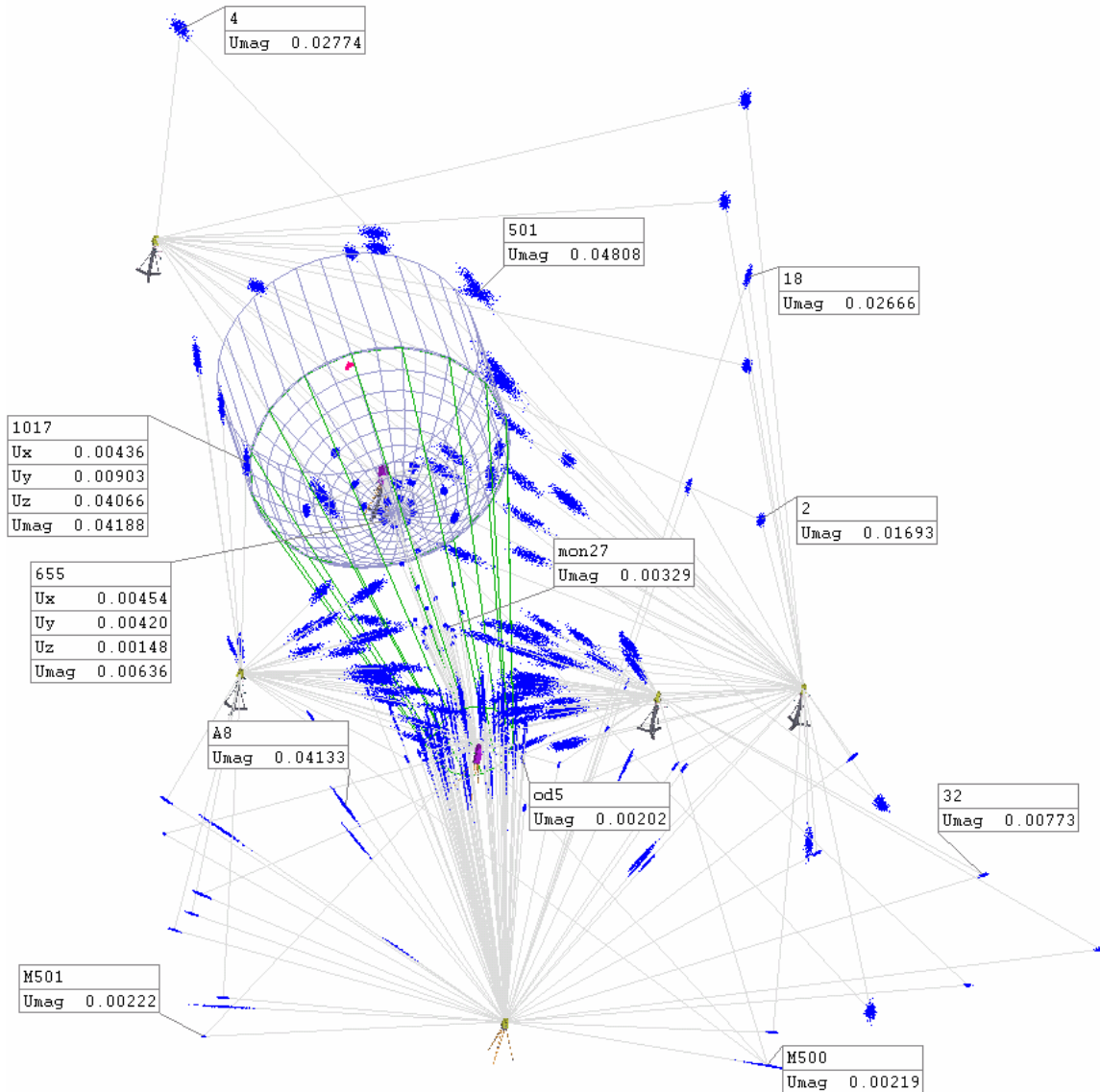
Submarine Module Measurement Job: Horizontal Configuration

This job highlights an important aspect of the USMN, the combination of measurements from different types of instruments. In this case, there are five total station theodolites used in conjunction with three laser trackers. Two laser tracker stations inside the submarine were necessary because with the first instrument setup, several of the points were not visible. This was not discovered until many measurements had been taken. Instead of discarding the previous measurements, another station was added to the network and located relative to common monuments. The reason for the mixture of total stations and laser trackers was the need for high accuracy on the measurements of the drive system mounting interface. Using a laser tracker for these measurements results in a much lower uncertainty value. Laser trackers are not, however, as well suited for the exterior measurements on the module. This is due to physical constraints and the

realities of measurement within a production environment. It would be difficult to carry the laser tracker target (a spherically mounted retro-reflector), all over the module without occluding the interferometer beam. When these occlusions happen, it is necessary to reset the interferometer by returning the reflector to the calibrated tracker nest or to another more easily accessible fixed known point in the workspace. Total stations, however, can simply point at reflective tape targets already placed on the module and measure directly.

In addition to the practical usage concerns for the different devices, there are uncertainty differences. The laser tracker can measure a range value to nominally 0.0006 inches, while a total station measures range to nominally 0.040 inches. The angular measurement uncertainties of the two devices also differ. The total stations typically outperform the laser tracker on angular measurements by a factor of two. In order to properly combine these instruments in a common measurement job, the relative uncertainties must be considered. The USMN handles this using the relative uncertainty weighting scheme presented in Section 0. The examples presented there detail the application of the uncertainty values to the weights in the point computation solution.

The uncertainty field computation for 300 points the results are presented below.



Horizontal Configuration Measurement Uncertainty Results

Total station #0 was used as the fixed reference in this case, so all uncertainties are reported relative to that reference. As stated previously, this is an arbitrary choice and the analysis can easily be repeated using another station or set of nominal values as the reference. There are several point uncertainties of interest in this analysis. First, notice the large improvement in uncertainty when a point is measured with two total stations as opposed to just one. This is made apparent by comparing point A8 with point 2. The uncertainty of A8 is based on the measurements of only station #0 and therefore is of the magnitude of typical total station uncertainty, 0.041 inches. Point 2, however, was measured by both station #1 and station #3. Its uncertainty is much lower, 0.017 inches. This is due to the relative uncertainty weighting in the point computation. The angular components of the total station measurements are weighted more than the distance components since the distance is less certain. As a result, the redundant data is used to determine the optimal point while taking advantage of the components of the

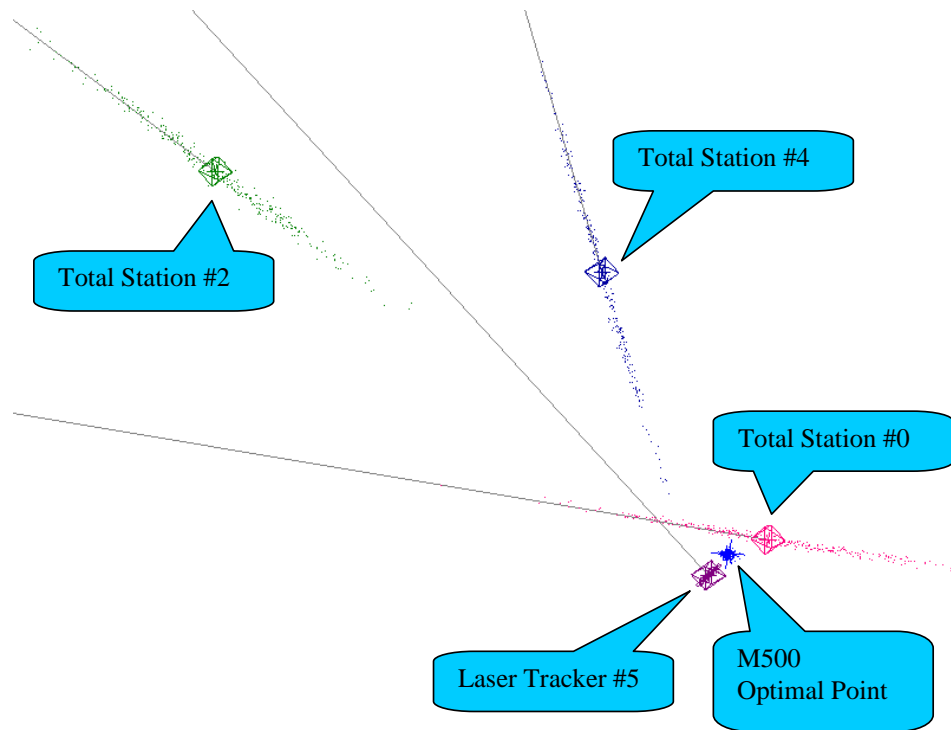
measurement that are more certain. A similar effect is observed with point 18. Its uncertainty is larger than that of point 2 due to the geometry of the measurements. The apex angle (the angle between the measurements) is much smaller, so the uncertainty field is larger along the range components of the measurements. Point 2 did not exhibit this behavior because the measurement geometry was such that the angular components of the measurements were able to sufficiently override the range measurements.

The laser tracker measurements in this job also show a reduction in uncertainty when multiple instruments measure a point. Point 655, for example, is measured by the laser tracker inside the module. Its overall uncertainty is 0.006 inches. This is actually larger than expected for a laser tracker measurement. It does, however, seem reasonable since the laser tracker is not the reference for the uncertainty statements. This means the uncertainty of point 655 is not only attributed to the instrument's uncertainty, but the uncertainty with which the instrument is located relative to the network reference. In this case, the reference is total station #0. The uncertainty for point mon27 (monument # 27) is 0.003 inches. This is almost half the uncertainty of point 655. The reason for this reduction is that mon27 is measured by 2 laser trackers, one inside the module and one outside. As with point 655, the uncertainty also includes the uncertainty in the total network relative to the reference.

Point M501 is a good example of a mixed measurement. It is measured by total station #0 and laser tracker #5. Its uncertainty is 0.002 which is quite low considering the other values in the network. Based on the geometry of the measurements, the reason for this low uncertainty is clear. The laser tracker's angular measurement is used to reduce the uncertainty of the ranging component of the total station.

Point M500 is also interesting since it is measured by three of the total stations and one of the tracker stations. The figure below shows a close-in graphical image of point M500 and the four individual measurements used to compute the optimal point. In addition, the combined uncertainty field for the optimal point is shown relative to the uncertainties in the other measurements. Note, however, that the uncertainties of the individual measurements shown in this figure are not the USMN uncertainties including the instrument location uncertainty. Instead, the individual measurement uncertainties in the figure only represent the uncertainty in each measurement assuming the instrument location is fixed. The resulting uncertainty field for M500 does, however represent the total uncertainty of the point relative to the network reference, since it was computed as part of the USMN solution process.

The effects of the relative uncertainty weighting scheme are apparent in the figure below. Notice that the optimal point matches the length of the laser tracker measurement closely. It also corresponds well with the angular measurements of the total stations. It does not, however, closely match the range measurements of the total stations, since the range component has a relatively high uncertainty. The numerical error results and the corresponding weighting factors are shown in the table below.



Point Computation with Mixed Instrumentation

	Horizontal Error	Vertical Error	Distance Error	Magnitude
Total Station #0	-0.0017	0.0042	0.0032	0.0056
Weights	11.2	11.2	0.24	
Total Station #2	-0.0052	0.0036	-0.0784	0.0786
Weights	5.0	5.0	0.24	
Total Station #4	-0.0008	-0.0088	-0.0403	0.0413
Weights	8.2	8.2	0.24	
Laser Tracker #5	0.0072	-0.0094	0.0005	0.0118
Weights	2.9	2.9	16.4	

Point Errors and Relative Uncertainty Weights for Point M500 table

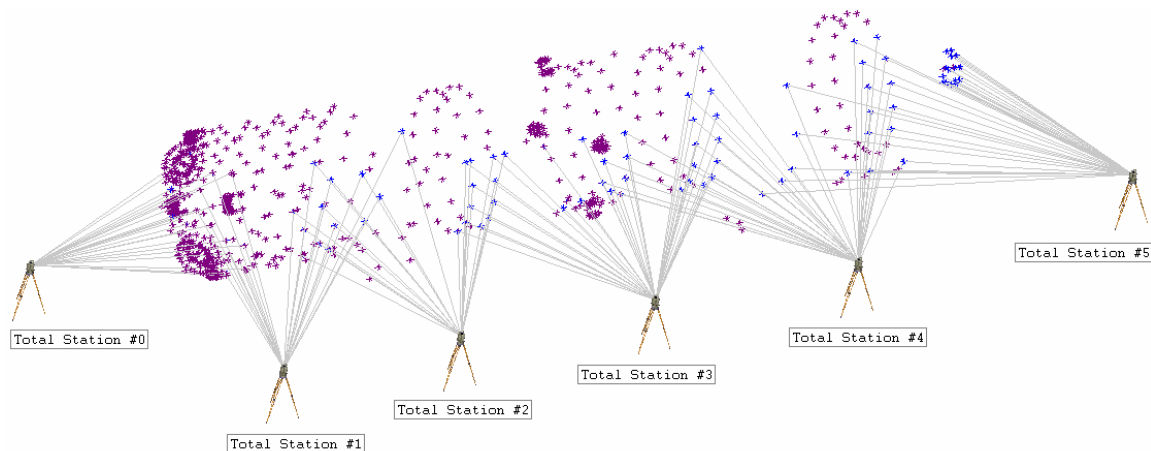
To summarize, the complexities of the measurement job presented in this section highlight the need for the USMN and its treatment of combined measurement systems. The manifestation of the instrument uncertainties in the final resulting measurements must be quantified in order to present the measurement results with confidence. This is especially true when changes in measurement geometry and instrument choice can have large effects on the uncertainties in the measurement results.

Nuclear Power Plant Steam Generator Replacement

The process of steam generator replacement was mentioned briefly in a previous section. The description highlighted the many steps in the replacement process where coordinate metrology plays a pivotal role. From determining the as-built geometry of the generators to guiding a robotic machining and then welding tool, measurement processes are critical to the success of the entire process.

One specific measurement process is presented here to demonstrate the effects of chaining instruments together along a large object. In this case, the actual measured coordinate values on the generator are used in a simulation process to determine the effects of the uncertainty. The measurement values from the actual survey were not used because they are not available. This is because many individual measurement software applications were used in various staged of the analysis process. Because of this, there is no reference back to the actual readings of the measurement instruments. This is another issue in the coordinate metrology industry. Often, as in this case, the final combined coordinate data set is provided without a traceable link back to the raw measurements of the instruments. This traceability problem is addressed by metrology applications that store not only the measured points, but the raw observations behind them. Without that level of data integrity, the original measurement data can be lost. Not only does this eliminate traceability, it also makes it impossible to completely re-analyze the data without making assumptions about the original process.

The measurement data obtained in this analysis was used to determine the location of the steam generator components relative to the reactor hot and cold pipes at the base of the generator. To accomplish this measurement on an object 3 stories tall and 12 feet in diameter, multiple measurement instruments were required. In this case, a series of 6 total station theodolites were used. The measurement geometry is shown in the figure below. Station #0 is shown on the left measuring the hot and cold reactor nozzles and station #5 on the right is measuring the main steam output nozzle.



Steam Generator Nozzle Measurement Geometry

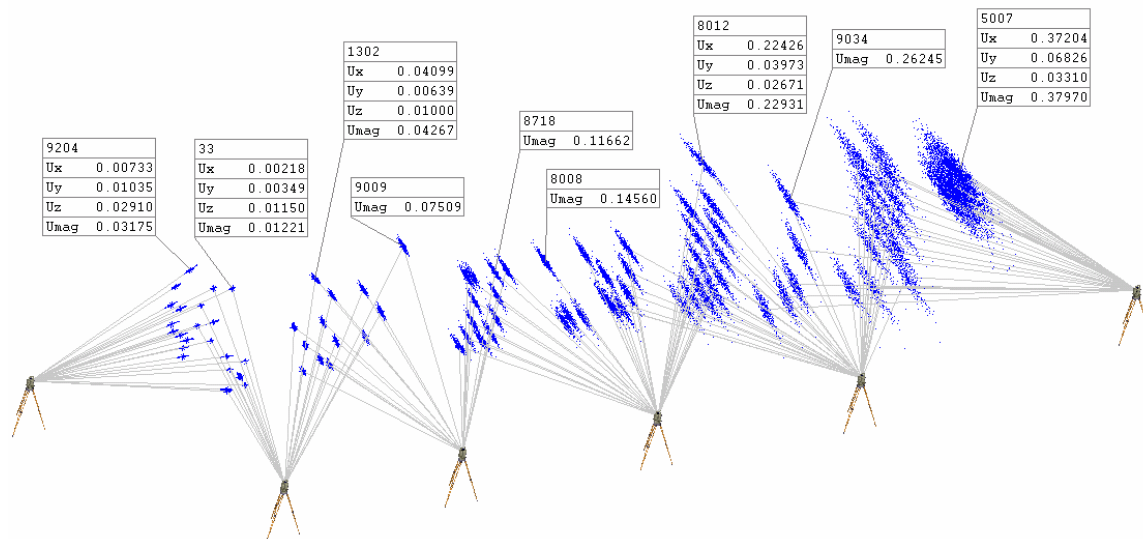
The particular instrument used in this measurement process was the Sokkia Net-2B. This instrument has the following uncertainty characteristics:

Horizontal and Vertical Angle: 2.0 arcseconds (1 sigma)

Distance: 0.0313 inches + 1.0 parts per million (1 sigma)

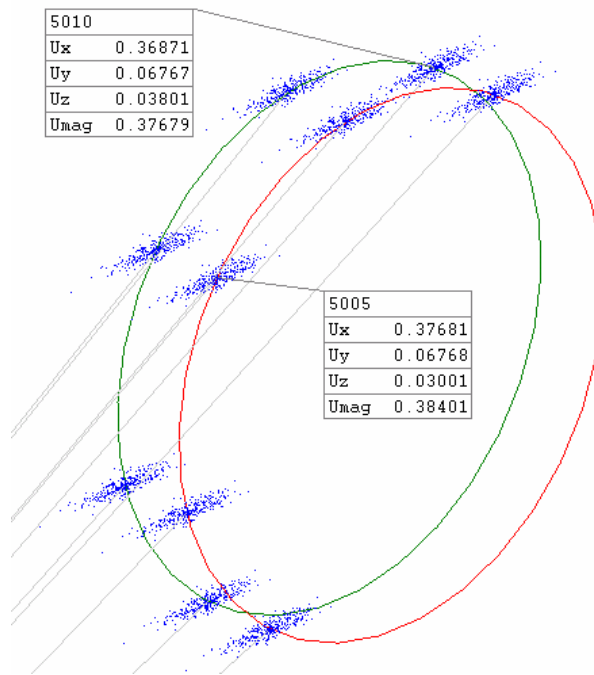
Though the instrument often exceeds these performance specifications, the manufacturer's published uncertainty values were used in this analysis since they will provide a conservative uncertainty statement.

The unified spatial metrology network contained a total of 106 points and the 300 point uncertainty field analysis for each point. The resulting uncertainties are shown in the figure below. All of the uncertainties are referenced relative to station #0, as it remained fixed during the process. Notice the increase in uncertainty as the measurements progress along the generator. At the lower end, the uncertainties are approximately 0.032 inches, but at the final station, the uncertainties are ten times greater at 0.380 inches.



Steam Generator Measurement Uncertainty Fields

One goal of this measurement job was to describe the steam nozzle relative to the base of the generator. In this case, the steam nozzle is at the end of the measurement chain. A closer view of those uncertainties is provided in the following figure.



Steam Nozzle Measurement Uncertainty Fields

The analysis procedure used on the measurements was to determine the best-fit circle for both sets of points on the steam nozzle. These were used to determine an axis direction as well as to topmost extent of the steam nozzle material. In order to determine the uncertainty of the circle fit operations, the geometry fitting uncertainty analysis methods described in Section 0 were applied to the measurements. The results of this analysis are presented in Steam Nozzle Circle Fit Uncertainty table.

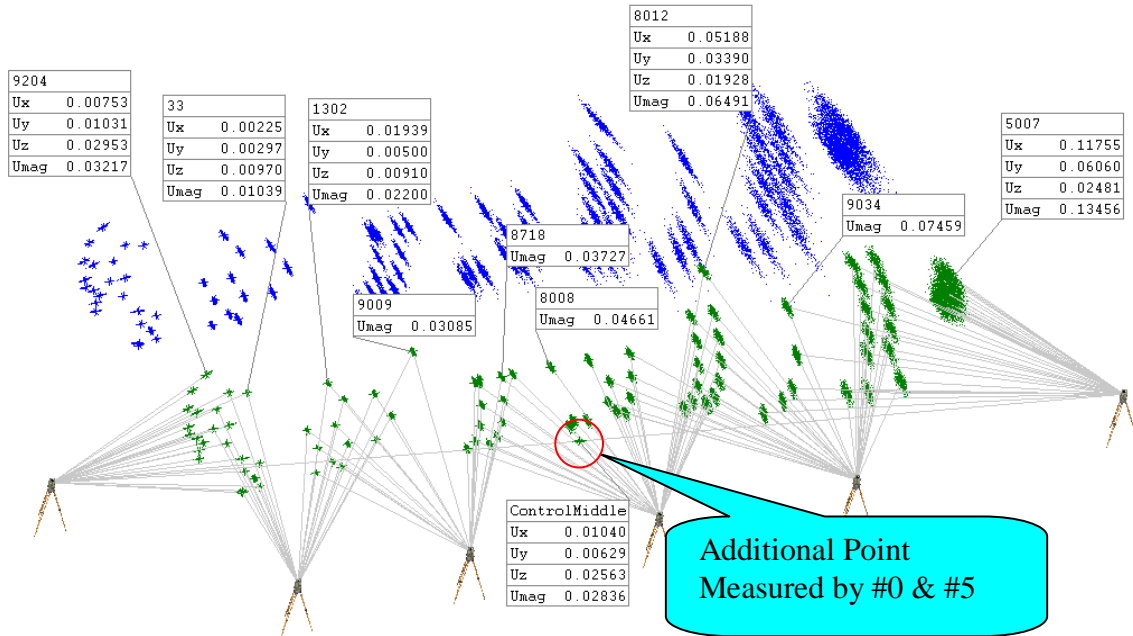
Component	Uncertainty	
Center Ux	0.372	
Center Uy	0.068	
Center Uz	0.032	
Center Umag	0.379	
U Diameter	0.023	
U Normal Vector	0.066	(degrees)

Steam Nozzle Circle Fit Uncertainty

These uncertainty values are significantly larger than the measurement technicians expected. In fact, had this analysis technology been available at the time of the measurement job, another approach would most likely have been used. This would include re-thinking the network geometry and possibly using another, more accurate instrument. Suppose a change in the measurement geometry were proposed to reduce the uncertainty. Using the USMN, this change can easily be tested so that the effect on the measurement uncertainty is known before the new measurements are acquired.

The following discussion will detail a simple modification to the geometry that significantly reduces the uncertainty in the measurements.

To demonstrate the uncertainty reduction from even one point closing the measurement loop, an additional point was added to the measurement job with simulated measurements from both station #0 and station #5. The point called “ControlMiddle”, is near the middle of the generator above station #3. The figure below shows the results of the uncertainty analysis with the additional point. The previous uncertainty fields are also shown off to the side to provide a visual reference.

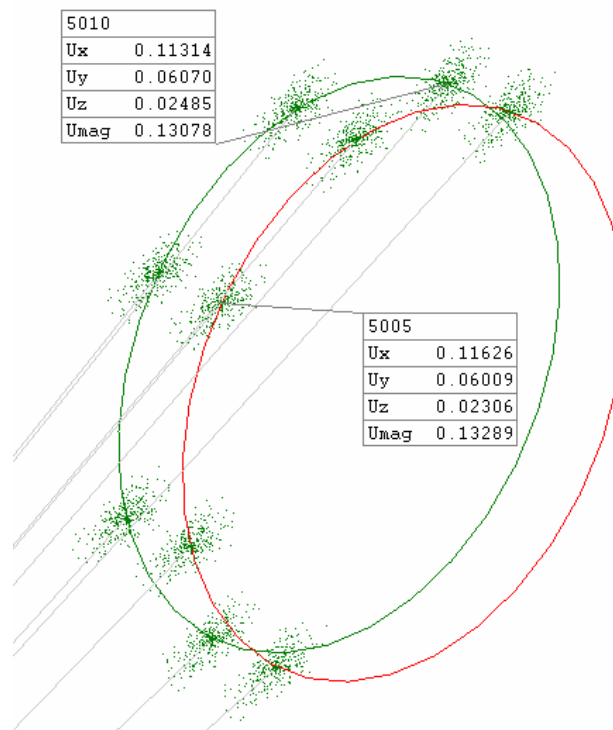


Uncertainty Reduction with One Point Closing the Loop

The addition of a point closing the measurement loop significantly reduced the measurement uncertainty. This was especially true at the end of the measurement chain. Point 5007, for example, had an original uncertainty of 0.38 inches. After adding the common point for the first and last stations, the uncertainty was reduced to 0.13 inches. The uncertainty reductions for several points are listed in the Uncertainty Change Due to Additional Point table. Also, refer to the figure below the table for a view of the revised uncertainty fields at the steam nozzle.

Point	Uncertainty: Original Setup	Uncertainty w/ Additional Point	Change	% Change
9204	0.032	0.032	0.000	1%
33	0.012	0.010	-0.002	-15%
1302	0.043	0.022	-0.021	-48%
9009	0.075	0.031	-0.044	-59%
8718	0.117	0.037	-0.079	-68%
8008	0.146	0.047	-0.099	-68%
8012	0.229	0.065	-0.164	-72%
9034	0.262	0.075	-0.188	-72%
5007	0.380	0.135	-0.245	-65%

Uncertainty Change Due to Additional Point



Steam Nozzle Uncertainty Fields With Additional Point in Network

In addition to the uncertainty reduction in the individual points, it is worth investigating the effects on the results of the circle fit on the steam nozzle. This analysis was repeated with the new uncertainty fields. The results and a comparison to the previous values are shown in Steam Nozzle Circle Fit Uncertainty Reduction table.

Component	Uncertainty	Uncertainty	Change
		w/ Additional Point	
Center Ux	0.372	0.115	-0.257
Center Uy	0.068	0.062	-0.007
Center Uz	0.032	0.023	-0.009
Center Umag	0.379	0.133	-0.247
U Diameter	0.023	0.022	-0.001
U Normal Vector (deg.)	0.066	0.069	0.003

Steam Nozzle Circle Fit Uncertainty Reduction

To summarize, the initial steam generator measurement geometry had a large uncertainty at the end of the measurement chain. By adding a single additional point to close the measurement loop, the uncertainty was reduced significantly. This is an excellent example of the benefits of measurement simulation, planning, and combined uncertainty analysis.

USMN Review

The USMN presents a solution technology to the combined coordinate uncertainty problem. It provides methods for quantifying the uncertainty, questions about the effects of individual instrument uncertainty on complex networks are answered.

Coordinate Uncertainty Field Representation

The uncertainty fields computed with the USMN represents the geometrical and statistical information for points and instruments. The results are relevant to the current working frame transformation. Also, due to their discrete nature, it is possible to apply these fields to computed geometry i.e., a best-fit cylinder, sphere, hidden/projected points.

Instrument Uncertainty Characterization

Manufacturer's typically state uncertainty results aimed at making their instrument appear the most appealing when compared to other competing devices. The shop-floor technicians who actually use the instruments in real production environments, however, seldom experience the ideal test conditions of a manufacturer's laboratory.

To bridge that gap, the USMN can be used to determine the uncertainty in the components of an instrument's measurements. This method may be applied under any set of measurement conditions in order to determine the real-world instrument performance. These values are then used as inputs to the Unified Spatial Metrology Network in order to determine, through simulation, the effects of the instrument uncertainty on the measurements of points and objects.

Measurement System Combination Methods

Measurement system combination is based on the uncertainty of each measurement relative to the other measurements in the network. The point computation methods in the USMN allow the network to determine the optimal combination of the measurements based on their respective uncertainties. Not only does this provide a more realistic

measurement result, but it also provides a better fit between data sets from different instruments by taking full advantage of the instrument characteristics as opposed to the current practice of matching up XYZ values.

The development of the Unified Spatial Metrology Network extends the simultaneous measurement system combination concept to include the combination of the uncertainties in the individual measurements. The result is a series of uncertainty fields based on the combined effects of all the instruments and nominal data sets in the network.

Task Specific Measurement Uncertainty

The application of coordinate uncertainty fields to other analysis processes is also handled by the USMN. Geometric object fitting, for example, in down-stream analysis processes are often applied to measurement results to obtain useful information. It is not enough to know the uncertainty of all the points measured on the surface of a cylinder. Instead, the user needs to know the resulting uncertainty in the best-fit cylinder.

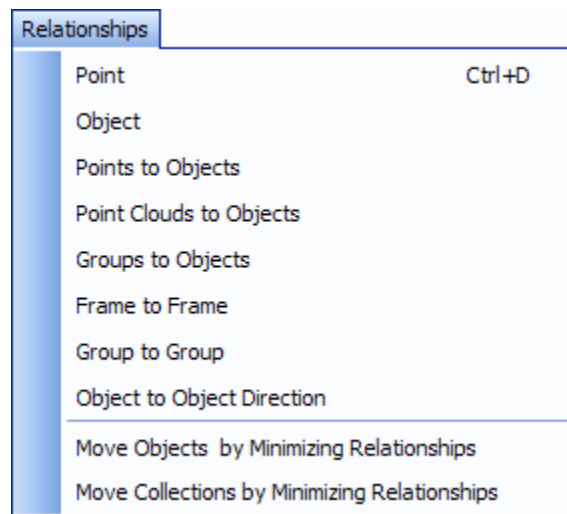
Summary

The USMN is a practical computational tool for quantifying the uncertainty fields in coupled spatial metrology systems. Measurement system combination must include uncertainty combination. The Unified Spatial Metrology Network provides this capability. The case studies in this manual also show the need for graphical metrology software to portray this information to the end user. Uncertainty specifications and estimates are often confusing to untrained operators. With some training and experience, operators can readily interpret uncertainty cloud depictions.

Perhaps the most important result of the USMN is that it is now possible to obtain realistic uncertainties for coupled measurement systems. In practice, when operators are questioned about the uncertainty of certain measurement configurations, they quote uncertainties that are on par with the manufacturer's specifications for the instrument. This is, of course, not realistic. Offering a realistic picture of the measurement uncertainty will, in the long run, allow operators to utilize the metrology equipment to its fullest potential. Being able to visualize the effects of varying a measurement configuration will facilitate greater resourcefulness and creativity on behalf of measurement system users.

RELATIONSHIPS

This chapter is focused at introducing Relationships. Relationships can be used for automatically updating dynamic changes between objects, make creating vector groups for analysis faster, provide a quick overview of the status of build operations and enable you to optimize your measurement directly against design constraints. By optimizing relationships they can provide a special type of spatial transformation capability. By simultaneously optimizing (i.e., best-fit) relationships i.e., the distance between (measured) points and known geometric shapes (e.g., points, lines, planes, and surfaces) the process fits measurements directly against their nominal features. The goal of the optimization is to find the spatial orientation of the measured points such that the relationship or group of relationships (i.e., distance between the points and their geometric shape) is minimized. The relationships of that can be used for fitting include:



Relationship Menu

Introduction to Relationships

Coordinate measurements made on parts and tools need to be converted from one three-dimensional coordinate system (e.g., one based on the instrument) into relevant airplane coordinate systems. While SpatialAnalyzer provides a number of means to accomplish this task (e.g., best fitting pts-pts, pts to surfaces optimization, and a wide variety of frame construction processes), the optimization of coordinate measurements of airplane parts and tools can require refinement of the orientation to meet special functional airplane manufacturing and design requirements.

The trailing edge of an airplane wing is a good example of the need for simultaneous optimization of point to geometric shape relationships for spatial transformations. To efficiently manufacture the best airplane wings we need to optimize the position of the fittings that hold the trailing edge control surfaces relative to one another, and relative to the wing as a whole. Point measurements made of the control surface hinge line fittings have tight requirements normal to their design hinge line, while their position along the line is not as critical. Measurements made on the spar's web surface (the primary structure that holds the hinge fittings) have tight requirements normal to the design

surface, but are not generally constrained on their position on the web surface. Measurements of the hole locations in the spar's web that are used to rivet the hinge line fitting to the spar have known positions on the surface, while their distance normal to surface may not be critical because the holes go all the way through the web structure; meaning the hole location on the surface is important not the depth of the hole.

In this case, at least three point(s) to geometry relationships should be created. The group of hinge line measurements should be constrained to their design hinge lines in the first relationship. A second relationship should capture the distance between the web surface measurements and the web's design surface. The third relationship should hold the distance between the measurements of the holes to each hole's design centerline. Typically, control surface hinge line design requirements are tighter than web surface or rivet hole location tolerances. Therefore the hinge line relationships should be considered more important and given a higher weight relative to the other relationships.

The best orientation of the measurements of the real part to the design geometry can be found by simultaneously optimizing the measured points against the constraints defined by the relationships. The optimization will yield the best spatial transformation between the measured point's current position and the new position that minimizes the relationships and helps manufacturing produce the best wing.

Relationship Optimization Example

An example job where we build up some simple design geometry and measurements To present a functional understanding of relationships in SA will start an empty SA job, add several geometric objects, points, an instrument, take some measurements, and then solve for the orientation of the instrument relative to the objects in the model with Relationships. We will explore the effects of balancing the relative weights of the relationships and examine how the weights allow you find the optimum orientation of measurements to the nominal model of the part or tool.

Next, we will add a second instrument and measure some points on the object from a different location. Once the points are measured, we will orient it into the model, thus bringing both instruments into a common coordinate system without common points.

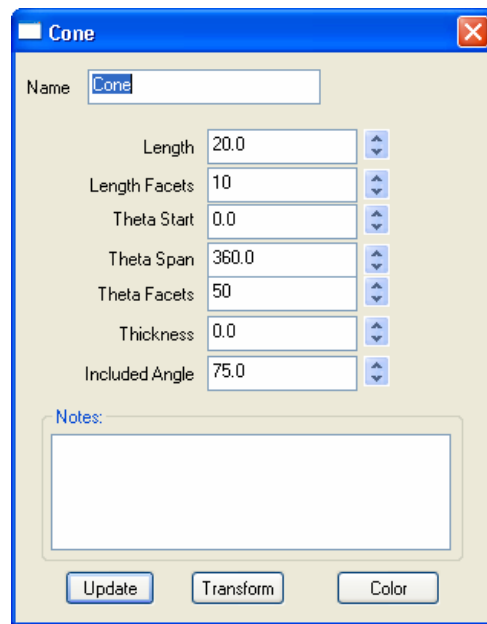
Add Simple Design Geometry

- Run SA and start with an empty file.

Add a Cone

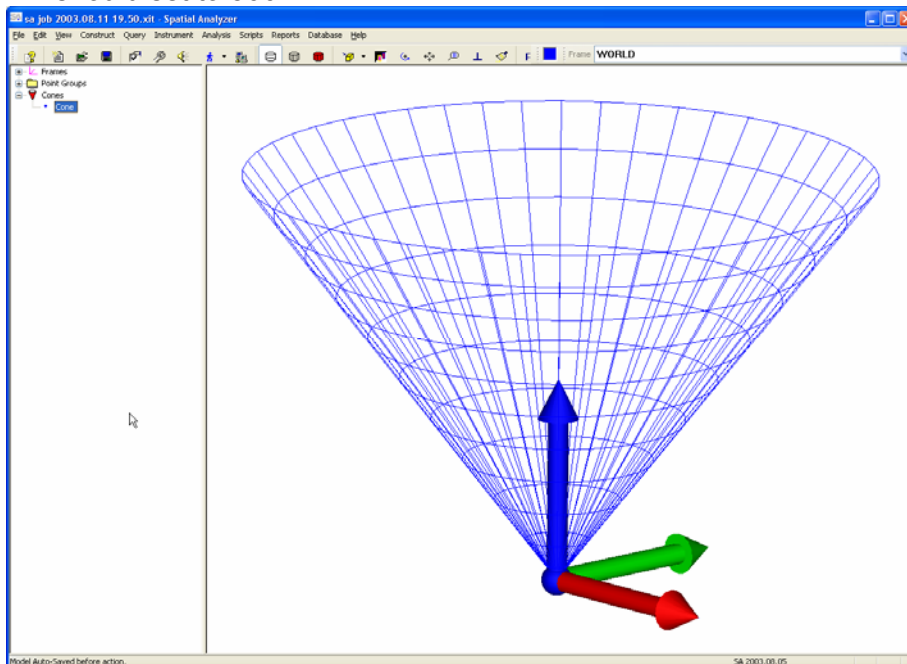
The first element that we add as a design constraint is a cone.

- From the main menu, select
- Change the Cone's attributes to match those shown in the dialog below.



Cone Properties Dialog

- Note the length unit is set at 20 inches or if your units are set to millimeters then it should set to 500 mm.

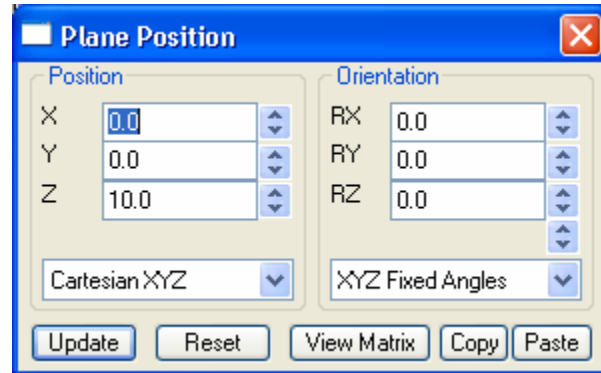


SA with a cone

Add a Plane

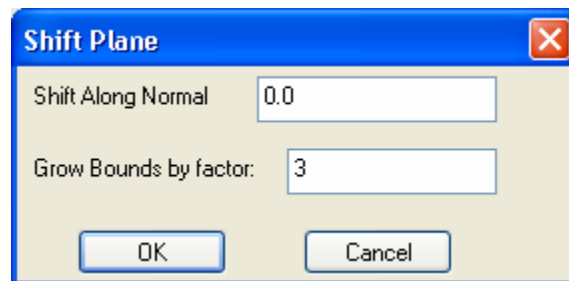
The second element to add is a plane. Change the working color to something different.

- From the main menu, select
After creating the plane a properties dialog appears. Modify the default plane properties by selecting the Transform button.
- Change the Z-component to 10 inches [254 mm] and then select the Update button as shown below.




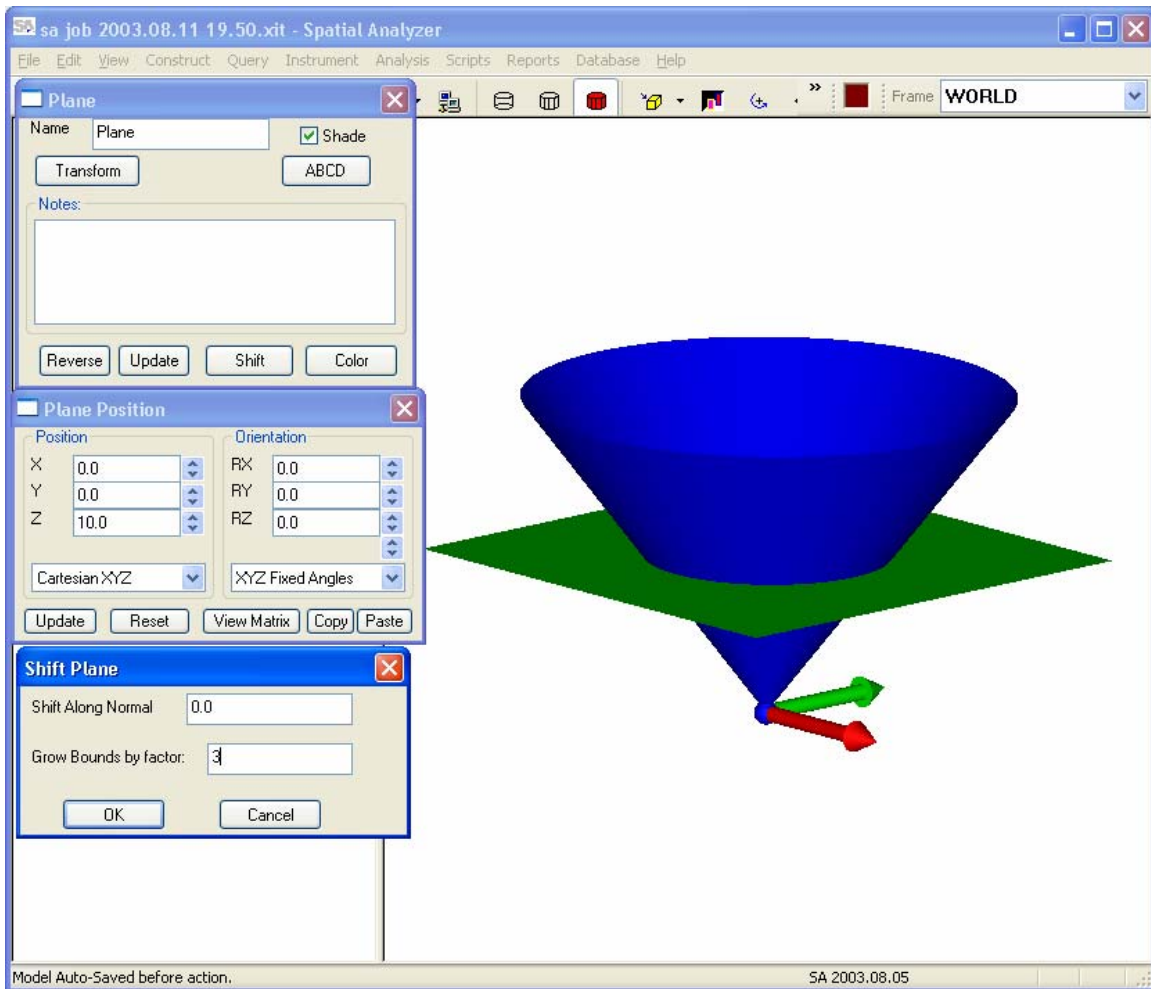
Plane Position Adjustment Dialog

- Close the Plane Position dialog.
Expand the graphical image of the plane by selecting the Shift button on the planes properties dialog. The plane does not have numerical bounds however its image in SA can be changed to help present a more representative model. Grow the planes graphical bounds by 3 using the Shift Plane dialog as shown below.



Shift Plane and Grow by Bounds Dialog

- Select the OK button.
Set the plane shading property to by selecting the Shade check box on the properties dialog.
- Shade the SA model by selecting the  button on the toolbar.
SA should appear as shown below.

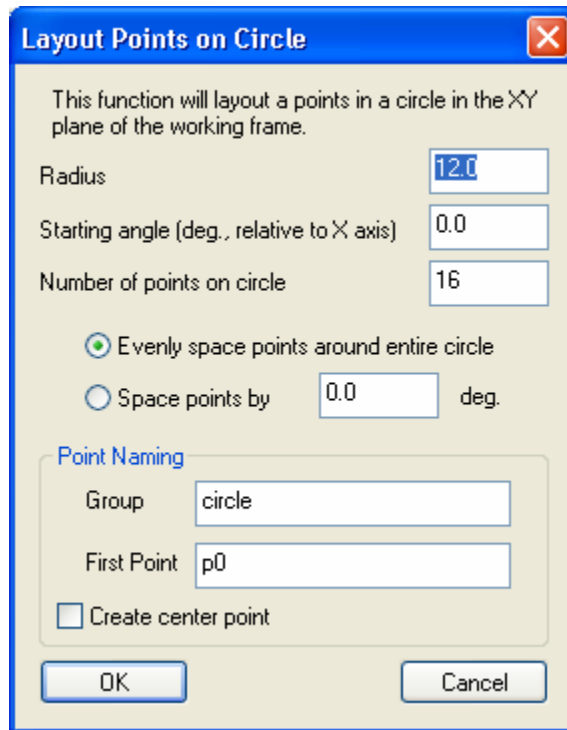


SA with Cone and Plane Design Geometry

Add a Circle of Points

The next element to add is a circle of point. Change the working color to a third color.

- From the main menu, select
- On the Layout Points on Circle dialog set the Radius of the point group to 12 inches [305 mm].

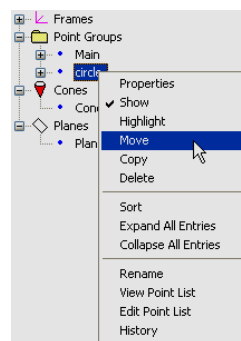


Layout Points on a Circle

- Select the OK button.

Move the 'circle' point group along the working frames Z-axis. There are several techniques to accomplish a move of this type. A relatively quick method starts by opening the Point Group branch in the database tree view. With the Point Group branch open:

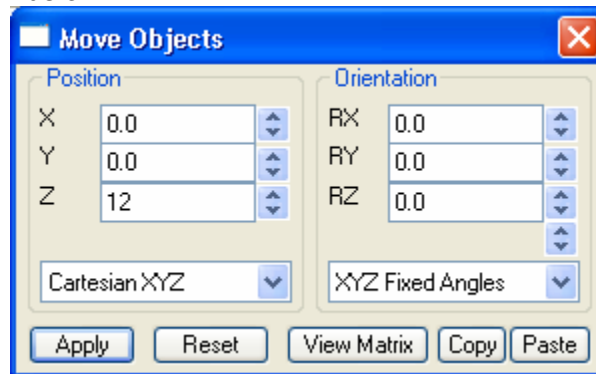
- Right-click on the 'circle' point group which brings up a menu of options. Select the Move menu choice.



Move function on object's right-click menu

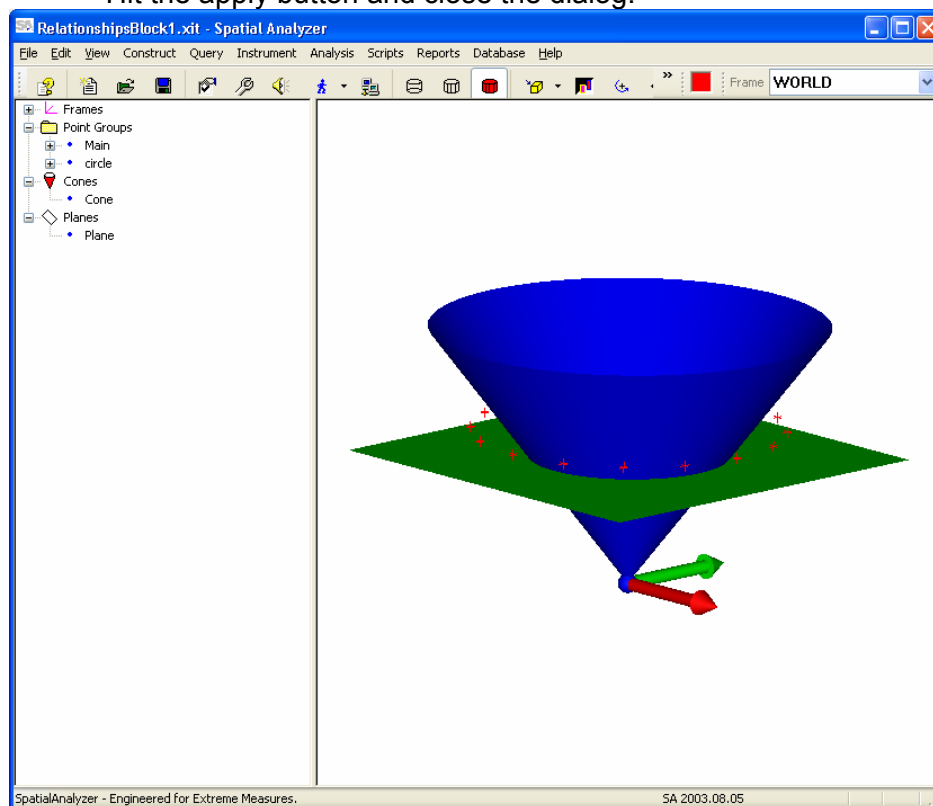
- The Transformation dialog appears which allows you to control the position and orientation of the object.
- Shift the group of point in the Z-direction by 12 inches [305 mm].

- Change the Z-component to 12 inches [254 mm] and then select the Update button as shown below.



Move Objects Dialog

- Hit the apply button and close the dialog.



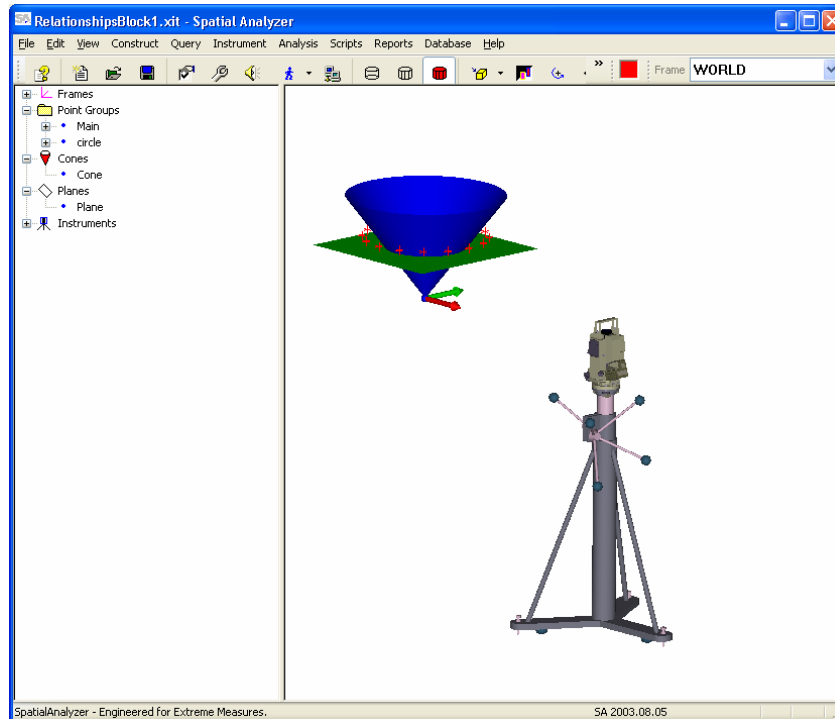
Cone and Plane

Add an Instrument + Fabricate Measurements

To make this a more interesting example we need to add an instrument to the job. By simulating measurements with noise to the design points the process should more closely model a real job.

Add an Instrument

Add the instrument of your choice using the menu option .



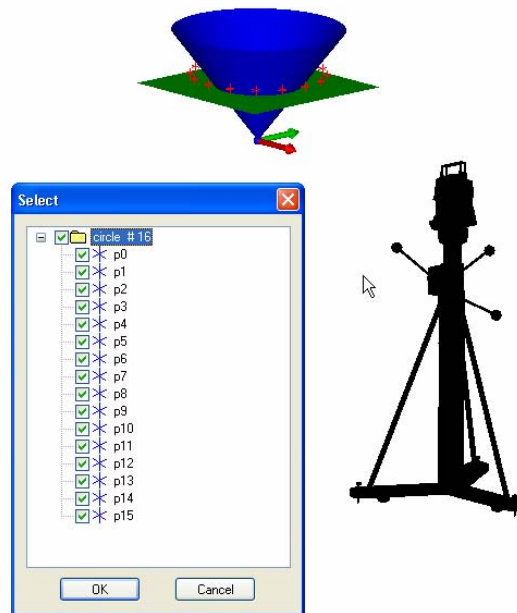
Instrument + Cone and Plane in SA

The instrument is placed 50 inches [1270 mm] down the X-axis by default.

Simulate Measurements with Noise

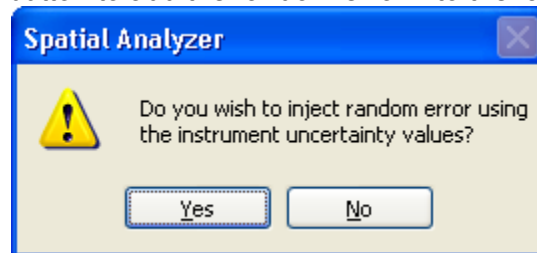
Simulating measurements to the design points is accomplished with options under the Analysis menu.

- Fabricate measurements to the design points using the menu option.
- Select the instrument either with the F2 key or select it graphically by double-clicking on it and then hit the enter button to accept.
- Select all of the points in the 'circle' group either graphically or with the F2 selection dialog as shown below.




Select the points to simulate measurements

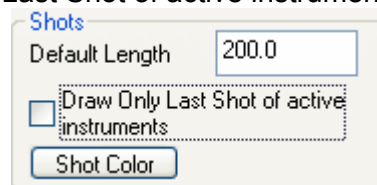
- Hit the enter key when you have selected all of the 'circle' points.
- Select the 'Yes' button to add the random error into the fabricated measurements.



Simulate measurement error

To visualize the measurement ray between the instrument and points we need to change a parameter on the Display tab of User Options dialog.

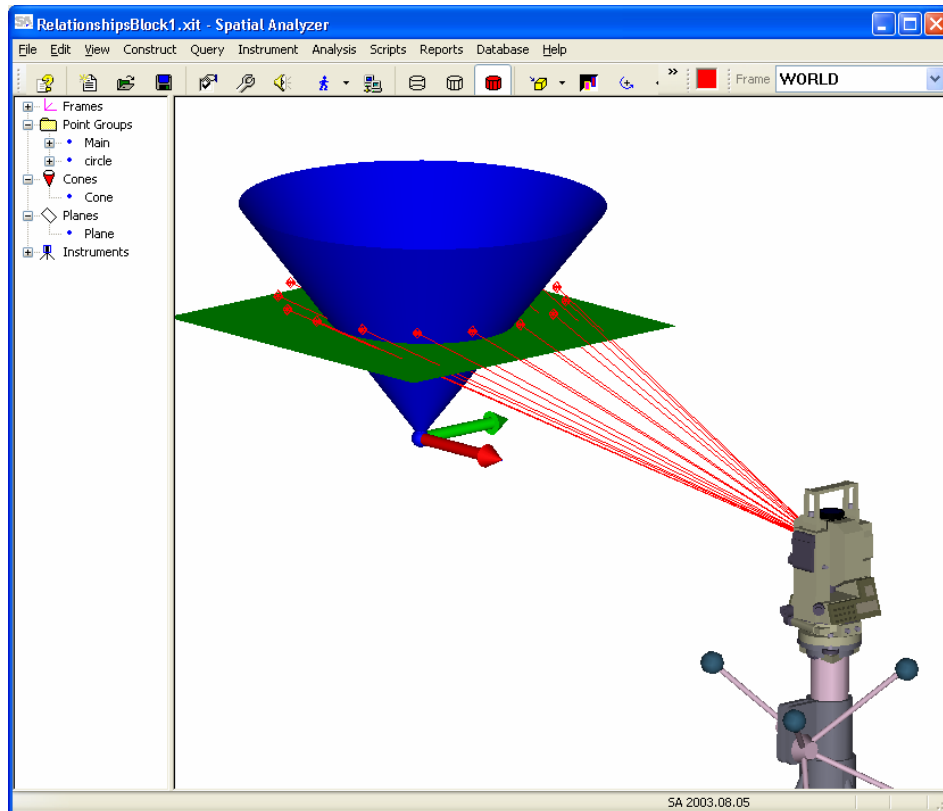
- Either select the  toolbar button or use the menu option to open the dialog.
- Find the 'Shots' area on the Display tab of the User Options dialog.
- Turn off the 'Draw Only Last Shot of active instruments' check.



Turn on Shot rays

The default shot color is gray. To change the shot color for all instruments hit the 'Shot Color' button.

- Select a different color either from the palette of basic choices or define a custom color with the controls. Hit OK on the Color dialog when you done.
- Hit OK on the User Options dialog.



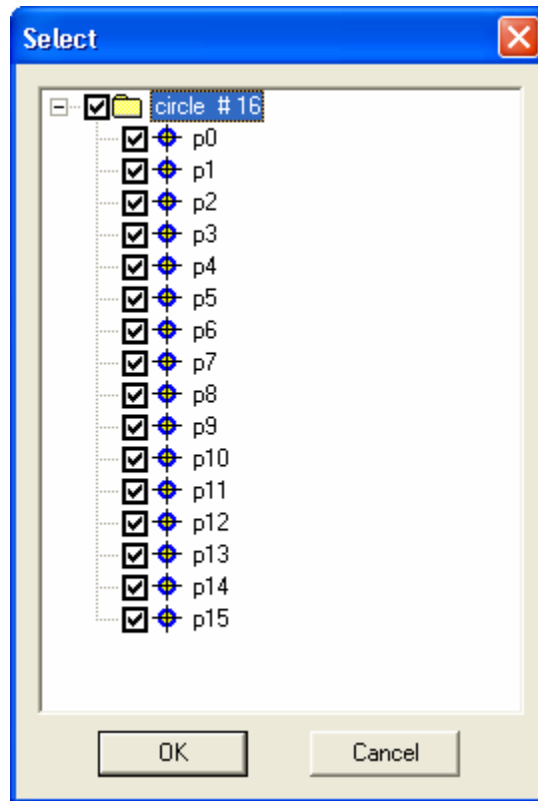
Fabricated measurements with noise

Making Relationships

SpatialAnalyzer's relationships can be used in a number of different ways. They automatically update showing dynamic changes, make creating vector groups for analysis faster, provide a quick overview of the status of build operations and enable you to optimize your measurement directly against design constraints.

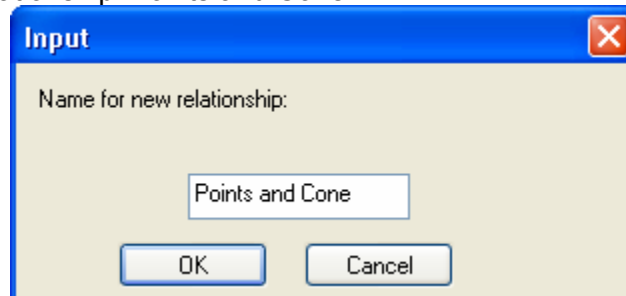
Make Points to Object Relationships

- Select the menu option
- Hit the F2 button to select the points with select dialog. Open the 'circle' group and select all of the points. Selecting the check box next to the group name automatically selects all of the items below it.



Select points for Relationship

- Hit the OK button on the Select dialog.
- Hit the enter key to move to the next step in creating this Relationship.
- Select the Cone when asked for Objects.
- Hit the enter key when done.
- Name the Relationship 'Points and Cone'

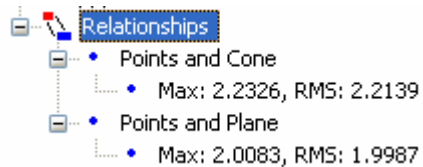


Name the relationship

- Repeat the process for the same points and the plane; except name the relationship 'Points and Plane'

Once the relationships are created open the Relationships database tree view branch. There should be two relationships defined for the current job. The relationships

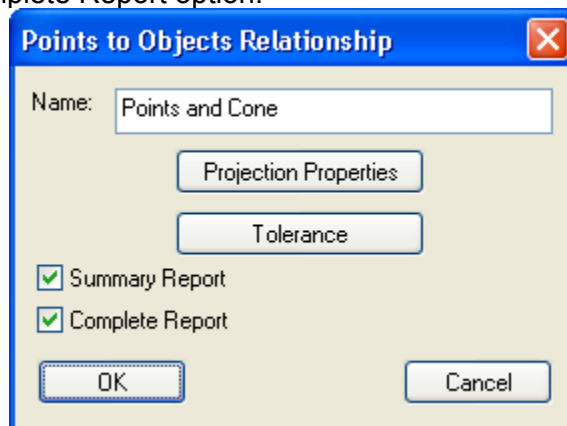
automatically compute and report the Max and RMS statistics between the all of the points in the relationship and the objects. The statistics in your job should be slightly different. The difference is due to the Random Error we introduced with the Fabricated Measurements.



Relationship Values on Tree view

The default reporting behavior for relationships only shows the Max and RMS statistics. Let's change one of the relationships to show a complete report.

- Select the 'Points and Cone' relationship on the tree view.
- Right-Click on the 'Points and Cone' relationship and choose the option on the menu.
- Turn on the Complete Report option.

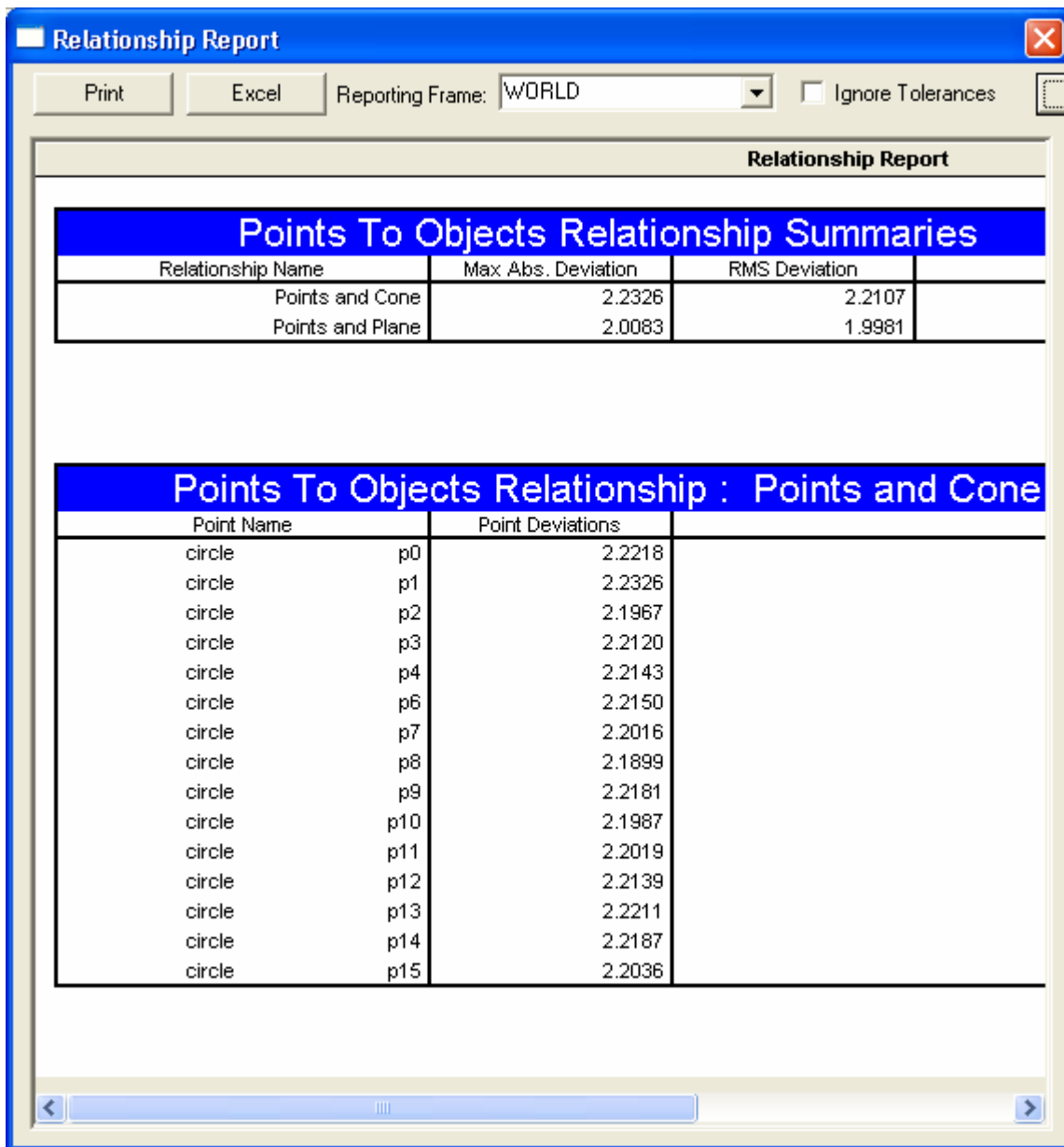


Properties dialog for Points to Objects Relationships

The Projection Properties dialog allows you to control a number of options on how the relationship evaluates the difference between the objects. The tolerance options allow you to configure the threshold of both plus and minus differences that the relationship maintains.

To view the change we made to the 'Points to Cone' relationship

- Select the main Relationship item on the tree view.
- Right-Click on it and choose the option on the menu.




Relationship Report

- Close the Relationship report by hitting the 'Done' button.

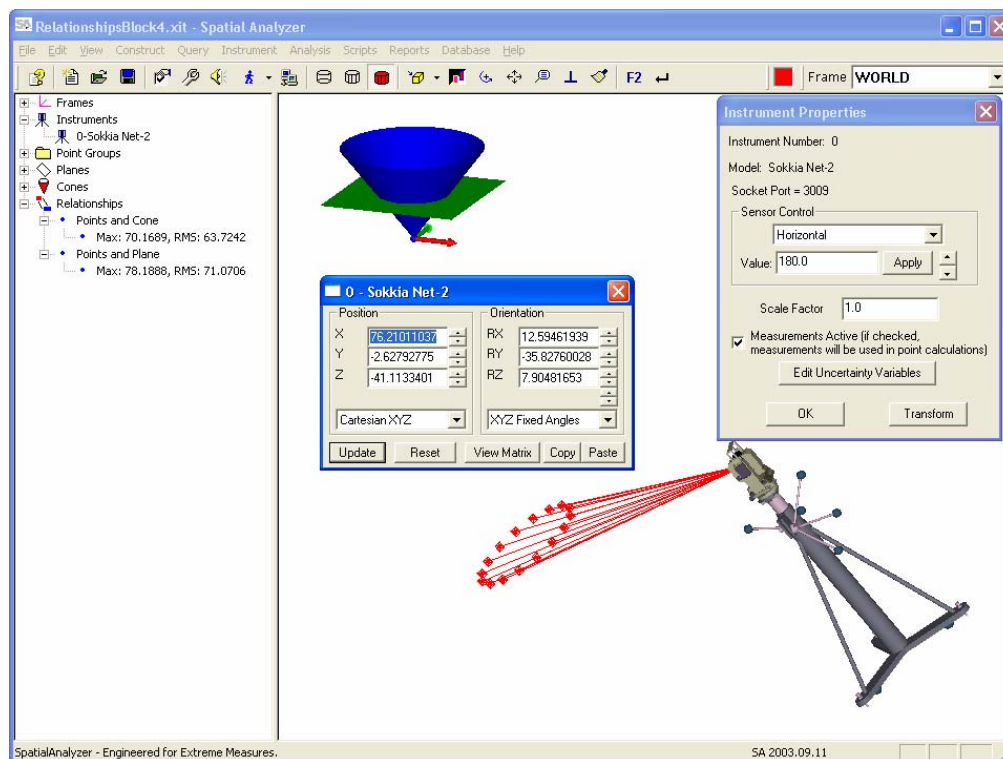
Move the Instrument and the measured points follow

Now that we have setup the design geometry, added an instrument with measurements to some nominal points and created the relationships we need to move the initial setup out to simulate a real job. Typically the instrument and it points are not close to the nominal design geometry. To model that aspect we will drag the instrument out away from it current location.

Dragging an instrument

- Set the view to use the Oblique perspective with the  view toolbar button.
- Select the main menu option
- With the left mouse button down move the instrument laterally relative to the view perspective. The movement of the instrument is amplified to help you move it quickly. To stop the motion at a particular location just let up on the left mouse button when it get to the desired position.
- Rotate the instrument slightly by clicking and holding the right mouse button. Keep the rotations under 90 degrees to keep your results consistent to the results found later in this manual. Rotations greater than 90 degrees will work but the optimization engine will find a solution that will be likely be 180 degrees from the one in this manual.

The figure below shows the instrument properties dialog and its transformation parameters. If you would like a reference to follow the figure has specific values for the instrument transform. Right-click on the instrument's name in the tree view and then select the menu option to set its specific location and orientation.



Shift Plane and Grow by Bounds Dialog

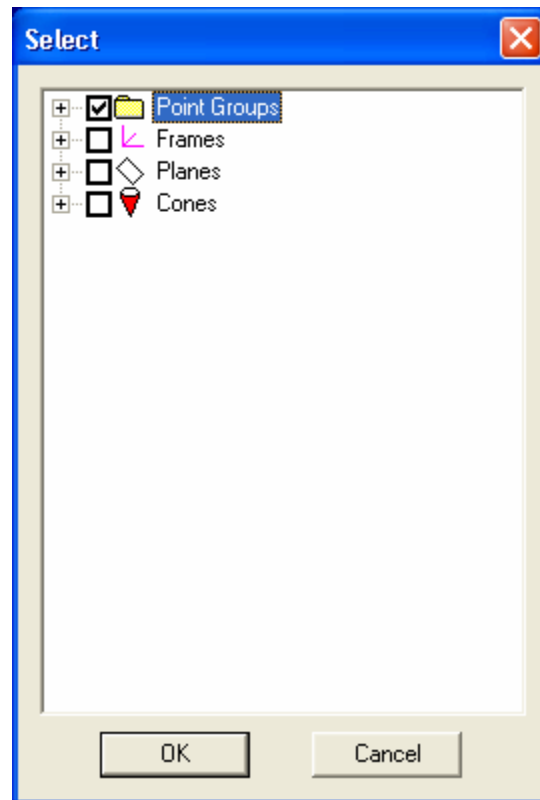
As the instrument moves in the graphics view each measurement ray moves with it and as a result the measured points. The points with their measurements turned off did not move with the instrument. You should also note that the relationship statistics in the tree view are also updated in real-time as the instrument moves.

Optimize with Relationships

In most applications one of the goals is to orient measured points to critical part features. SpatialAnalyzer is able to help you accomplish that task by letting an optimization routine use relationships to drive a fitting process. The net effect is the distances defined between objects in each relationship are minimized. You can use this functionality to produce a best-fit between the points and the critical features. So by creating relationships between the measured points and the part's critical features you can find orientations that yield the best performing part. Since SA's relationships can be made between any object types you're able to define a comprehensive set of part features and then optimize their orientation. That's cool!

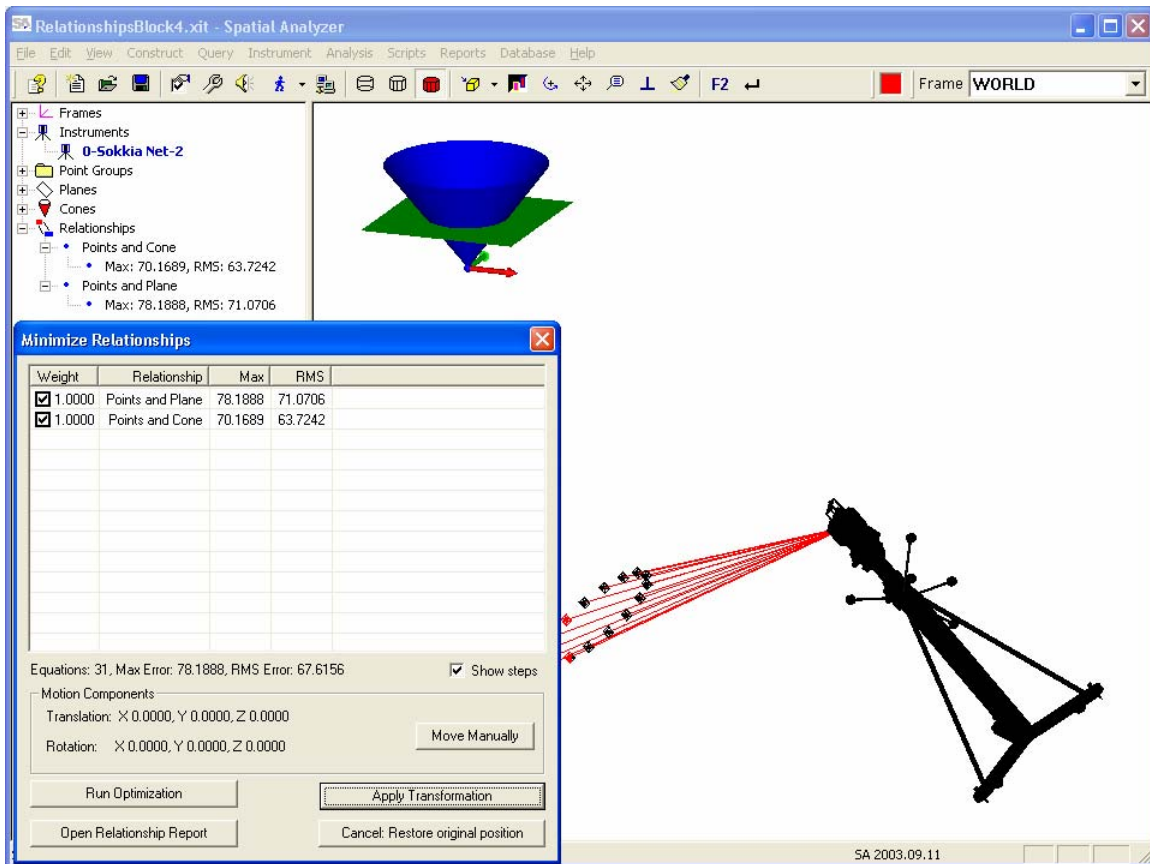
Transform by Minimizing Relationships

To begin the fitting process; select from the SA menu. The first selection asks you to choose which objects to move. Hit the F2 key and just select the points by clicking on the Point Groups check box. If you don't want all of the point groups to move then open the Point Group branch of the tree and then select just the objects you want or need to move.



Select Object to Move

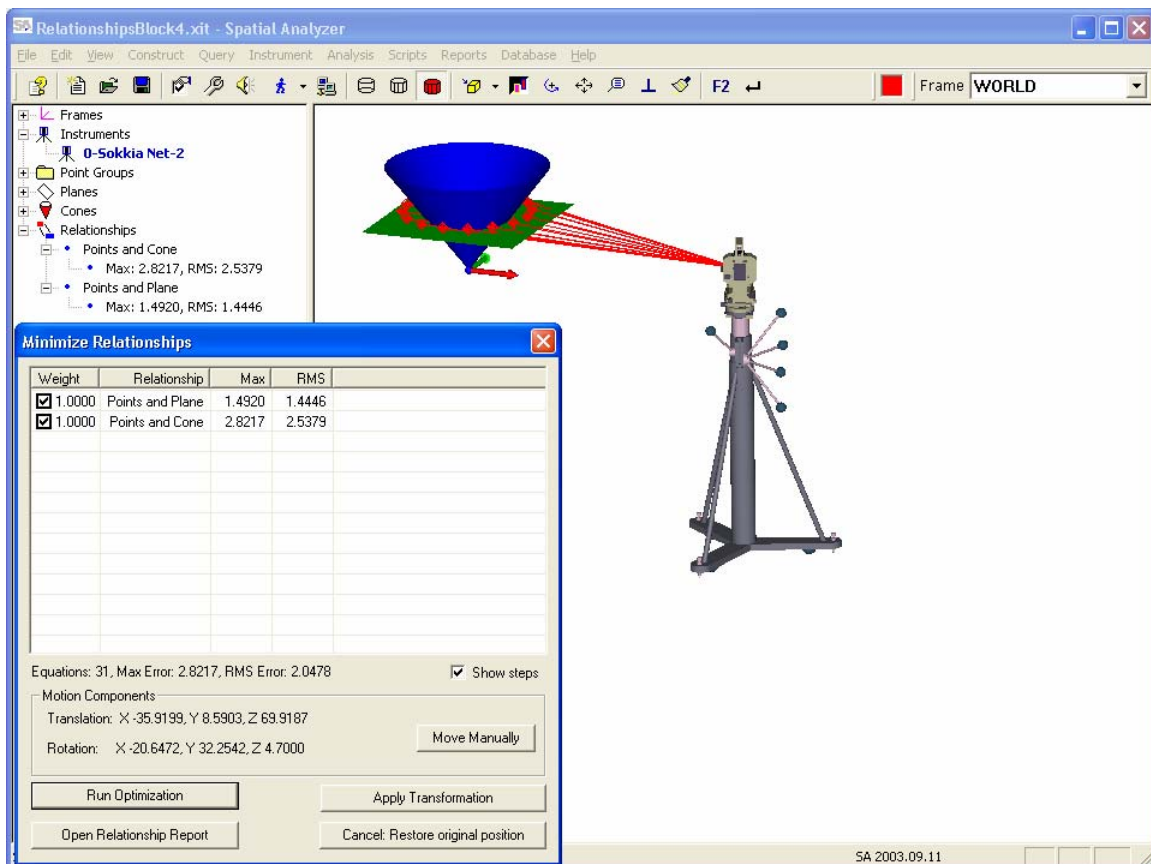
Hit the enter key to end the object's to move selection task. The next step asks you to select the instruments to move. Either graphically select the instrument or hit the F2 key and select the instrument in the dialog. Hit enter when done to complete the selection process.



Minimize Relationships

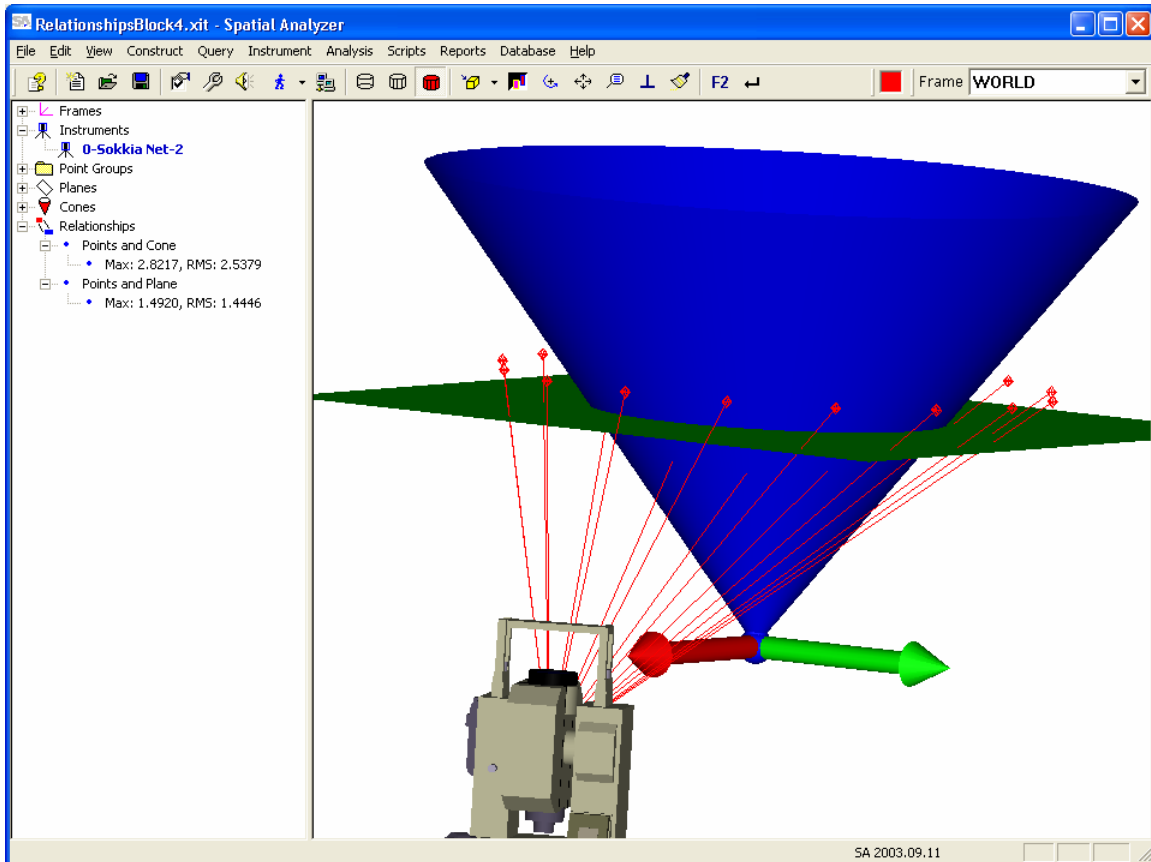
After completing the objects and instruments to move selections the Minimize Relationships dialog is then initialized and presented. An example is shown in the figure above. The Minimize Relationships dialog displays the current relationships their weight and summary statistics along with a number of buttons to start the optimization and then apply the computed transform which will minimize the listed relationships.

Hit the Run Optimization button to solve for the best fit orientation between the circle of points and the cone and plane. When you hit the button you'll be able to see the instrument and points move in towards the cone and plane. The figure below shows the results after the fit. Hit the Apply Transformation button to accept the results.



Results from Minimizing Relationships

Zoom in on the points and rotate the view around to get a perspective for how the points are oriented relative to the cone and plane. The figure below shows that the points are oriented above the plane and radially distributed about the cone. Note how the distance between the points and the plane is balanced against the distances between the points and the cone.



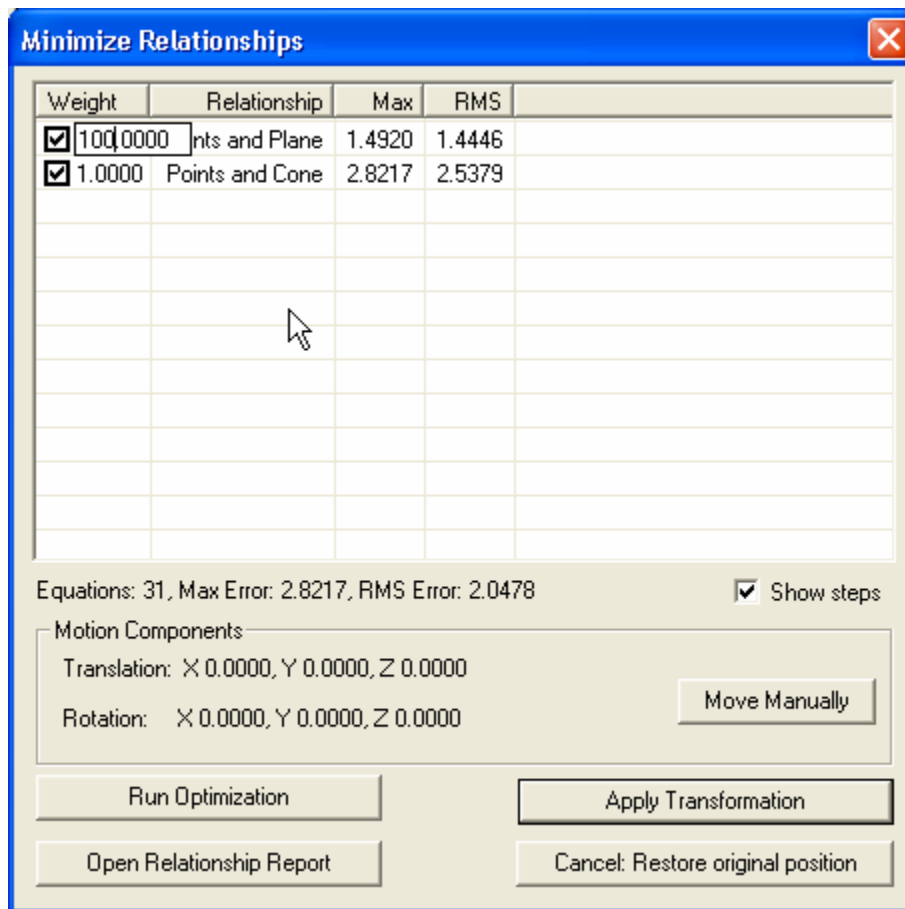
Close-up results of Minimizing Relationships

This optimization allows us to look at some other aspects of relationship fitting. The transformation was not defined in all 6-degrees of freedom (i.e., an infinite number of solutions exist relative to rotations about the z-axis plus 180 degrees relative to either the x or y-axis.) Even though the solution was not uniquely defined the process solved to a stable answer that made sense. Another nice feature of relationship fitting is that we did not have to be concerned about matching point names. The relationship maintains the association between the specific points and the object.

Weighted Relationship Fitting

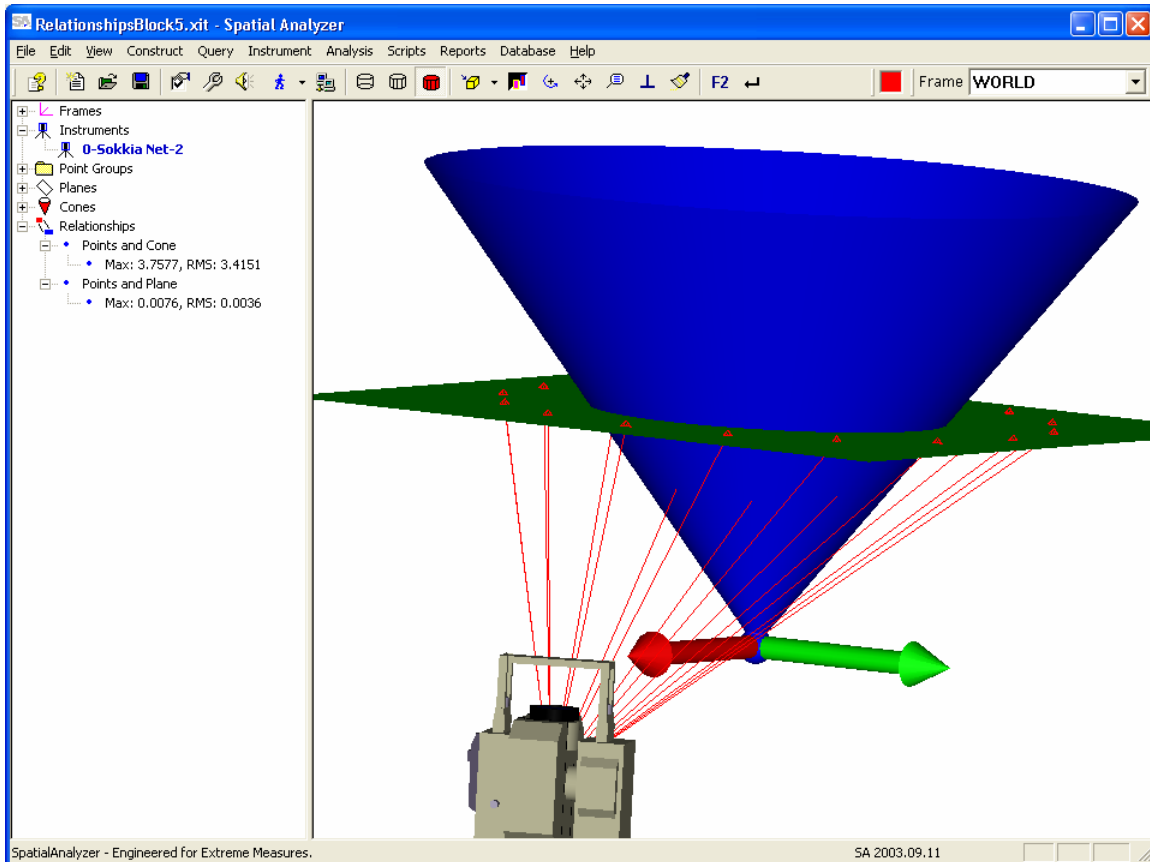
In most cases specific features typically are considered primary while others are important they are used as secondary datum's. Relative weighting of the relationships helps you to achieve this effect.

- Go back to the function.
- Make the same selections for Object's to Move (e.g., Point Groups) and Instrument's to Move (e.g., All Instruments).
- When the Minimize Relationships dialog is up, change the weight of the Points to Plane relationship from 1.0000 to 100.0000. To change the weight; select the 1.0000 on the Points to Plane row with the left-mouse button and then slowly click it again with the left-mouse button. This action puts you into an edit mode to change the weight. (Aren't Windows Controls fun!)



Weighting Relationships

After the optimization is complete hit the Apply Transformation button. The figure below shows the new optimization solution.



Results after weighted Relationship Optimization

By weighting the plane to point relationship with a higher value we are asking the optimization engine to consider this feature as more important. We can see the effect between the two relative weighted constraints in the figure above. The points are down on the plane and yet they are still centered on the cone.

Add Additional Constraints with Relationships

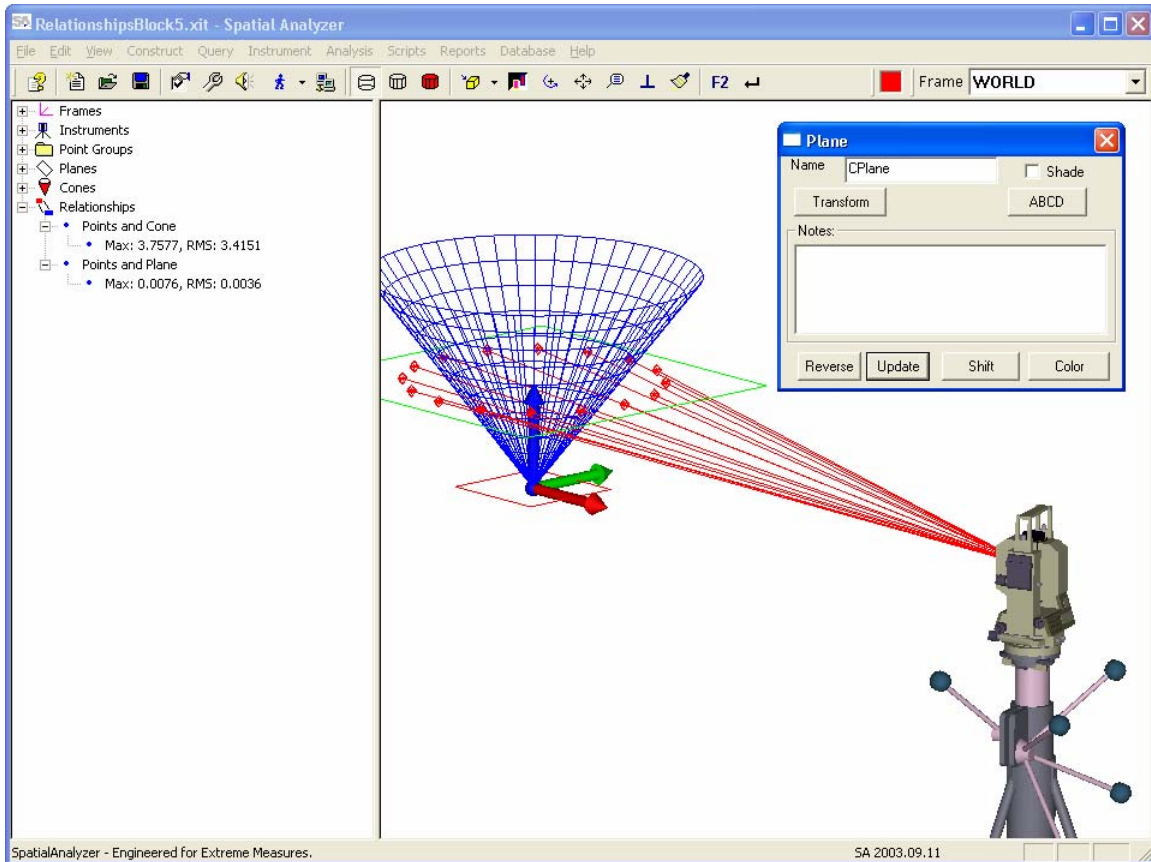
The instrument and points are now oriented to the cone and plane. However their position relative to rotations about the Z-axis was not defined in earlier steps. In this case the rotation about the axis could be called clocking. To control the clocking of the instrument and points about the Z-axis we can add clocking plane constraint into the job. By making a relationship between the plane and point to control the clocking rotation, we can then re-solve the new set of relationship to achieve the orientation we need.

Add a Clocking Plane

The first step in further controlling the orientation is to add the controlling feature. In this case it is another plane.

- Add another plane into the job by using menu option.
- After adding the plane; change its name to 'CPlane' by editing the name in the plane properties dialog.

The figure below shows the now plane and its properties dialog.

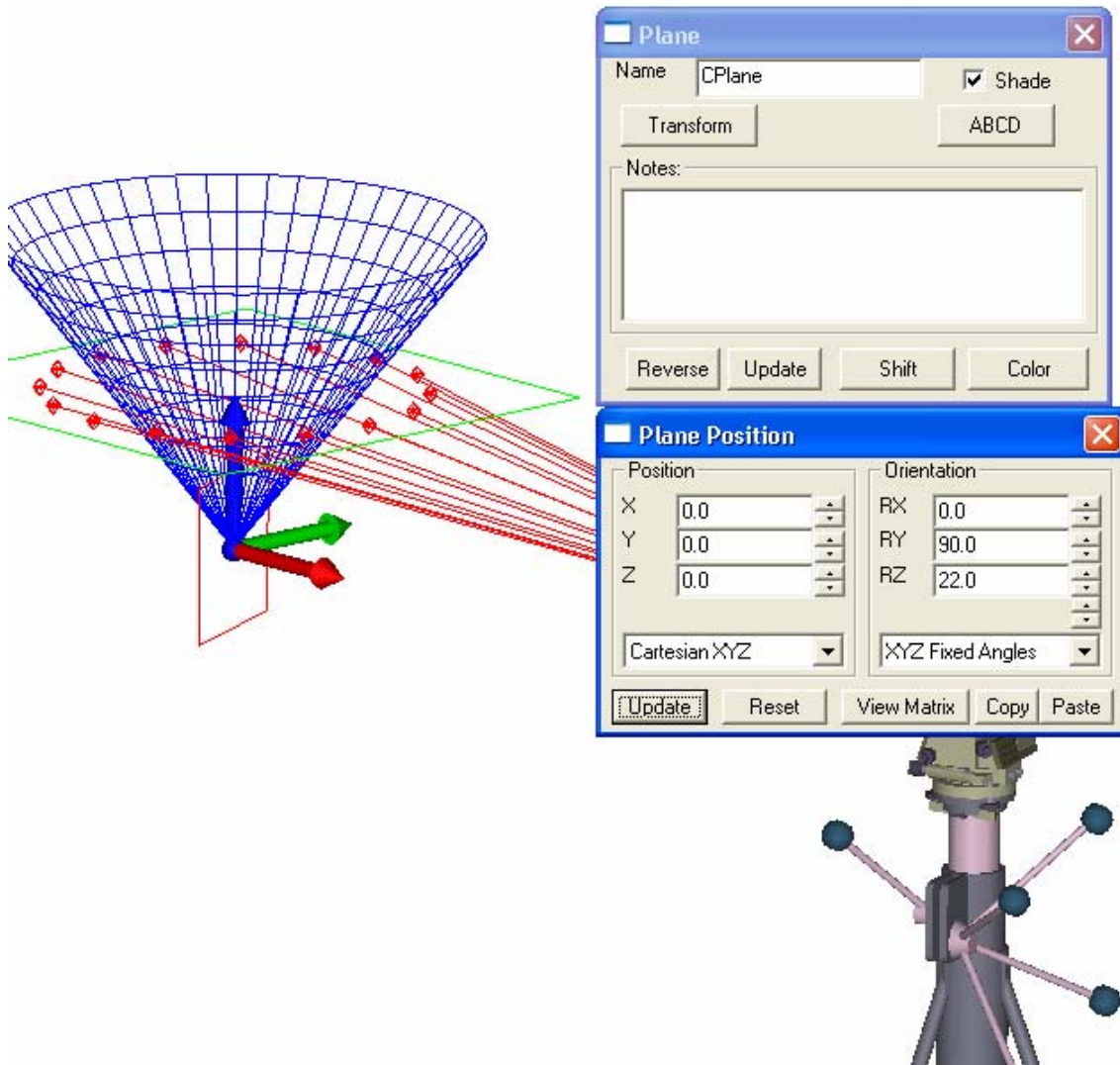


Adding Clocking Plane for Transformation

To position the plane to control clocking we need to rotate it about either the X or Y axis and if need rotate it about the Z-axis to set the clocking angle. One method to accomplish that is to transform it.

- Select the Transform button on the planes properties dialog.
- Change the RY component to 90 degrees
- Change the RZ component to 22 degrees
- Select the Update button and then close the dialog

The figure below shows the Plane Position dialog with the rotation values.



Rotate Clocking Plane into Position

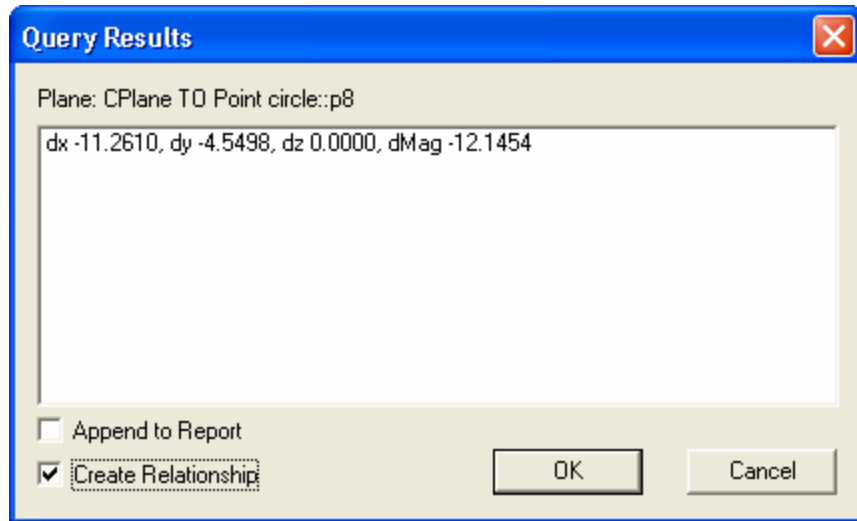
Add a Clocking Plane Relationship

To use the new feature in the optimization we need to model it with a Relationship. In this case we will choose a single point and an object to form the relationship.

- Turn the point names on by selecting the
- Select the menu option.
- Graphically select the point named 'p8'
- Graphically select the clocking plane 'CPlane'

The Query Result dialog appears with the difference between the point and the object. For this application we are creating a relationship.

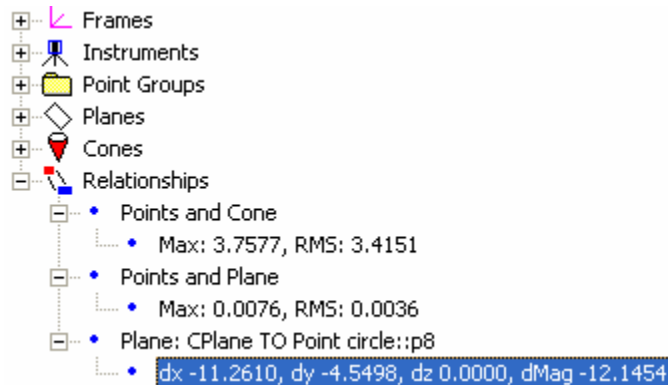
- Select the Create Relationship option on the dialog
- Close the dialog



Create Point to Object Relationship with Query

The differences between the point and plane shown in the figure above will likely be different in your job because of the noise injected when we fabricated the measurements and moved the instrument with the drag instrument option.

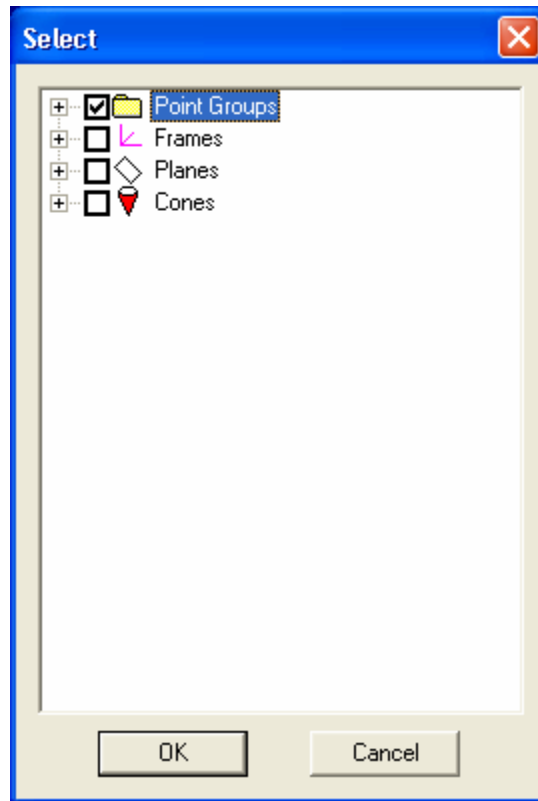
The tree view shows the new relationship.



Relationship Report on Tree view

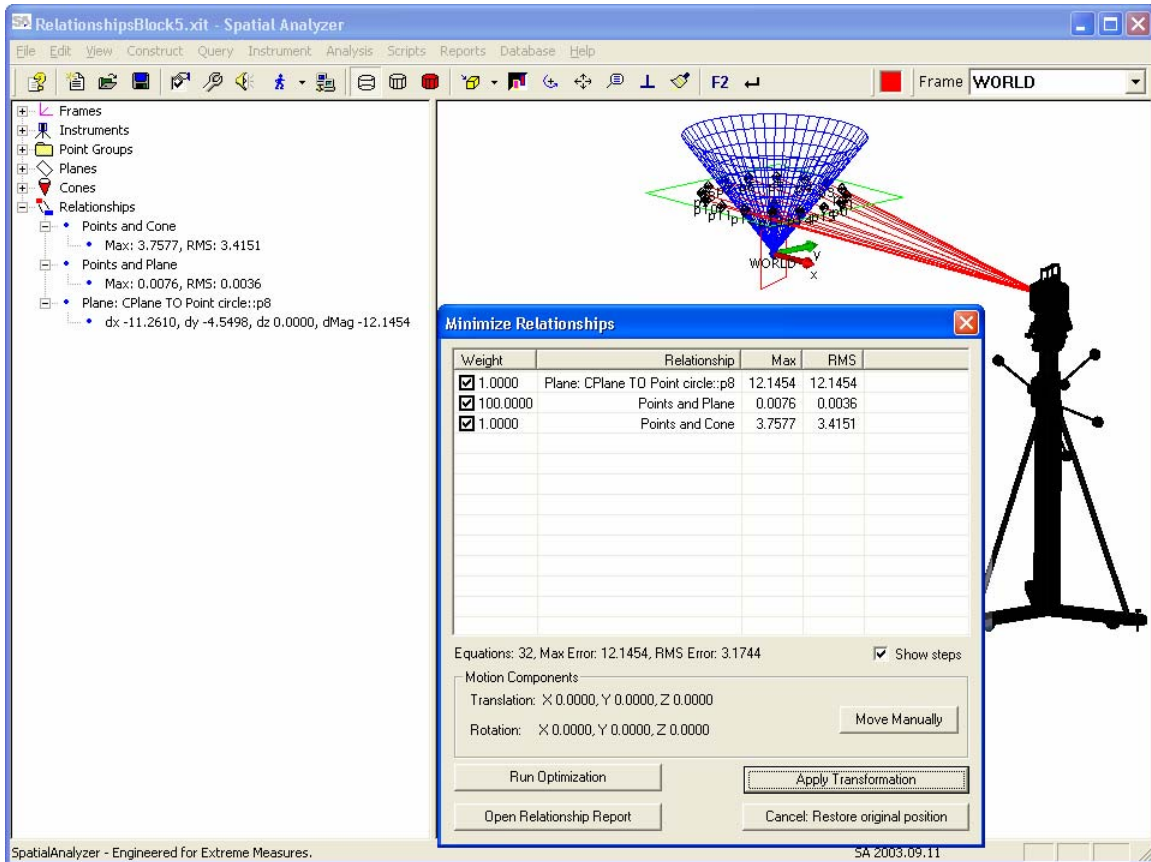
Minimize all the Relationships Simultaneously

Follow the same process to use the relationships to drive the fitting process; select from the SA menu. The first selection asks you to choose which objects to move. Hit the F2 key and just select the points by clicking on the Point Groups check box.



Select objects to move during Relationship Fit

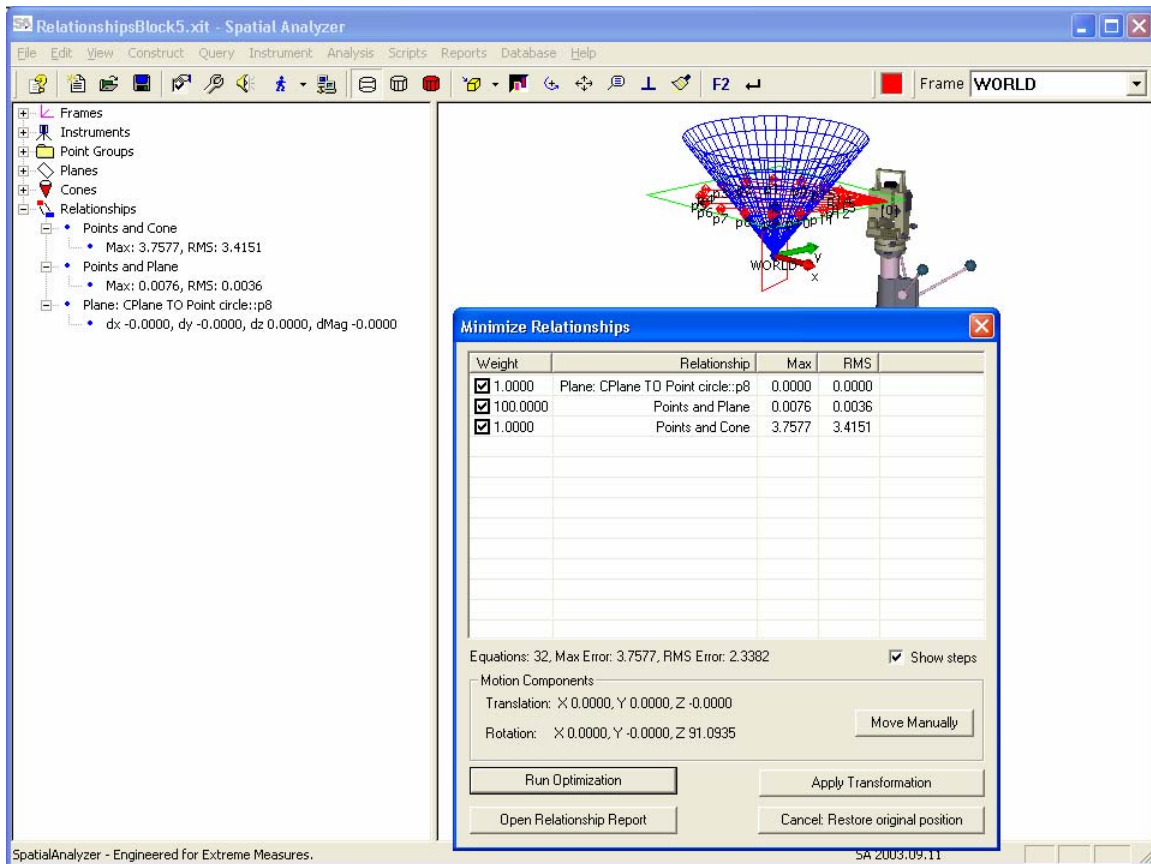
Hit the enter key to end the object's to move selection task. The next step asks you to select the instruments to move. Either graphically select the instrument or hit the F2 key and select the instrument in the dialog. Hit enter when done to complete the selection process.



Minimize Relationships

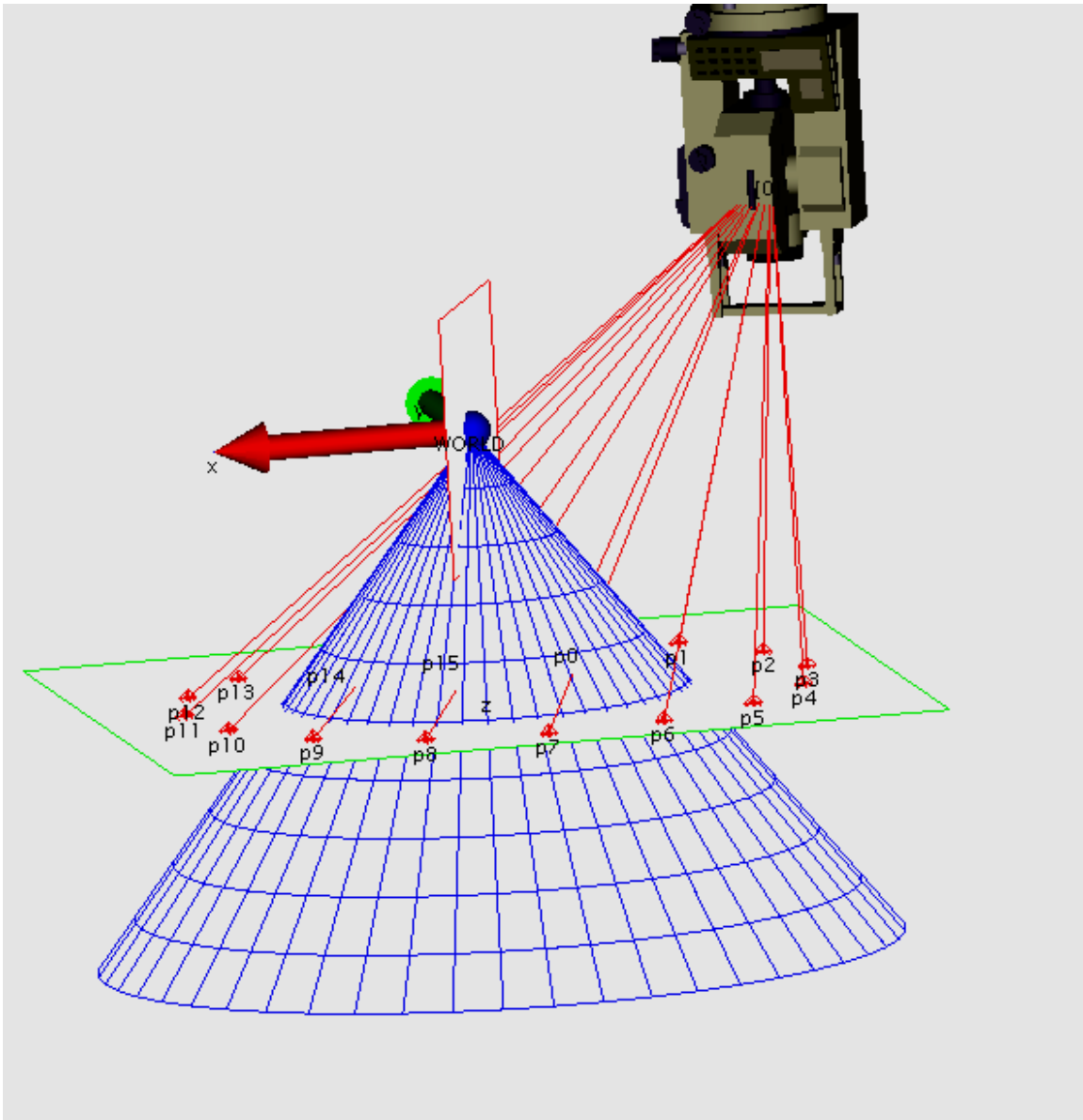
After completing the objects and instruments to move selections the Minimize Relationships dialog is then initialized and presented. The Minimize Relationships dialog displays the new relationship along with the two previously defined.

Hit the Run Optimization button to solve for the best fit orientation between all of the relationships. When you hit the button you'll be able to see the instrument and points move to minimize the relationships. The figure below shows the results after the fit.



Results after adding clocking plane constraint

The results show that the clocking plane relationship defined the relative position for the points and instrument about the z-axis. While the clocking plane relationship was doing its job the plane and cone relationship constraints were also being optimized. The three relationships work together to define the functional orientation of the measured points and the geometry.



Close-up of results after adding clocking plane relationship

The figure above shows the relative position of point p8 and how it now lies on the clocking plane.

We should note that the orientation still has multiple solutions. Other valid orientations are available by rotating the instrument and point by 180 degrees. To uniquely define the orientation another clocking plane or additional features need to be added and then relationships between those features and their measurement created.

WATCH WINDOWS

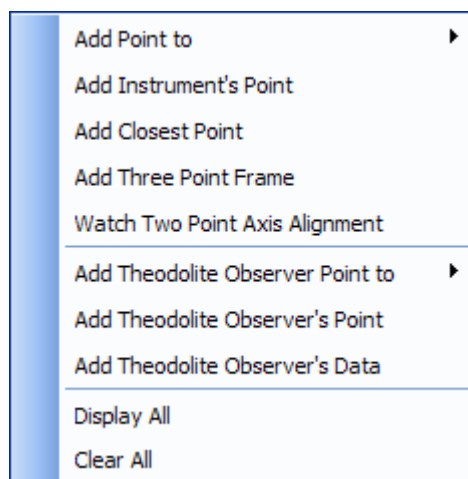
SpatialAnalyzer uses watch windows to display live data taken from measurement devices. These watch windows may be used to compare the current measurement from an instrument to a surface, another point, the closest point, or other entities.

These windows are re-sizeable and configurable to suit the user's preferences. In addition, when a tolerance is set, these windows will indicate out of tolerance conditions by changing the background color of the window.

Watch windows will update when the instrument performs a Query. This is different than when the instrument takes a measurement. Queries are sent to **SpatialAnalyzer** to update the watch windows and the graphics but are not saved as measurements. You can query an instrument from its instrument interface. In addition, where applicable, you can set the instrument to query at a given frequency. This is the mode you will usually place the instrument in when using watch windows.

Complete Coordinate Instruments

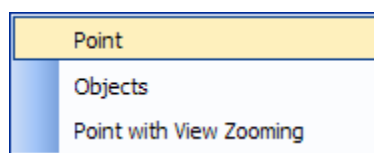
Instruments that yield a complete coordinate value without cooperation from other devices use the first set of watch windows. From the View menu, select Watch Windows. This will give you a menu describing the different watch windows that are available. The second set of watch windows go with the Theodolite Data Observer, which will be discussed later.



Watch Window Menu

Add Point to ...

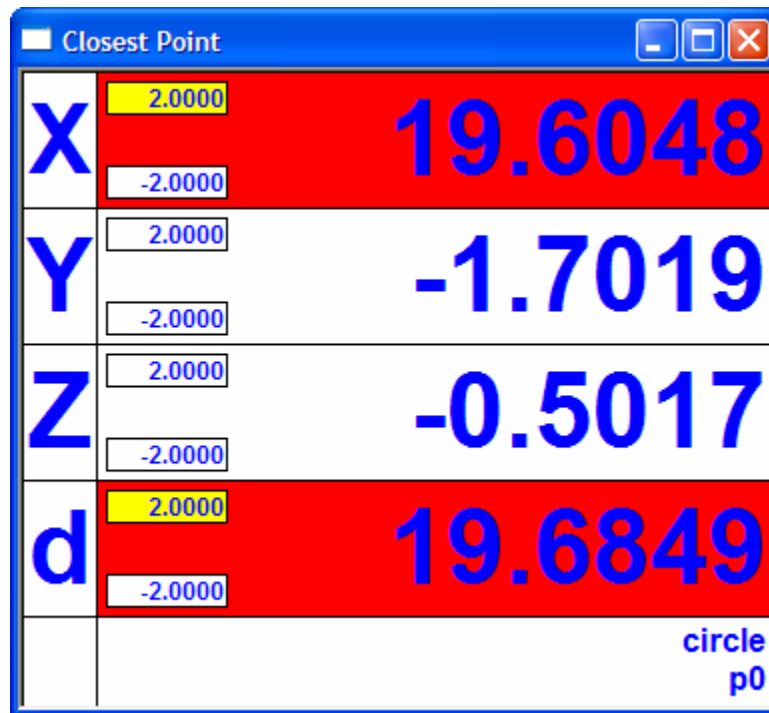
This option displays another menu as seen in the figure below.



Point Watch Window Options

When you select any of these options, you will be asked to select the instrument providing the data (unless there is only one instrument, in which case the choice will be made automatically). Next, you will be asked to select the entity you wish to watch. Once the entity is selected, you will be asked for an optional tolerance.

If, for example, you selected Point from the menu in the figure above the following watch window would appear after the appropriate selections:



Point to Point Watch Window

The point that is being compared is listed in the title bar of the dialog. The deltas are given in the body of the dialog along with the magnitude of the delta vector. In this case, the magnitude is beyond the specified tolerance and is therefore highlighted in red.

The watch windows for the other geometries are similar, but contain different information in some cases.

Add Instrument's Point

This watch window displays the current coordinate from the instrument in the working coordinate frame.

Add Closest Point

This type of watch window will compare the instrument's current coordinate to all the points in the model. Whichever point is closest will be compared to the instrument's point and the delta's displayed.

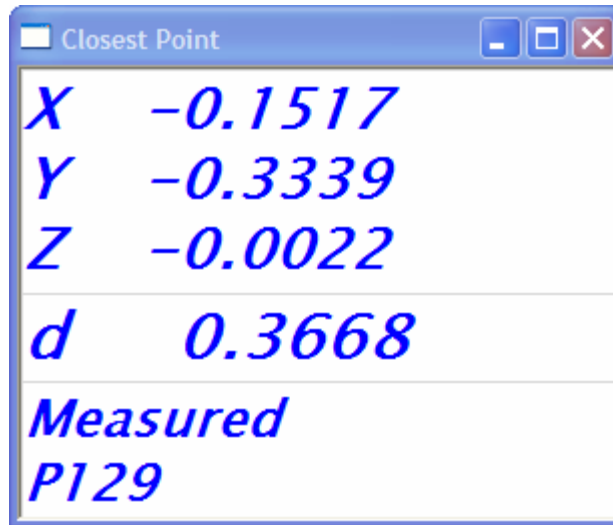


Figure 0-1 Closest Point Watch Window

The closest point functionality is useful for tasks such as tool building where you must position a series of monuments to match a set of design coordinates. It eliminates having to go back to the Analyzer and specify a different watch window for each detail you are building.

Theodolite Data Observer (TDO)

In the case of theodolites, multiple devices must be used to create a single coordinate value. The TDO performs this function. It allows you to then create watch windows using the current point computed based on all instruments being updated in the TDO.

The watch windows listed in this section may be linked to the TDO. The options are similar to those for complete coordinate instruments. For more information on the TDO, see the Menu Reference chapter.

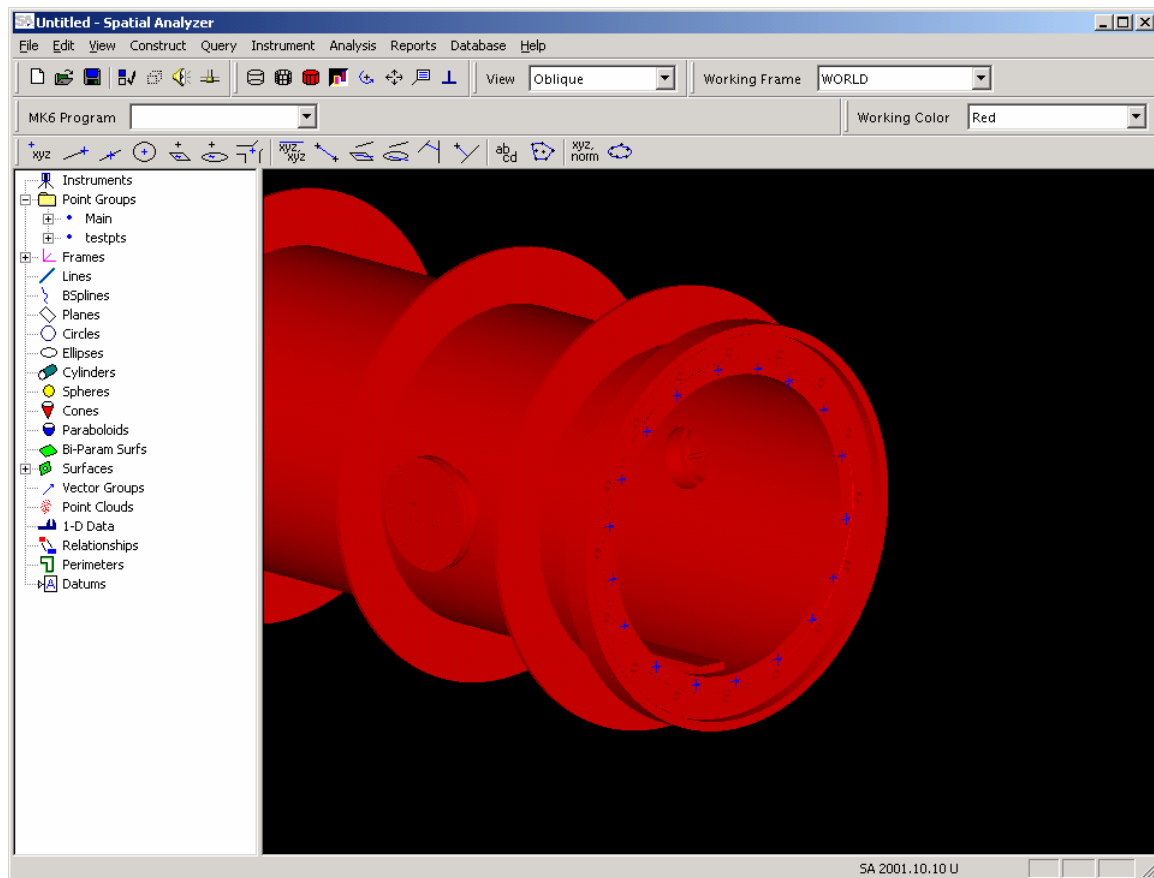
SIMULATION AND SURFACE ANALYSIS

Measurement Survey Simulation

There are many reasons for wanting to simulate a measurement survey. Perhaps you wish to verify that the instrument can actually see the target without obstruction. Perhaps you wish to determine what uncertainty you might expect from a given instrument set-up. By trying several different configurations you might gain insights that help you to optimize the configuration to minimize uncertainty.

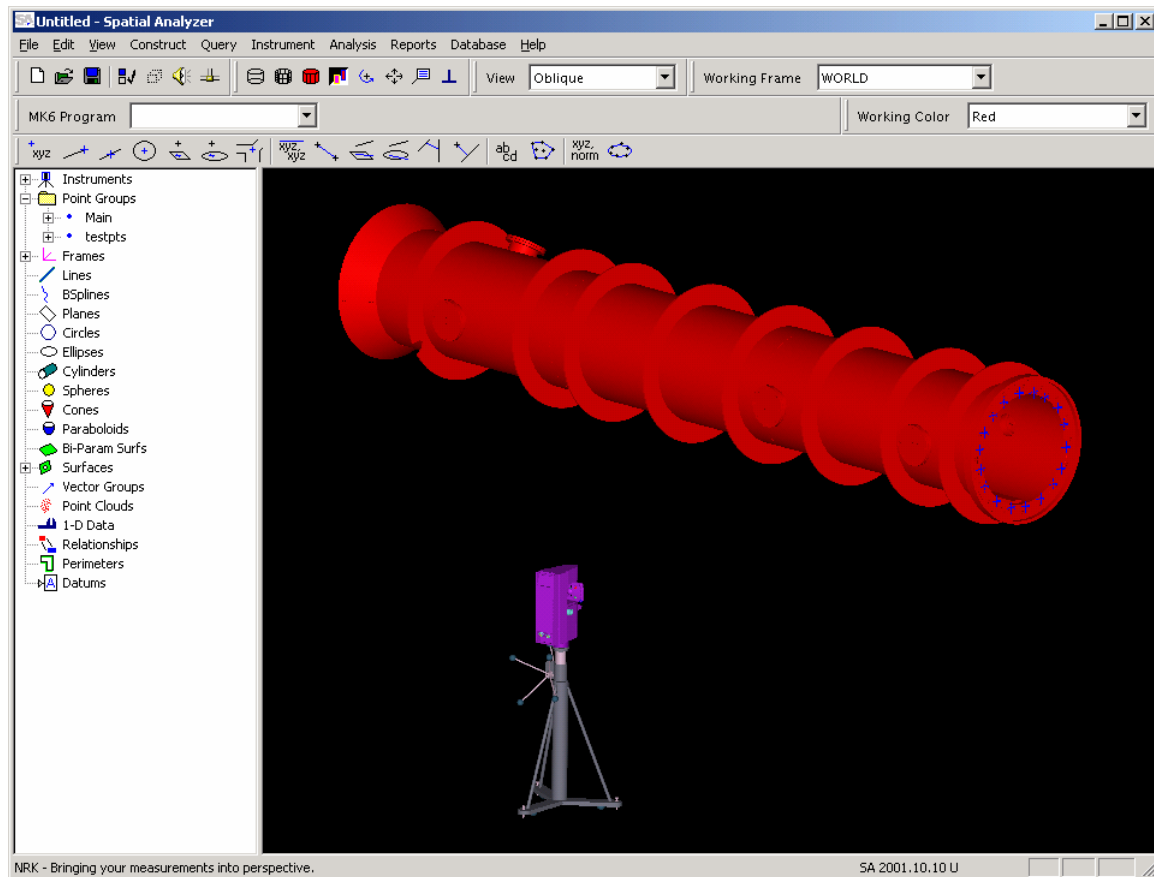
Let's use a relatively simple real-world example to illustrate simulation. The figure below depicts a part imported from a CAD file. Suppose we needed to survey the end flange of this structure to determine how flat the flange actually was. When this task is done for real, some of the flange irregularities will be due to the deformities of the flange and another component will be introduced due to errors in the data acquisition. Let's compare two possible survey configurations with an SMX tracker.

The figure below depicts the CAD part in SA. We have manually constructed several points on the face of the flange by selecting the Construct->points->On CAD surface by clicking. These points are in the testpts group.



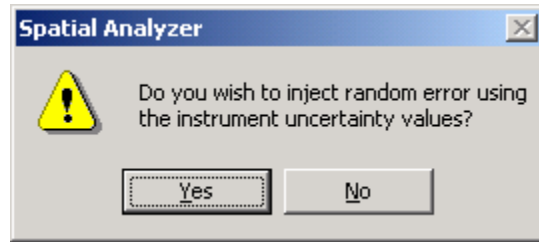
CAD part in SpatialAnalyzer with Points on the front surface

Now let's add a tracker to the model. You'll notice that the instrument gets added a position near the origin. You will now have to roughly position the instrument where you think it may actually lie during the survey. This is done by either editing the instrument properties or by doing the position instruments by dragging. Either option is appropriate.



CAD part in SpatialAnalyzer with Laser Tracker

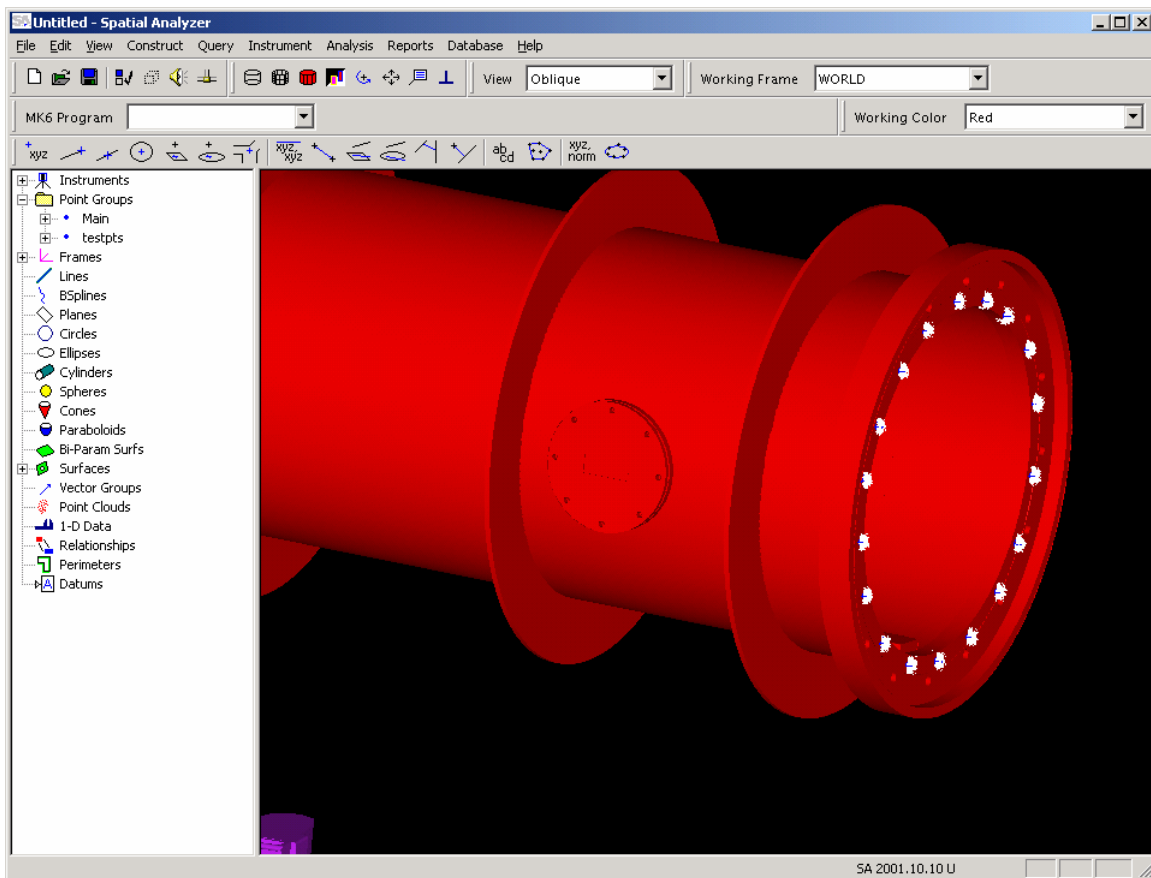
Now we can simulate observations of these targets. If you wish to change the instrument accuracy properties, you may do so in the instruments properties page. This might be useful if you wanted to simulate performance in non lab-like conditions, as is the case when you use the instrument outdoors. To fabricate shots, select the Analysis->Simulation->Fabricate Measurements menu option. You will be immediately prompted for the instrument, select the tracker, and the points for which you wish to fabricate shots. You can either select these points individually by clicking on them, from a tree view by hitting the F function key, or by drag capturing the points by holding down shift and dragging the mouse. Also any combination of the above may be used as well. You will be prompted with the following dialog.



Random Noise Injection Dialog

Select yes and you will see the points (drawn as pluses) converted into targets (drawn as diamonds).

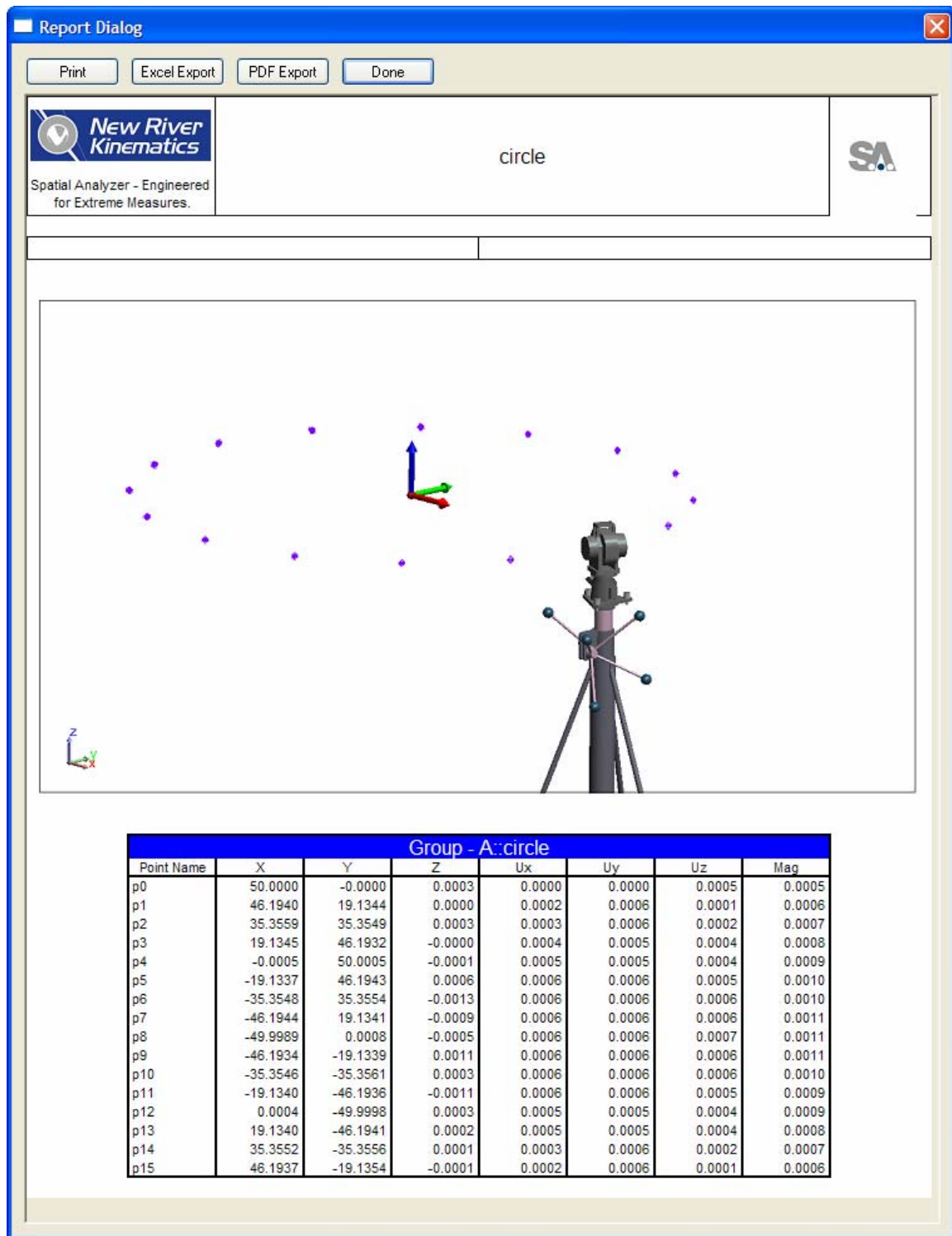
Now we have data with fabricated data that contains simulated instrument error. We could view the uncertainty clouds of these points by selecting the Analysis->Draw target sensitivity clouds. This produces the following.



Simulated Measured Points on the front surface

Note that in this configuration the clouds appear as disks that seem to penetrate the surface. We could also view this information qualitatively by clicking on a point to access its uncertainty along with other properties or by viewing the coordinate report and right

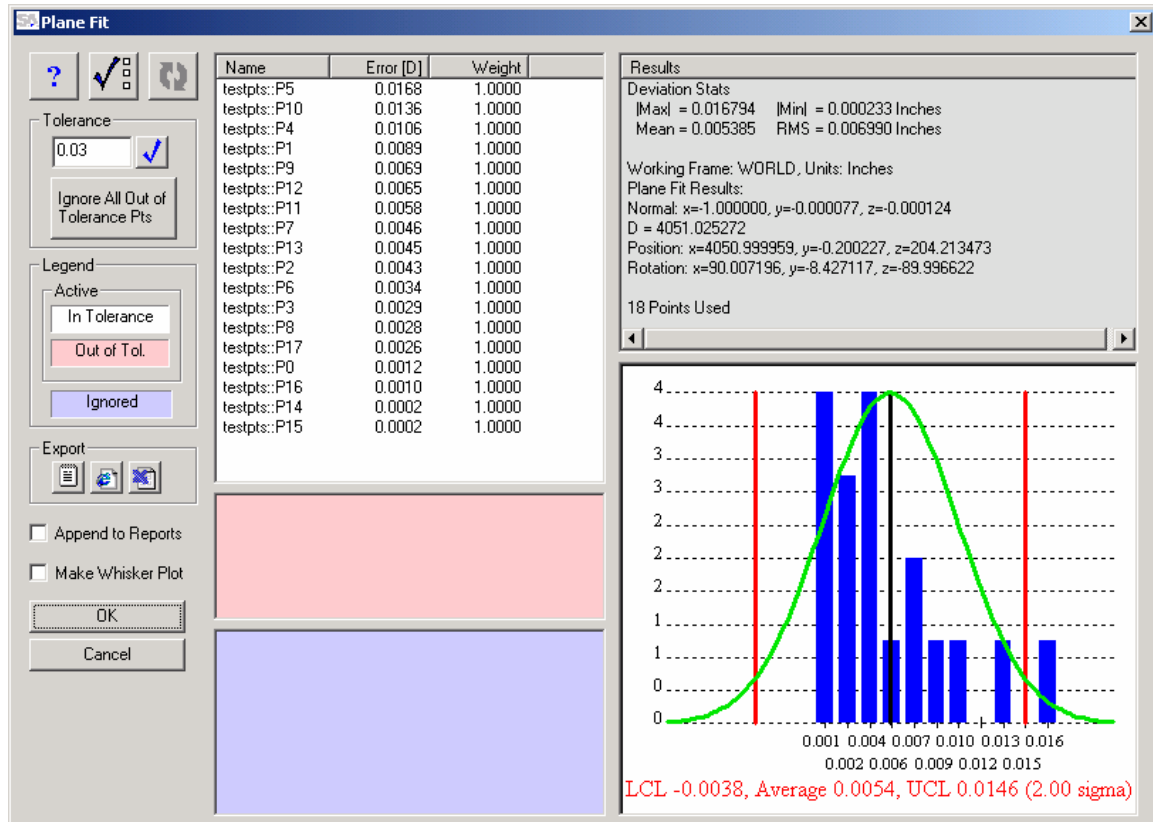
clicking in the uncertainty columns. You will see pop-up menu with options for computing uncertainties – select compute all uncertainties and the report should appear as follows.



Coordinates Report with Uncertainties

Values in the reports are always presented in the current working coordinate frame. You could change frames and view the report again to see that data relative to some other reference.

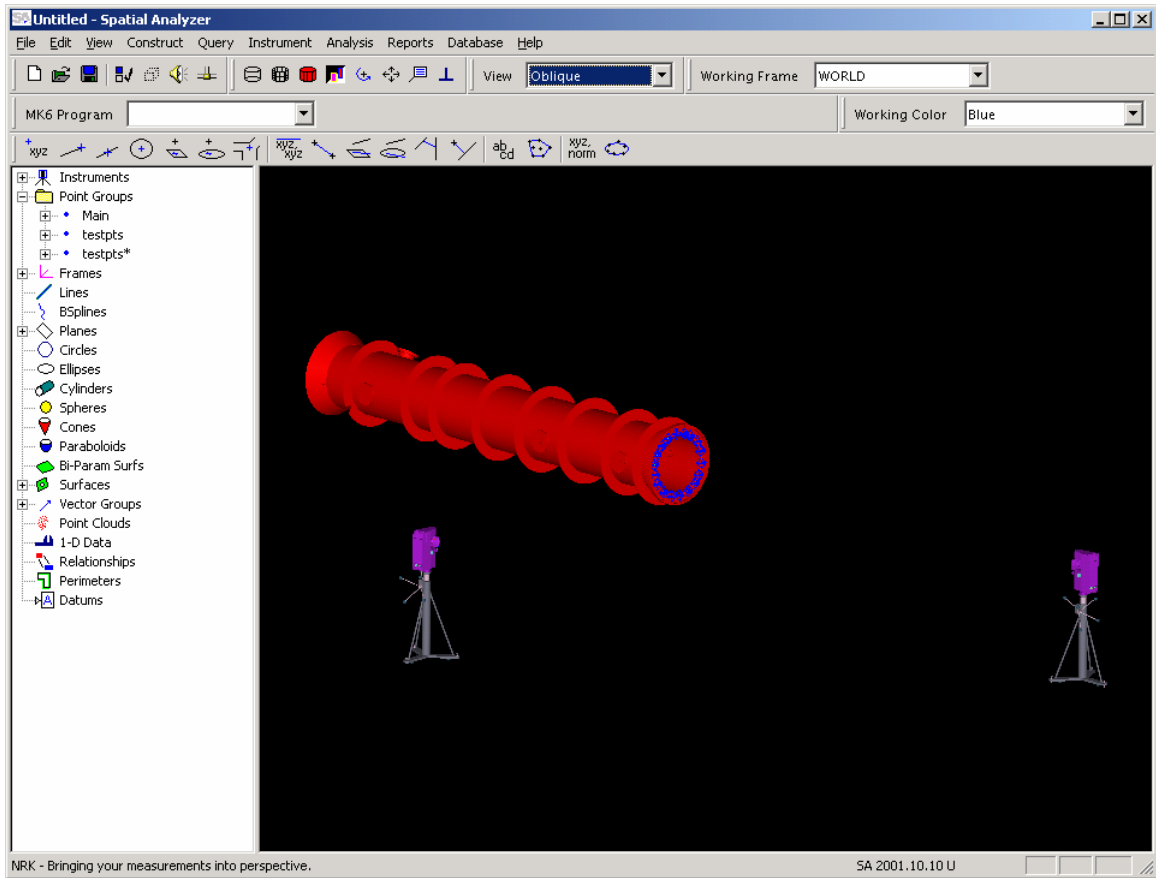
This tells us about the uncertainty of the points, but maybe what we are really interested in is how planar are the points? After all we started by wanting to characterize the flange flatness. We can get a better feel for how the instrument and set-up effect this by fitting a plane to the points. To do this select the Construct->Plane->Fit to points menu option. You will be prompted to select points and after doing so a fit will take place and the fit dialog will appear.



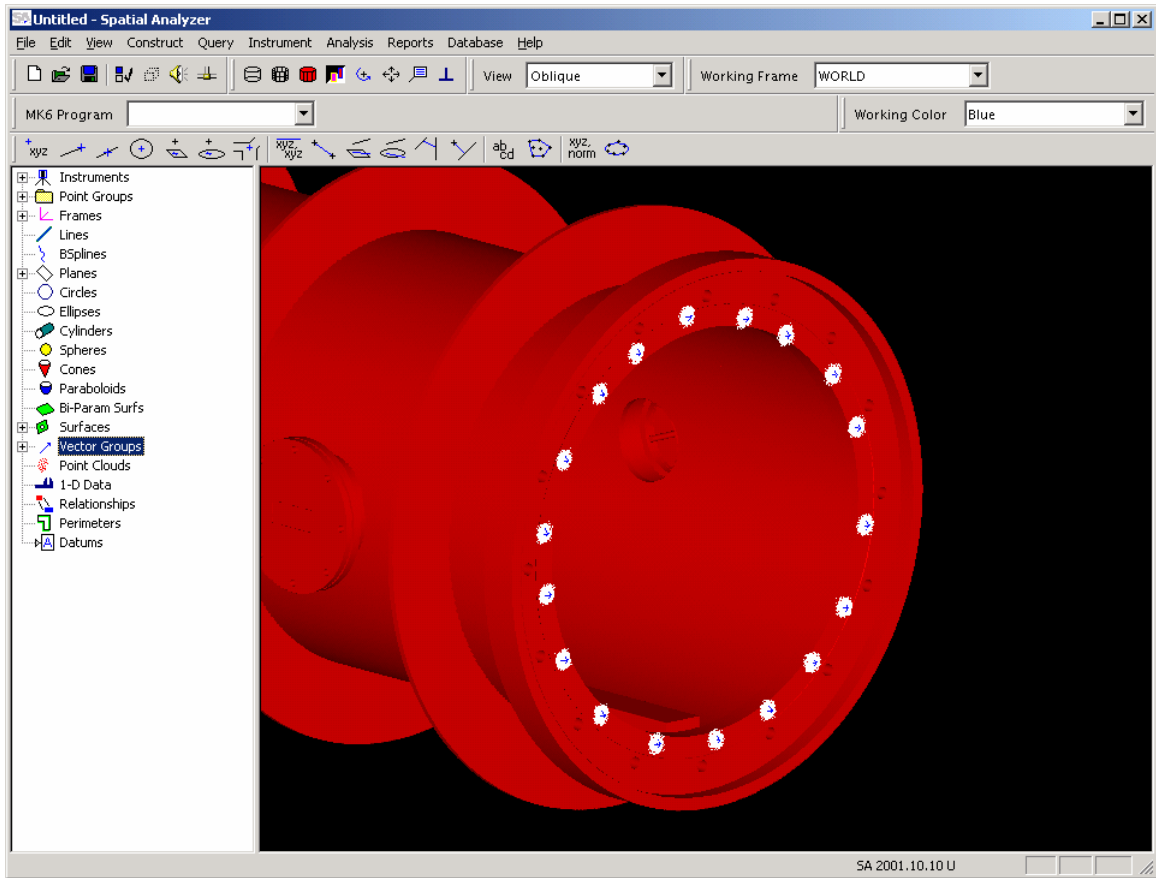
Geometry Computation Results Dialog

Notice that in this dialog you can view the planar deviations as well as the resulting plane parameters. For example the max deviation is 16 thousandths and the plane axis slightly off of normal. Note that this dialog also presents several export options, the ability to append the report to the fits report, and the ability to make whisker plots when done.

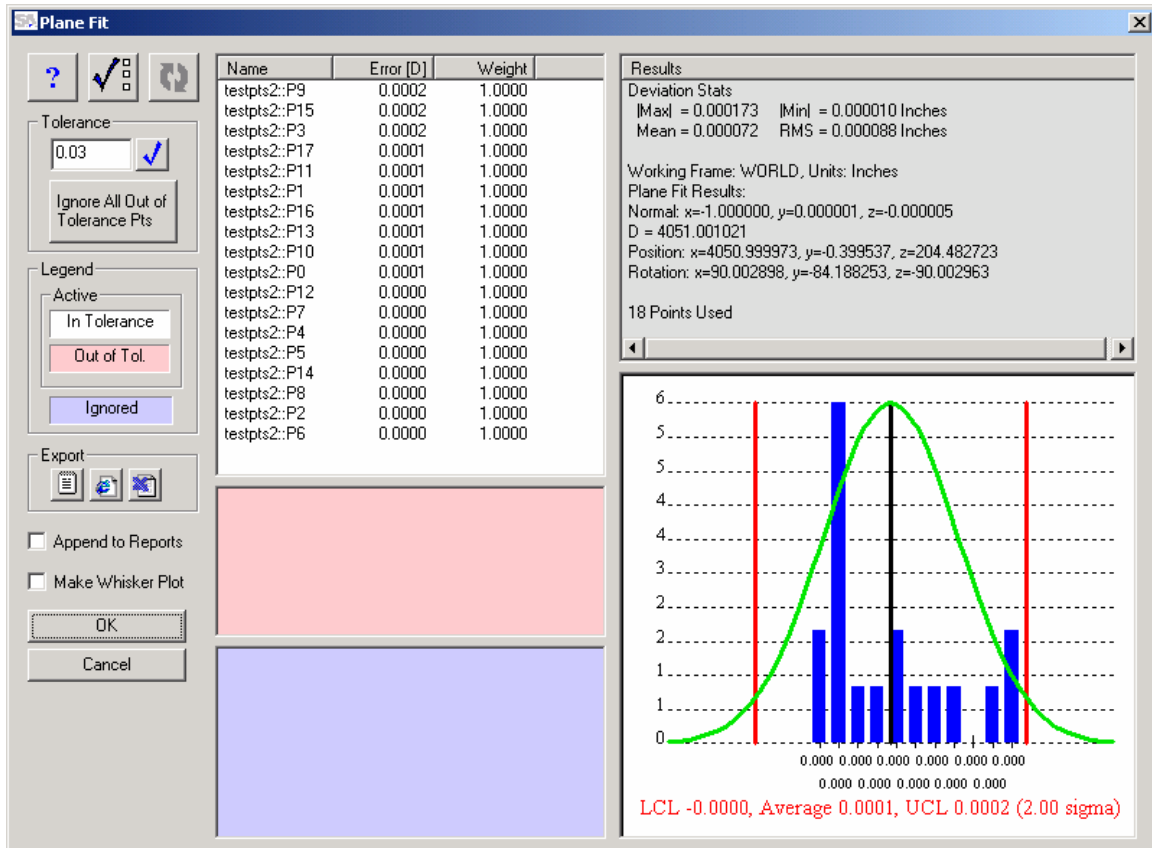
Now we can investigate doing this same operation from a different instrument position. We can either move the instrument or add another. With either process just follow the procedure outline above. The following screen shots and results were taken from such a set-up.



A Second Tracker is Added



Points are Re-measured from the Second Tracker Location



Plane Geometry Computation Results Dialog with two Instrument Positions

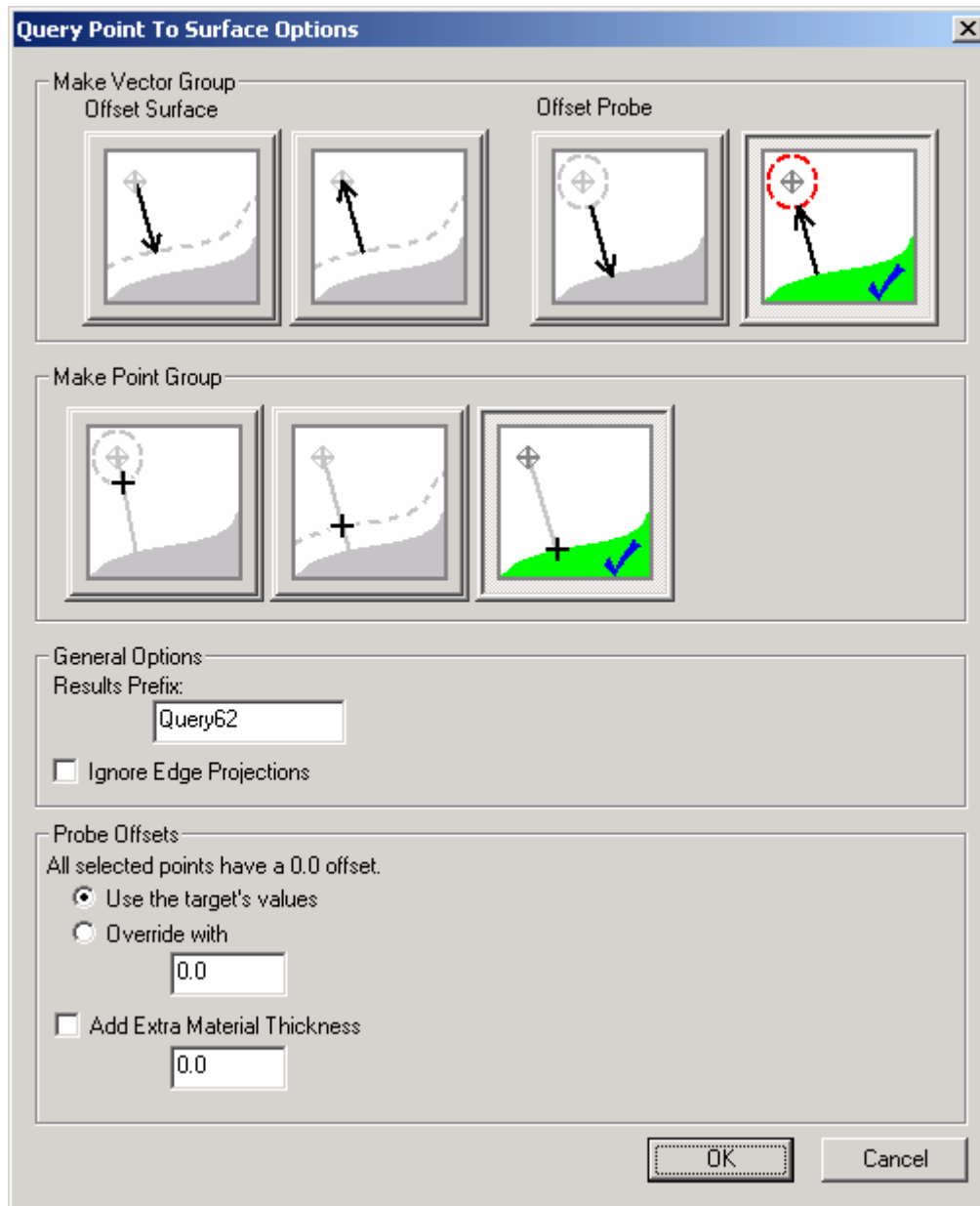
Note that the maximum deviation is now less than 1 thousandth of an inch. Clearly if you wish to certify that the flange was flat to within 0.015 inches, the first set-up would be inappropriate because the instrument errors are larger than what you're trying to certify. That's even with a perfectly flat flange!

Surface Analysis

SA allows you to easily compare your measurements to the CAD model. To do this, you must first have the measurements transformed into the CAD coordinate system. This can be done using tooling points, or the surface geometry.

Select **Query->Points to->Surfaces**. This will prompt you to pick the surfaces, then the points for the query. Remember, you can individually click the points, Shift-drag a window over them, or press F2 to pick them from a list.

Once this selection is made, you will see the query options dialog:



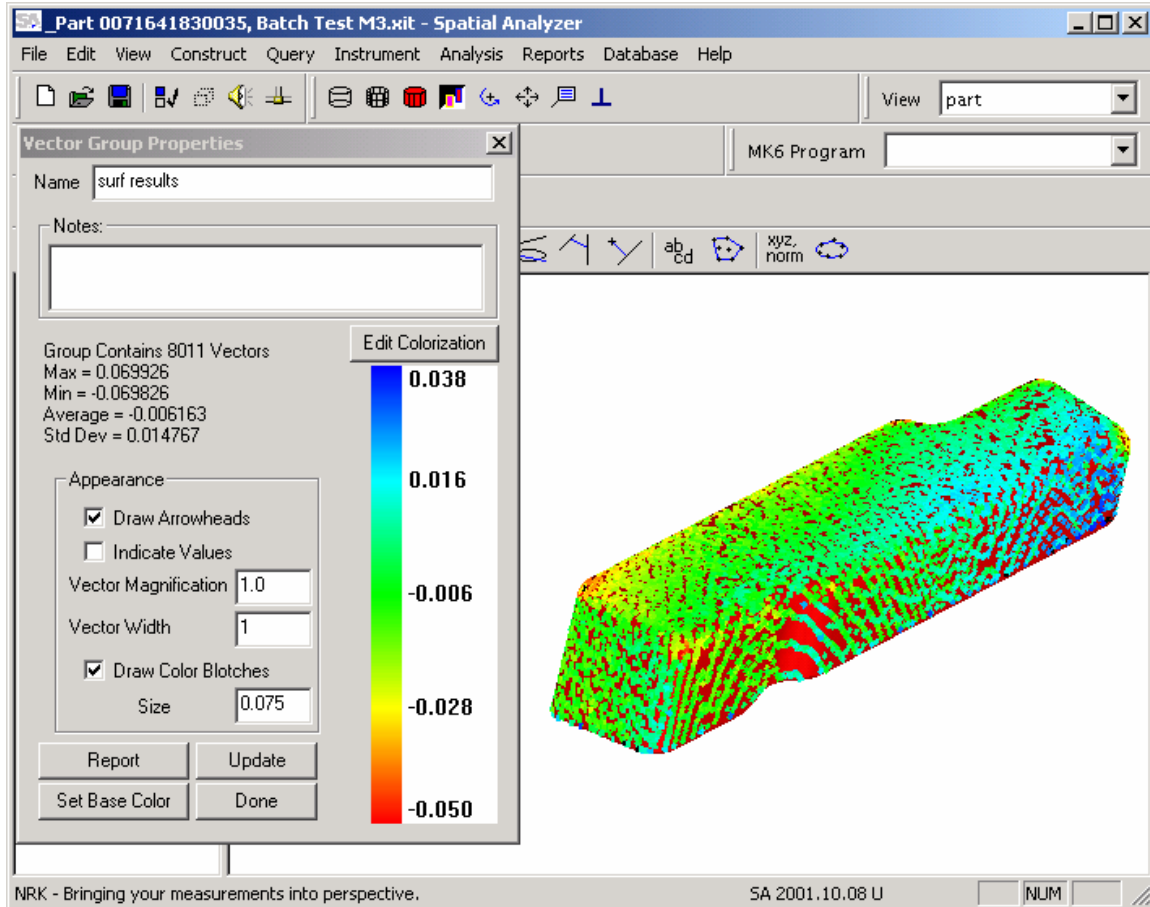
Query Points to Surface Options

In this dialog, you specify how the query should be performed. There are many options, especially in dealing with reflector offsets. Notice that there are two basic categories: make a vector group, and make a point group. The vector group option will create a whisker plot showing the errors. The point group options will create points projected to the surface, reflector, or offset surface.

You can input the name prefix that will be used for the created objects. Also, you can press multiple buttons in each category. The probe offsets area will show you the range of offsets contained in the measured points, and allow you to override the value. In addition, you can add extra material thickness. This is useful when you have a ¼ inch

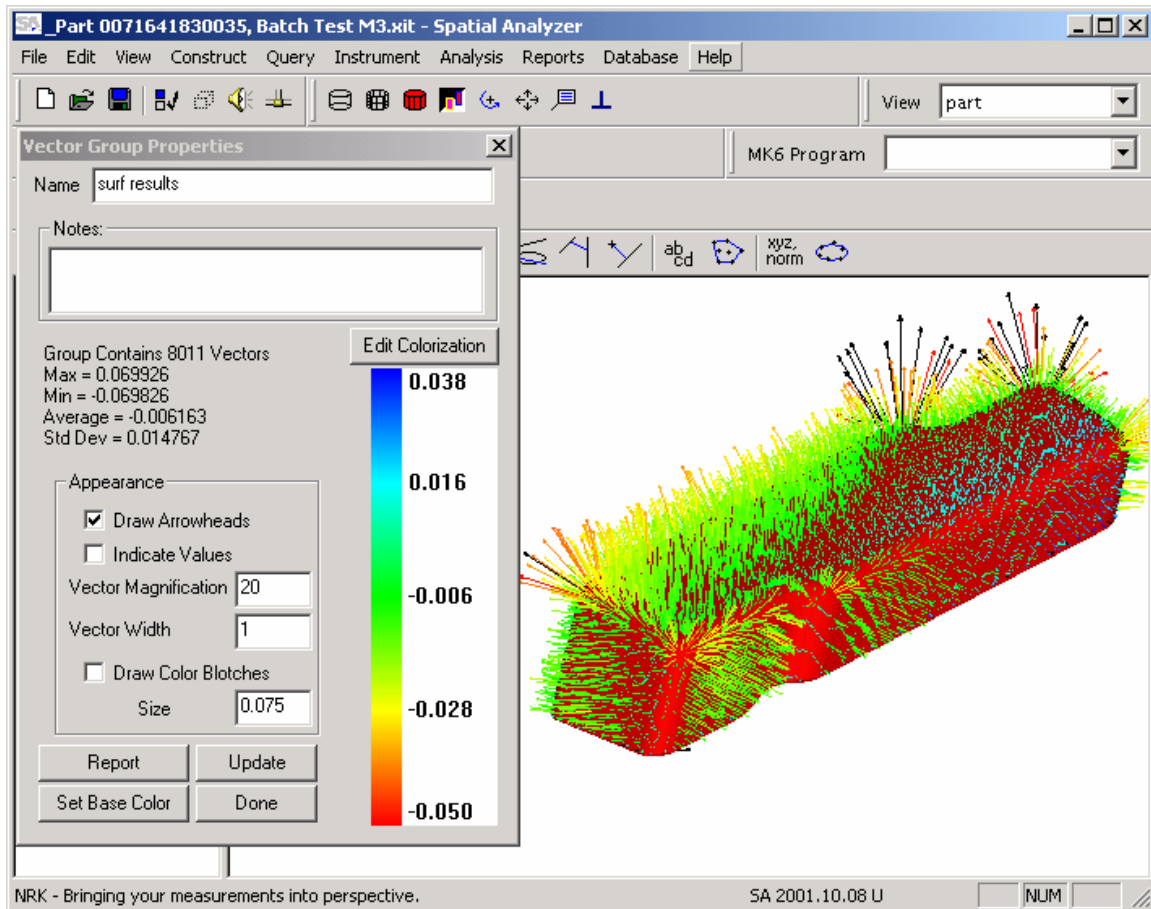
steel surface, for example, but are measuring on the opposite side of the CAD definition. In such a case, enter 0.25 as the material thickness.

Once you have selected the appropriate options, press OK. This will create the objects you requested. In the case of whisker plots, the color legend will also appear:



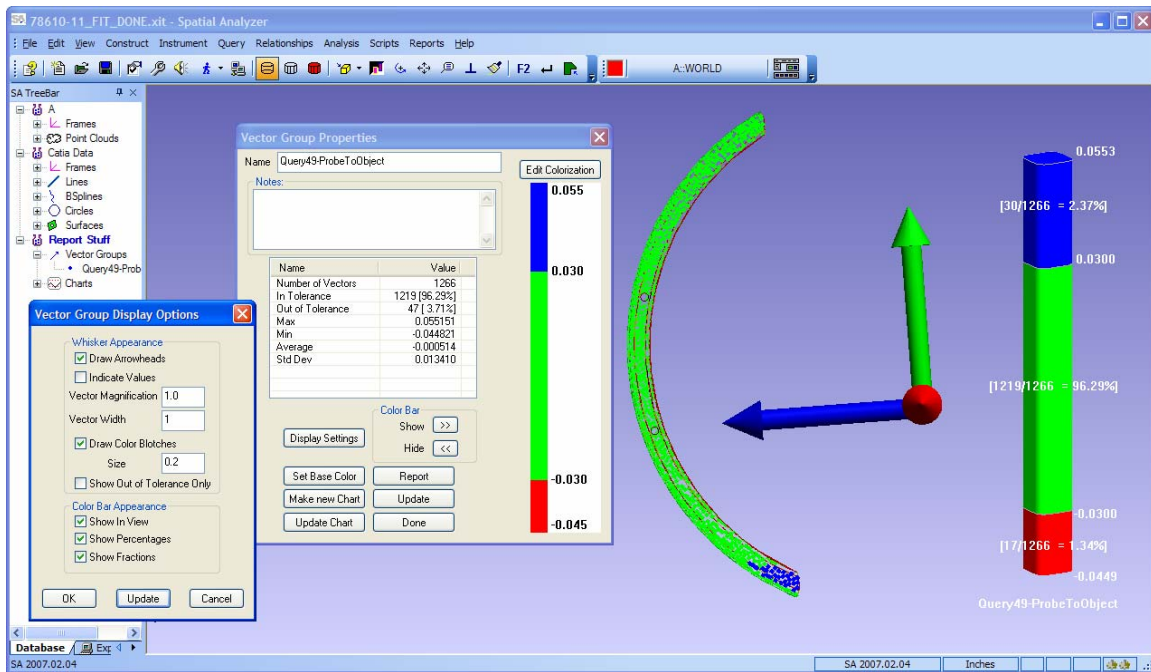
Point to Surface Query Results with Color Blotches

The vector group properties dialog allows you to change display modes. In this case, we are using the “color blotches” option. We can also switch to whiskers:




Point to Surface Query Results with Vectors

You can also change the colorization options by pressing **Edit Colorization** :



Query Colorization Options Dialog

This will allow you to control (in great detail) the colorization options applied to the whiskers or blotches in the plots.

Back in the main Vector Group dialog, you can also press  to get a special report on the vector query:

Vector Container Report											
<input checked="" type="radio"/> Show Delta and Magnitude <input type="radio"/> Show i, j, k, and Magnitude											
<div>Print Excel Export Done</div>											
Vector	Begin (Inches)			End (Inches)			Delta (Inches)				
Name	X	Y	Z	X	Y	Z	X	Y	Z	Mag.	
p4680	961.4760	-163.8528	150.6052	961.4739	-163.8451	150.6478	-0.0021	0.0077	0.0427	-0.0434	-
p4681	961.0664	-165.6891	149.4405	961.0653	-165.6732	149.4388	-0.0011	0.0159	-0.0017	0.0160	+
p4682	961.3082	-165.5978	150.0482	961.3078	-165.5928	150.0477	-0.0003	0.0051	-0.0005	0.0051	+
p4683	961.5474	-165.5154	150.6485	961.5473	-165.5134	150.6482	-0.0001	0.0020	-0.0002	0.0020	+
p4684	961.7079	-165.3376	151.0647	961.7331	-165.3590	151.0784	0.0251	-0.0213	0.0137	-0.0357	-
p4685	961.6644	-164.8811	151.0067	961.6828	-164.8692	151.0165	0.0184	0.0119	0.0097	-0.0240	-
p4686	961.5200	-164.2731	150.7160	961.5308	-164.2605	150.7493	0.0108	0.0127	0.0333	-0.0372	-
p4687	961.0859	-165.6782	149.5380	961.0848	-165.6617	149.5363	-0.0011	0.0165	-0.0017	0.0167	+
p4688	961.3290	-165.5901	150.1478	961.3284	-165.5809	150.1469	-0.0006	0.0091	-0.0010	0.0092	+
p4689	961.5638	-165.5032	150.7370	961.5638	-165.5029	150.7370	-0.0000	0.0003	-0.0000	0.0003	+
p4690	961.7028	-165.2971	151.1028	961.7417	-165.3203	151.1185	0.0389	-0.0232	0.0156	-0.0479	-
p4691	961.6249	-164.7921	150.9657	961.6453	-164.7785	150.9787	0.0205	0.0136	0.0130	-0.0278	-
p4692	961.4853	-164.1895	150.6871	961.4970	-164.1749	150.7279	0.0117	0.0147	0.0408	-0.0449	-
p4693	961.0807	-165.6743	149.5735	961.0796	-165.6583	149.5718	-0.0011	0.0160	-0.0017	0.0162	+
p4694	961.3228	-165.5851	150.1803	961.3223	-165.5779	150.1795	-0.0005	0.0072	-0.0008	0.0072	+
p4695	961.5592	-165.5007	150.7727	961.5591	-165.4995	150.7725	-0.0001	0.0012	-0.0001	0.0012	+
p4696	961.6069	-164.7713	150.9704	961.6286	-164.7571	150.9831	0.0217	0.0142	0.0127	-0.0289	-
p4697	961.4687	-164.1707	150.6958	961.4788	-164.1588	150.7275	0.0101	0.0119	0.0317	-0.0353	-
p4698	961.0635	-165.6746	149.5793	961.0625	-165.6588	149.5776	-0.0011	0.0157	-0.0017	0.0158	+
p4699	961.3051	-165.5843	150.1841	961.3047	-165.5787	150.1835	-0.0004	0.0056	-0.0006	0.0056	+
p4700	961.5406	-165.4982	150.7737	961.5408	-165.5006	150.7739	0.0002	-0.0023	0.0002	-0.0024	-
p4701	961.6798	-165.2924	151.1392	961.7263	-165.3213	151.1584	0.0465	-0.0289	0.0193	-0.0581	-
p4702	961.5989	-164.7837	150.9962	961.6230	-164.7685	151.0075	0.0242	0.0153	0.0113	-0.0307	-
p4703	961.4503	-164.1687	150.6981	961.4626	-164.1552	150.7316	0.0123	0.0135	0.0335	-0.0381	-
p4704	961.0342	-165.6786	149.5547	961.0332	-165.6634	149.5531	-0.0010	0.0152	-0.0016	0.0153	+
p4705	961.2782	-165.5904	150.1644	961.2777	-165.5826	150.1636	-0.0005	0.0078	-0.0008	0.0078	+
p4706	961.5148	-165.5049	150.7559	961.5147	-165.5043	150.7558	-0.0000	0.0007	-0.0001	0.0007	+
p4707	961.6628	-165.3105	151.1414	961.7078	-165.3457	151.1620	0.0450	-0.0351	0.0206	-0.0607	-
p4708	961.6041	-164.8336	151.0502	961.6333	-164.8163	151.0585	0.0292	0.0173	0.0082	-0.0349	-

Vector Query Report on the Dialog

This report will show several formats and includes the color information from the whisker plot. There is a print option as well as an excel export.

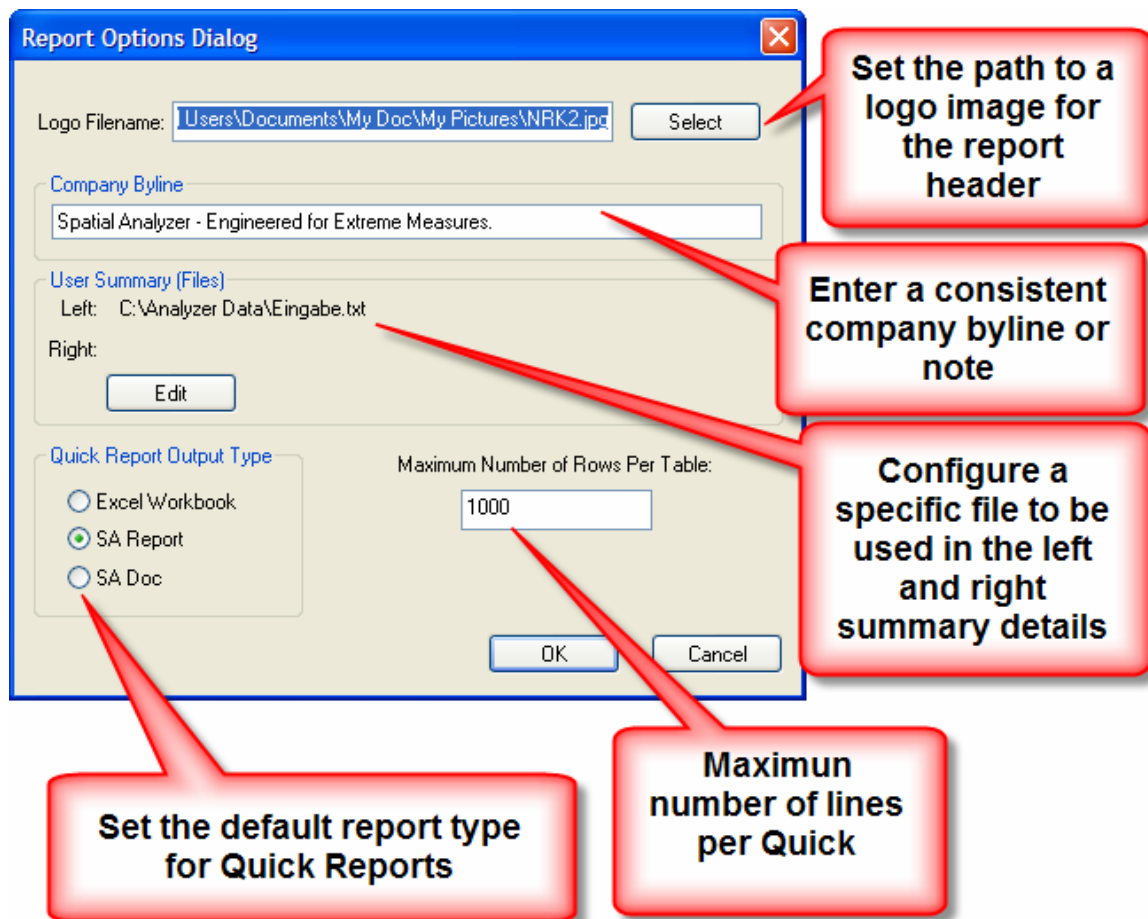
REPORTS

SpatialAnalyzer provides several different methods for viewing your coordinate measurement data. Quick Reports can be configured to create SA Documents, Excel Spreadsheet, and PDF's. Configure a Report Template with a specific sequence of reported items and graphics views. **SpatialAnalyzer** is able to output the data and combine the data from any of the Report Views into an HTML document.

SpatialAnalyzer also provides an easy way for you to generate Microsoft Office document based reports with imbedded graphics, report tables, and user notes. This capability works if you have Office XP or a later version of Microsoft's Office Suite.

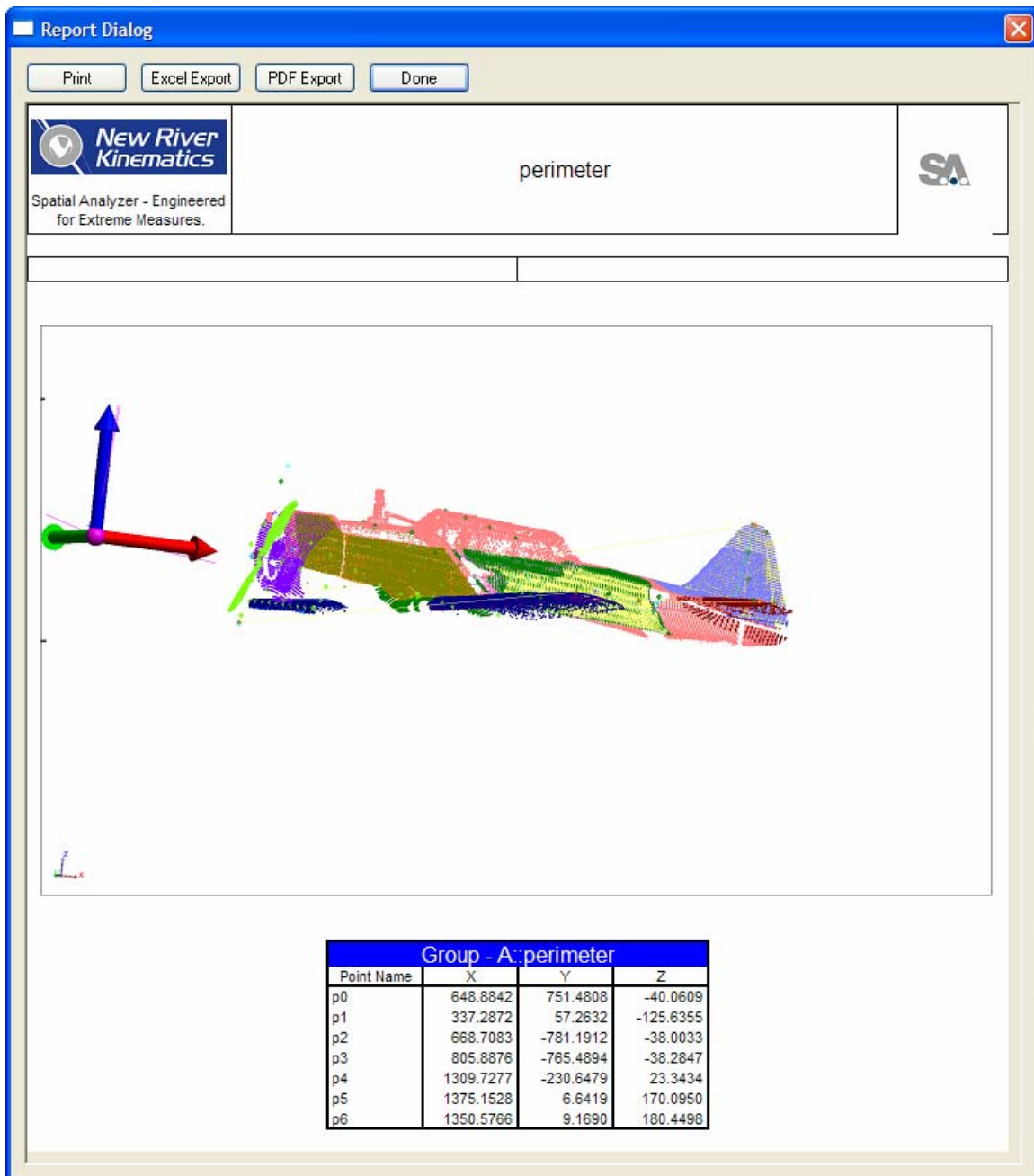
Configure Default SA Report Options

Configure the default SA report with the Report >> Option dialog. There are a number of configurable attributes, including the output type. Set this to automatically create an Excel Worksheet, SA Doc or SA Report.



The maximum number of rows per report default is 1000. Change this number to match your typical output requirements.

An example SA Report is shown below.



Example SA Report

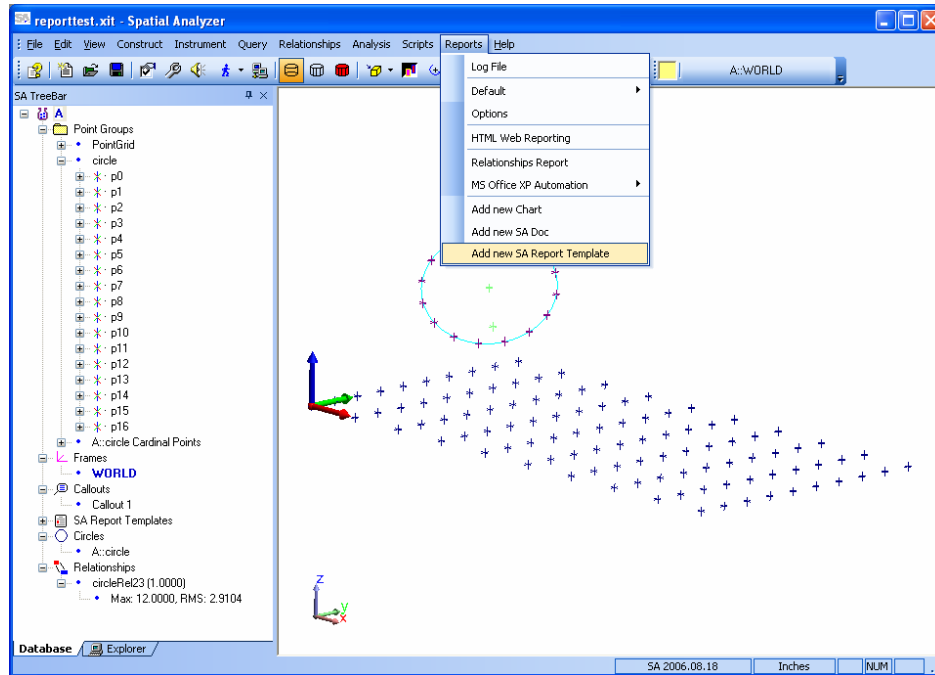
Report Templates

In previous SA versions, the report template allowed you to specify a title, user summary files, a graphical view, and a job item and these were sent to the report in a fixed order.

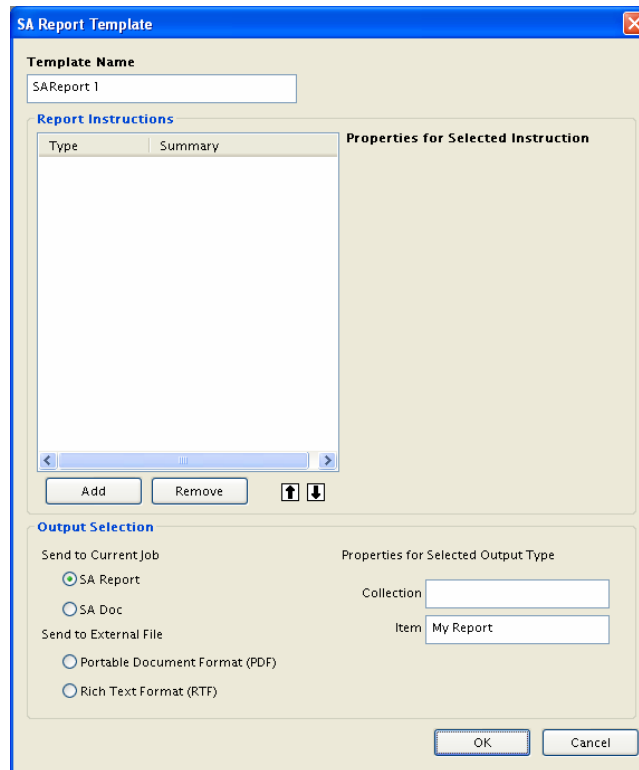
As of SA version 2006.08.18, the report template now allows you to specify multiple titles, user summary files, graphical views, and job items and control the order in which they get sent to the report.

Report Templates Walkthrough

Select Reports >> Add new SA Report Template from the menu.

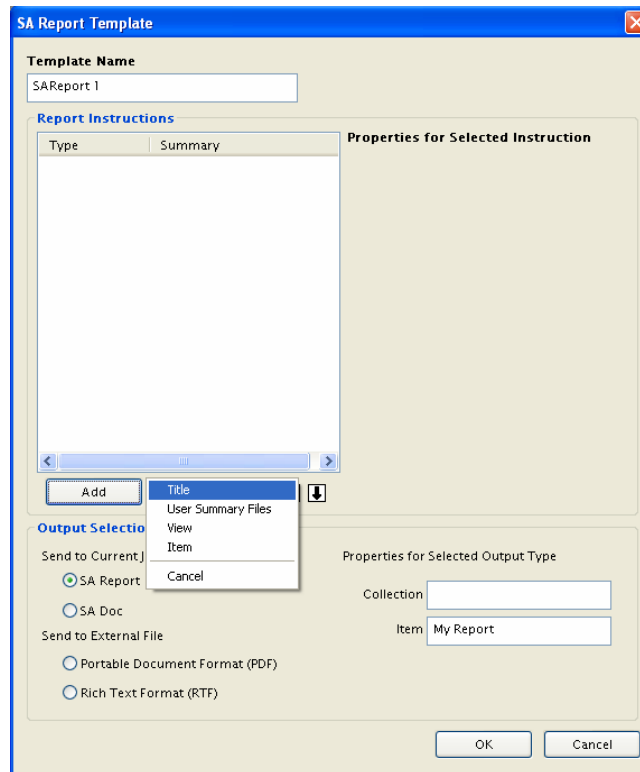


The SA Report Template window is shown with a default name, empty instruction list, and default output selection

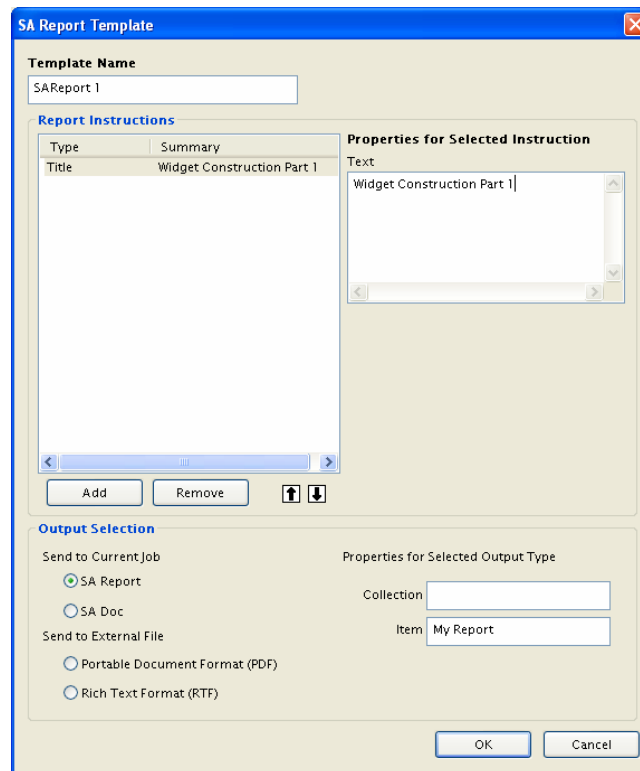


You'll need to add one or more Report Instructions to the template. Report Instructions are the graphical views, job items, title blocks, and user summary files which you want to be included in the generated report

To add a Report Instruction, click on the Add button and then select the type of instruction you would like to add from the menu which is displayed



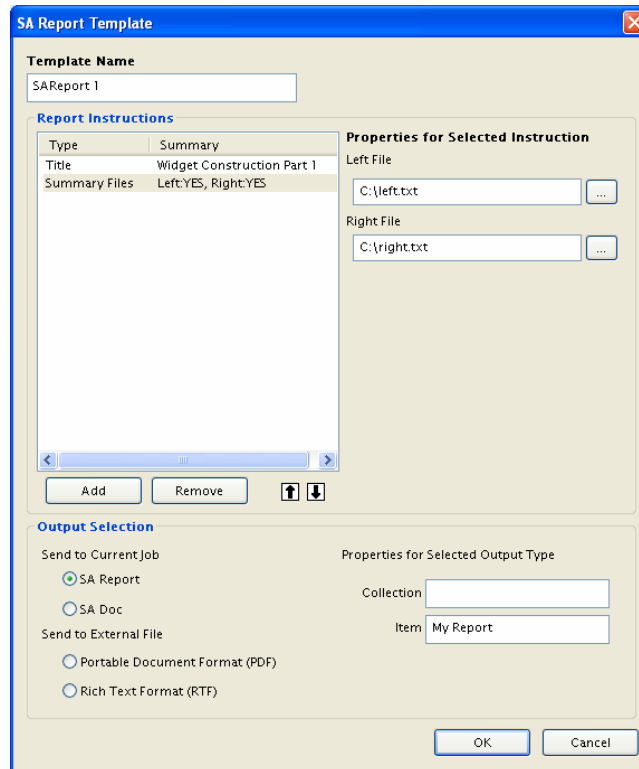
The new instruction will be added to the list and the properties for that instruction will be shown to the right of the list



With a report instruction selected in the list, you can enter the desired properties using the controls to the right of the list

The above example was for the “Title” instruction which inserts a title block into the report using the entered text

The “User Summary Files” instruction inserts the contents of two text files into the report



The “View” instruction inserts graphical views (either the current view or the specified callout view) into the report

SA Report Template

Template Name
SAReport 1

Report Instructions

Type	Summary
Title	Widget Construction Part 1
Summary Files	Left: YES, Right: YES
View	Callout: A::Callout 1

Properties for Selected Instruction

☐ Current View
☒ Callout View

Select

Collection: A

Callout View: Callout 1

Add Remove Up Down

Output Selection

Send to Current Job
☒ SA Report
☐ SA Doc

Send to External File
☐ Portable Document Format (PDF)
☐ Rich Text Format (RTF)

Properties for Selected Output Type

Collection:

Item: My Report

OK Cancel

The “Item” instruction inserts job items (objects or relationships) into the report

SA Report Template

Template Name
SAReport 1

Report Instructions

Type	Summary
Title	Widget Construction Part 1
Summary Files	Left: YES, Right: YES
View	Callout: A::Callout 1
Item	Object: A::circle

Properties for Selected Instruction

☒ Object
☐ Relationship

Select

Collection: A

Item: circle

Add Remove Up Down

Output Selection

Send to Current Job
☒ SA Report
☐ SA Doc

Send to External File
☐ Portable Document Format (PDF)
☐ Rich Text Format (RTF)

Properties for Selected Output Type

Collection:

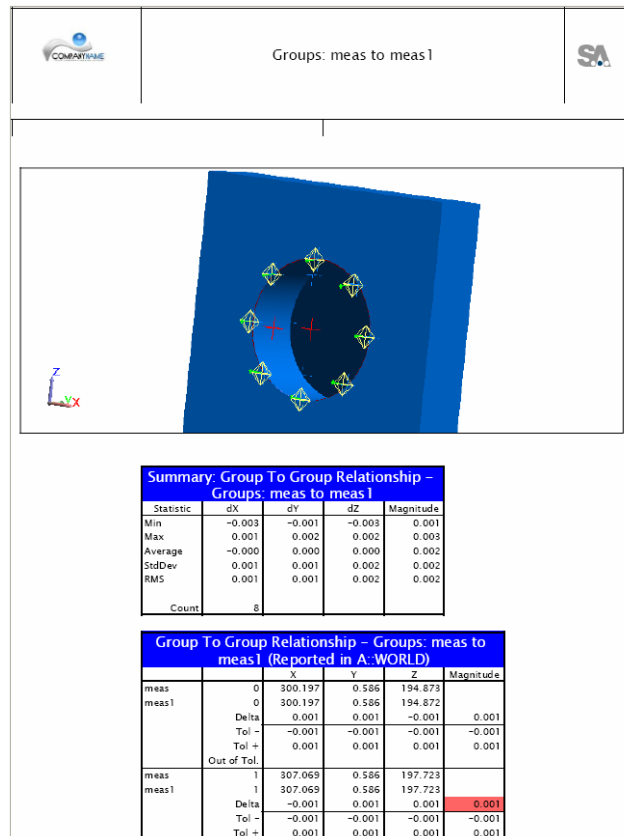
Item: My Report

OK Cancel

Add as many Report Instructions as needed for the application. The order is configurable with up and down arrows. When the report is generated, the Report Instructions will be executed in the order shown in the list. Select the desired output type and destination info and then click OK. Locate the new report template in the Job database tree, right-click on the template, and select “Generate.”

Quick Reports

SA’s Quick Report offers a quick way to create reports that include a graphical element as well as tables. For example, in the following survey, a comparison between two point groups is reported.

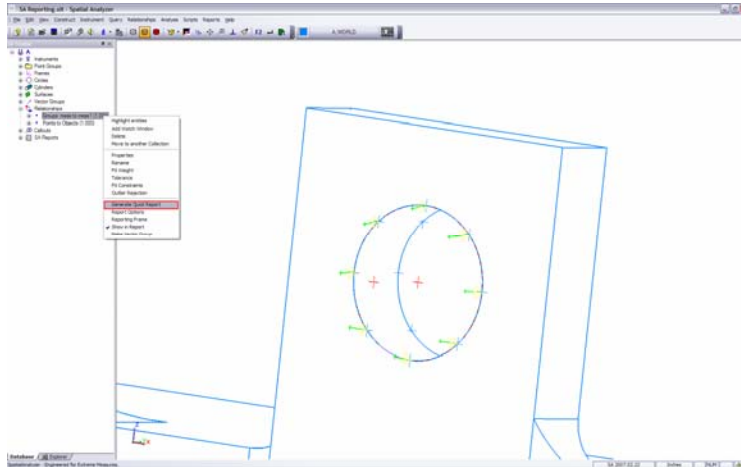


The red highlighting in the magnitude column of the second point indicates a tolerance violation. The amount in excess of tolerance is also displayed. Tolerance, ideally specified in the CAD model, may be set manually within SA. It may be asymmetric and independent for every component of every point. Tolerance may also be set generally for an entire relationship. When specified redundantly, the more specific setting overrides the less specific setting.

Here is the user interface that appears at both relationship and point levels. Additionally, point properties can be set once for any selected set of points (even across groups and collections).



Regardless of how the tolerance information was specified, or even if tolerance information was unavailable, Quick Reports are created by simply right-clicking and Generate Quick Report from the pop-up menu as shown below.



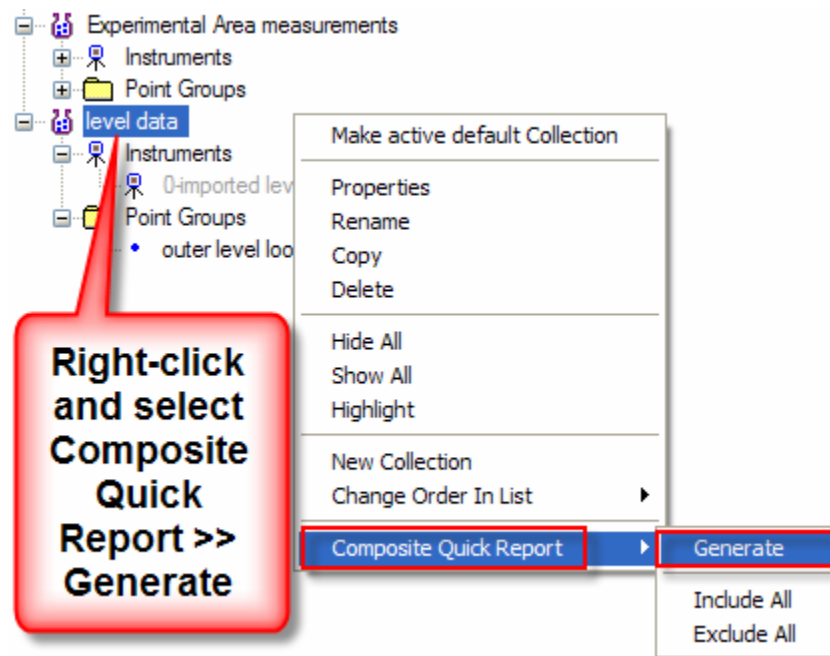
Notice the subsequent menu item, Report Options. Selecting it allows considerable customization. In Components, reporting of either or both of the points as well as the calculated difference can be suppressed. Under Format, a choice of horizontal or vertical orientation of X, Y and Z Cartesian components is provided, as well as the option to concisely report each point on just a single line. At the bottom, the tolerance information can be excluded from the report. And finally, the report can be sorted alphabetically by point name (by default data is reported in the order it was measured).



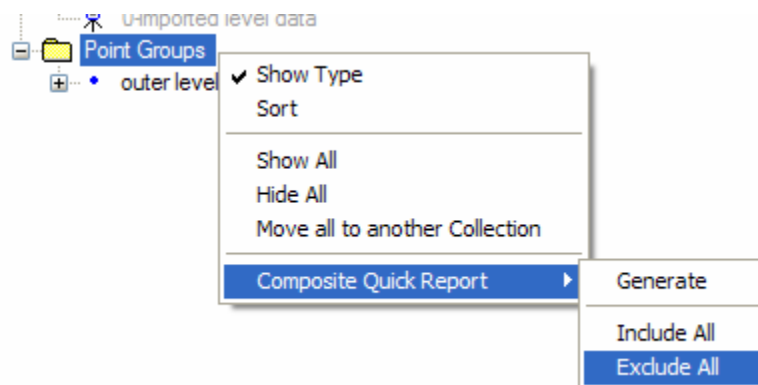
Composite Quick Reports

Making reports that include all the objects in a collection and for all the objects across collections is done by making Composite Quick Reports. The figure below shows the

Composite Quick Report selection for a collection. The same functionality is available for individual object types.



Options to Include All or to Exclude All allow control the information that is included in Composite Quick Reports. All objects are included by default. To exclude all the object of a particular type within a collection, select the object type on treeview then right-click it to show the menu. Select the Composite Quick Report >> Exclude All option to eliminated them from Composite Quick Reports. The figure below shows an example.



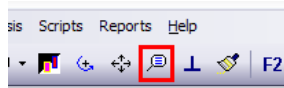
Including these objects back into future Composite Quick Reports is control using the Include All option. This mechanism works each category -- object grouping, instruments, and whole collections.

Extending Quick Reporting

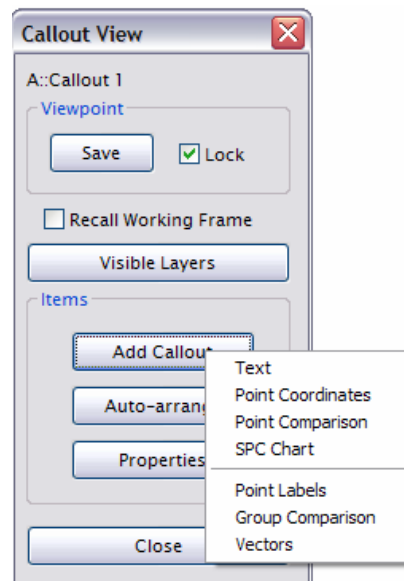
As flexible as it is, the basic premise of Quick Reports is one picture followed by one table, where the table is selected from the Treebar and the picture is the current

contents of the graphic window. What if the picture annotations in desired? What if more than one table should be reported?

Annotating the graphic window in SA is accomplished with the use of Callouts. Freeform text, point labels, coordinate values, point comparison and vector labels and values are some of the typical annotations Callouts support. Callouts are created by clicking the cartoon bubble on the Toolbar,

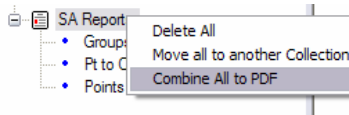


which opens the dialog below. Starting at the top, in the Viewpoint section, pressing Save causes the currently display graphics window to be displayed when the callout is displayed. Selecting Lock prevents the user from changing the viewpoint by panning, rotating or zooming. Recall Working Frame ensures that the viewer sees the same coordinate values as the callout creator (by changing working frame). The Visible layers button allows hiding and showing particular CAD layers (which can also be created in SA). Pressing the Add Callout button opens the pop-up menu facilitating various annotations. Text allow freeform labels, Point Coordinates and Point Labels support titling individual points while Point Comparison, Group Comparison and Vectors provide annotation of comparisons, such as between nominal and measured values. Auto-arrange provides a quick way to spread out the callouts for easier placement.

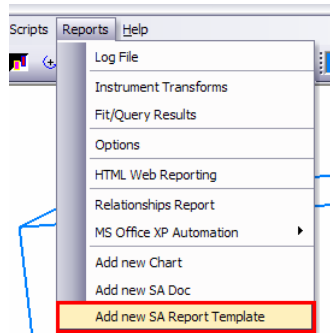


Template Reporting Example

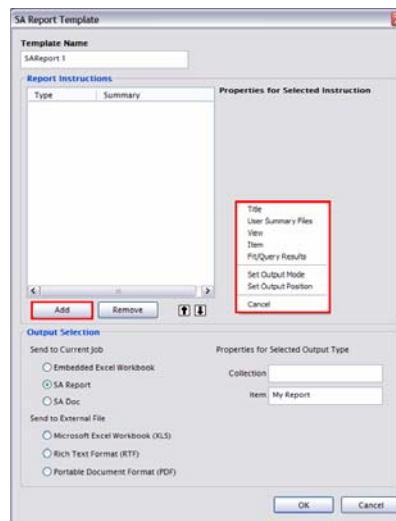
OK, so now that we have a nice graphic, how do we make a report with several tables? The simplest way is to create additional Quick Reports and the collect them all together into one document by right clicking the SA Reports group heading.



But that's not Template reporting, so let's drive on. Clicking Reports on the main menu opens up this menu branch. Select the bottom entry, Add new SA Report Template.



This will open the SA Report Template dialog below. In the middle of the window is an empty list of instructions. Pressing the Add button invokes the pop-up menu to its right, allowing us to fill it in. But first, notice that we can direct the output not only an SA Report embedded in the current job as a Quick Report does by default, but also to Excel, PDF, and Word.

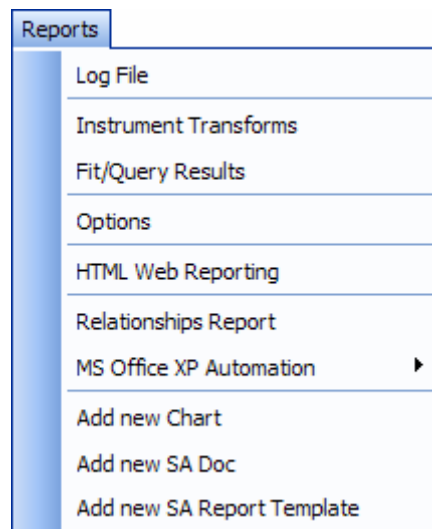


Completing the list of Report Instructions is simply a matter of choosing the parts we want from the Add menu. Title and User Summary Files create the titles and subtitle text. View defines the graphic element, and supports Callout Views as well as Views. Set Output Mode allows toggling between Insert and Replace, while Set Output Position allows targeting particular cells of an Excel spreadsheet. This is especially powerful when the output selection is an external Excel workbook while using an input Excel template file as a report basis.

Once completed, the Report Template appears in the SA Treebar. View the report by Right clicking and selecting Generate. Using Templates extends the flexibility of Quick Reports by allowing finely detailed customization of report element selection as well as output redirection.

Opening a Report View

To open a new Report View, go to the Report menu and choose form the available Report Views. This action will convert the graphics view area in **SpatialAnalyzer** into a spreadsheet Report View and populate it with your data. You can view the data in different coordinate frame types (Cartesian, Cylindrical, and Polar) and change the output decimal precision. To switch from one Report View to another, go to the Report menu, and select the new view. Selecting the "Return to Graphics" menu item takes you back into the Graphics View.

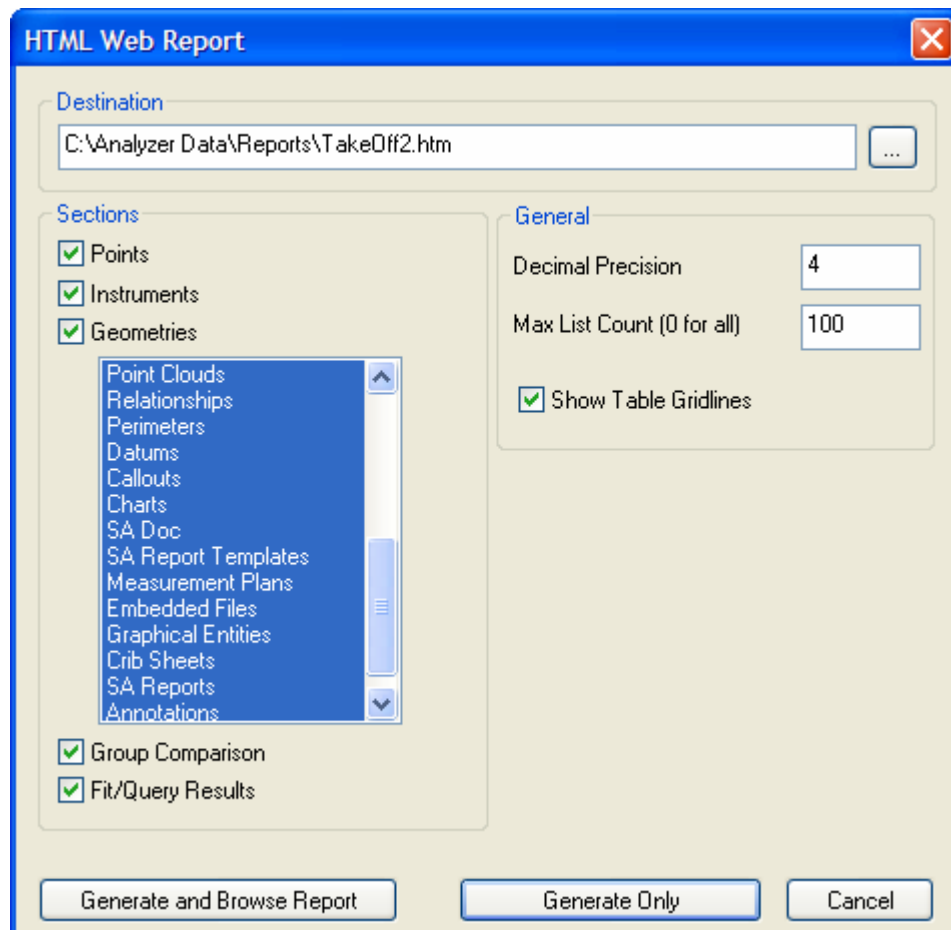


Report menu

HTML Web Reporting

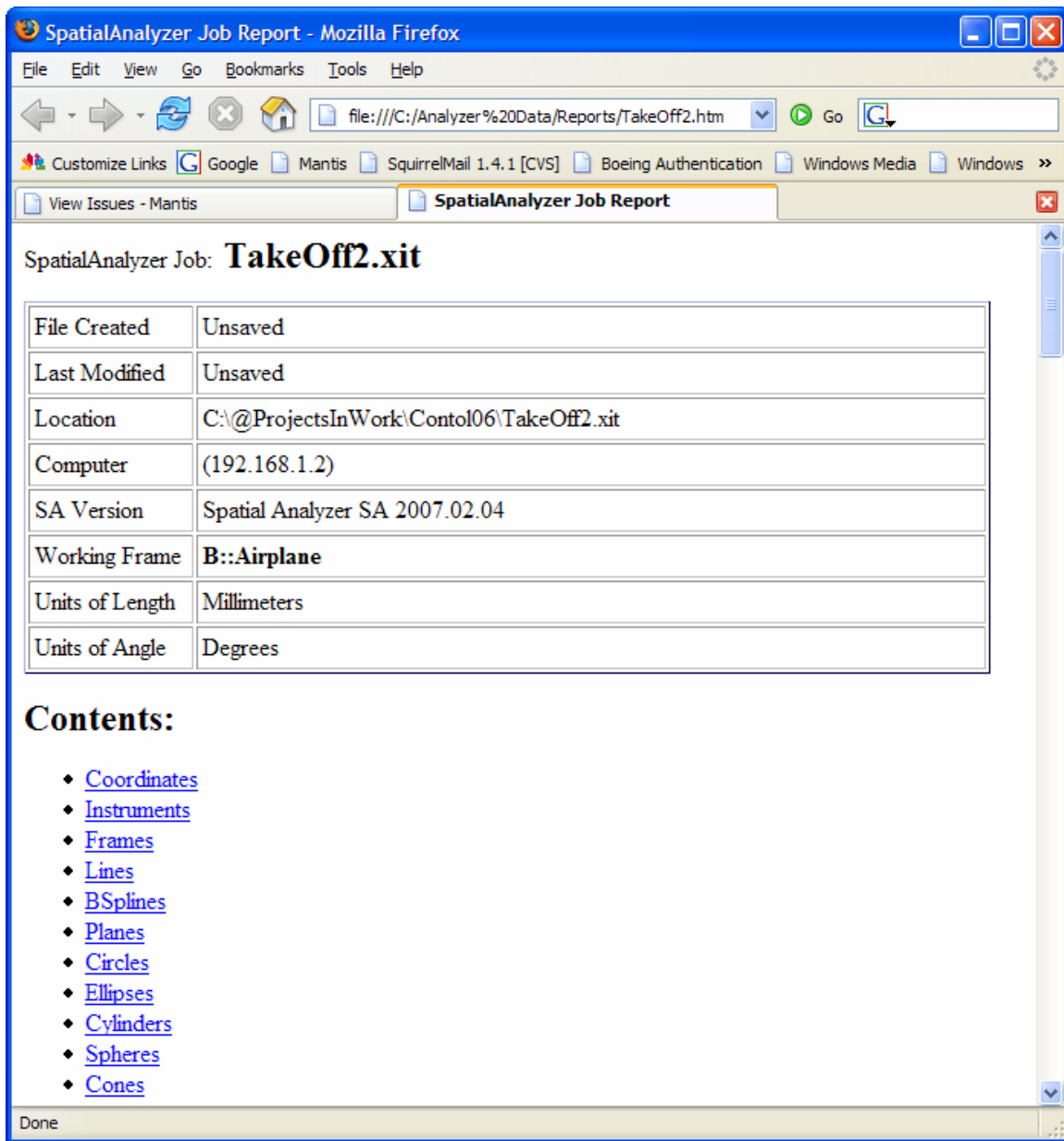
SpatialAnalyzer is able to combine and then output the Report Views into a single HTML document. You can choose which Report Views you want on the report with the HTML Web Report Dialog shown below. It enables you to print all the views into one report and makes it easy for you to share the data over the Web. The common HTML format ensures your customer will be able to open and print the document.

From the HTML Web Report Dialog name the HTML file and select the views you wish to print and hit OK.



HTML Web Reporting Dialog

If you have a default browser configured on your computer, **SpatialAnalyzer** will automatically start it and load the Report into it. This allows you a chance to preview the output before you pass it along to a customer.



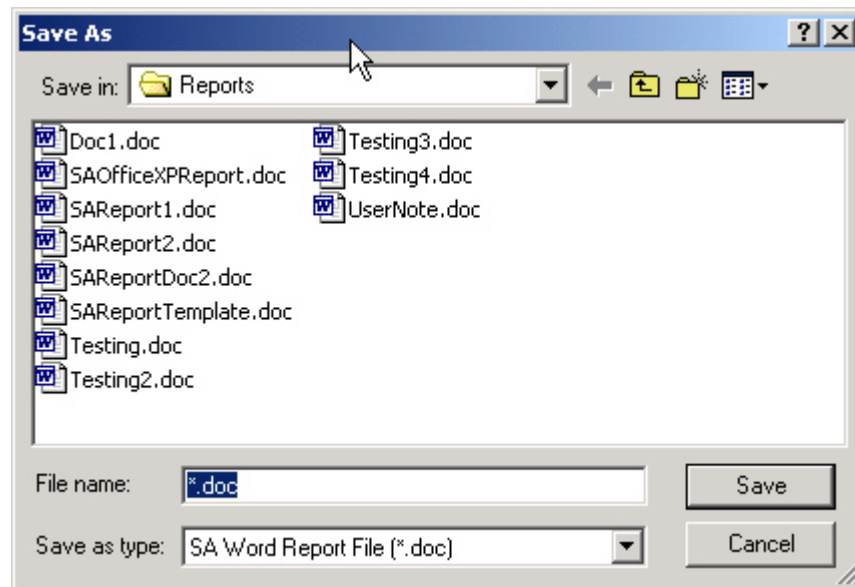
SpatialAnalyzer HTML Web Report in a Browser

Office XP Reporting

Documenting and communicating your measurement process steps and results frequently involves creating reports with images, tables of data of fit results, and notes that describe a particular setup, event, or potential problem that needs to be addressed. Microsoft's (MS) Office Suite of applications is particularly useful for accommodating these needs. To assist you in completing these responsibilities, **SpatialAnalyzer** provides a direct link into MS's Office XP suite of applications. The interface supports sending graphical views, object reports, and user notes to a Word XP document from menu choices in **SpatialAnalyzer**. MS Office XP versions (i.e., version 10+ or 2002) of Word and Excel or later are required to use this capability.

Creating a Word XP Report document from within **SpatialAnalyzer** involves three steps. Initializing the link between **SpatialAnalyzer** and MS Word is the first one. The document that is created will automatically add the SpatialAnalyzer version number and current date and time to the top of the file. The creation date and time are also added to the report's footer.

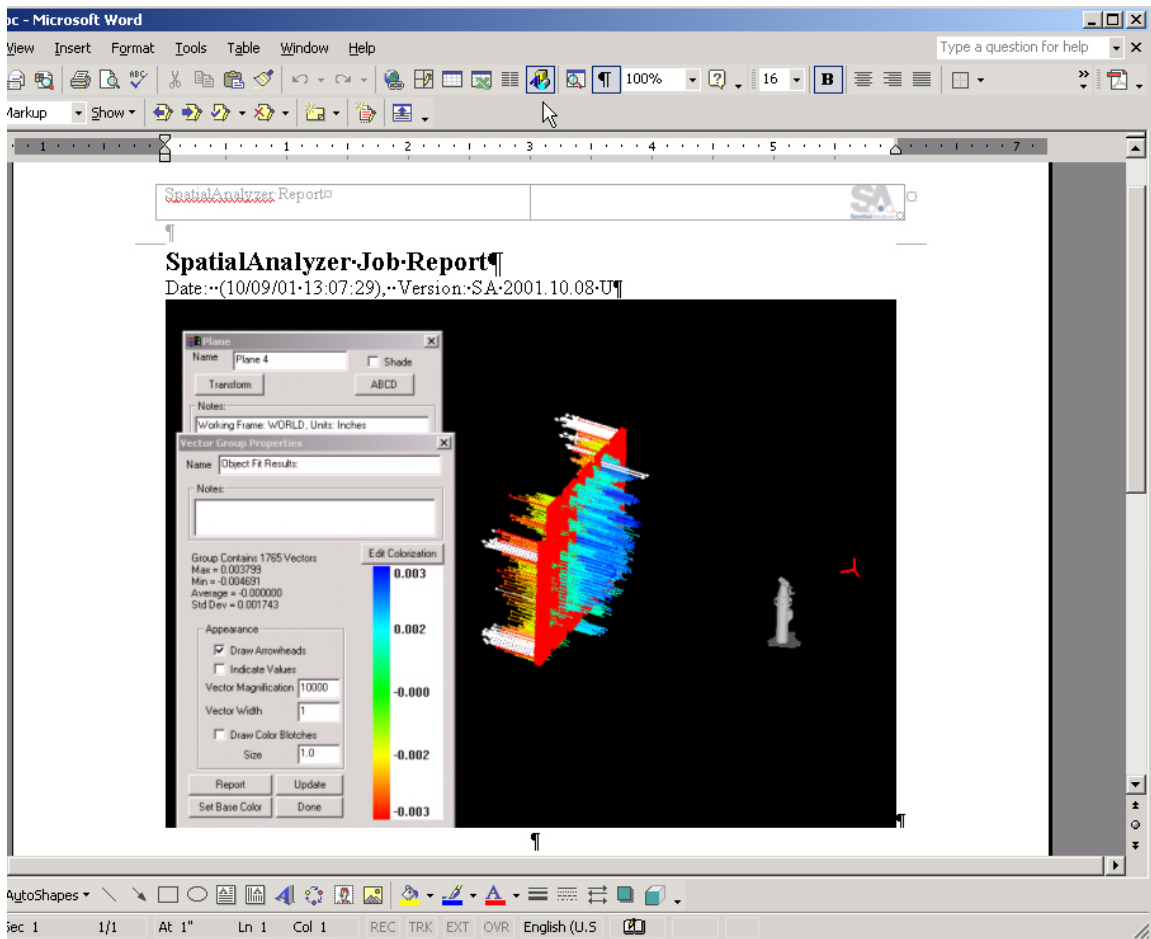
After initializing the connection you can then add graphics window views, object reports, and user notes directly into the document. Saving and Closing the Report is the final step. Each of these steps is accomplished from the Reports menu in **SpatialAnalyzer**.



File Save Dialog for Initializing an Office XP Document

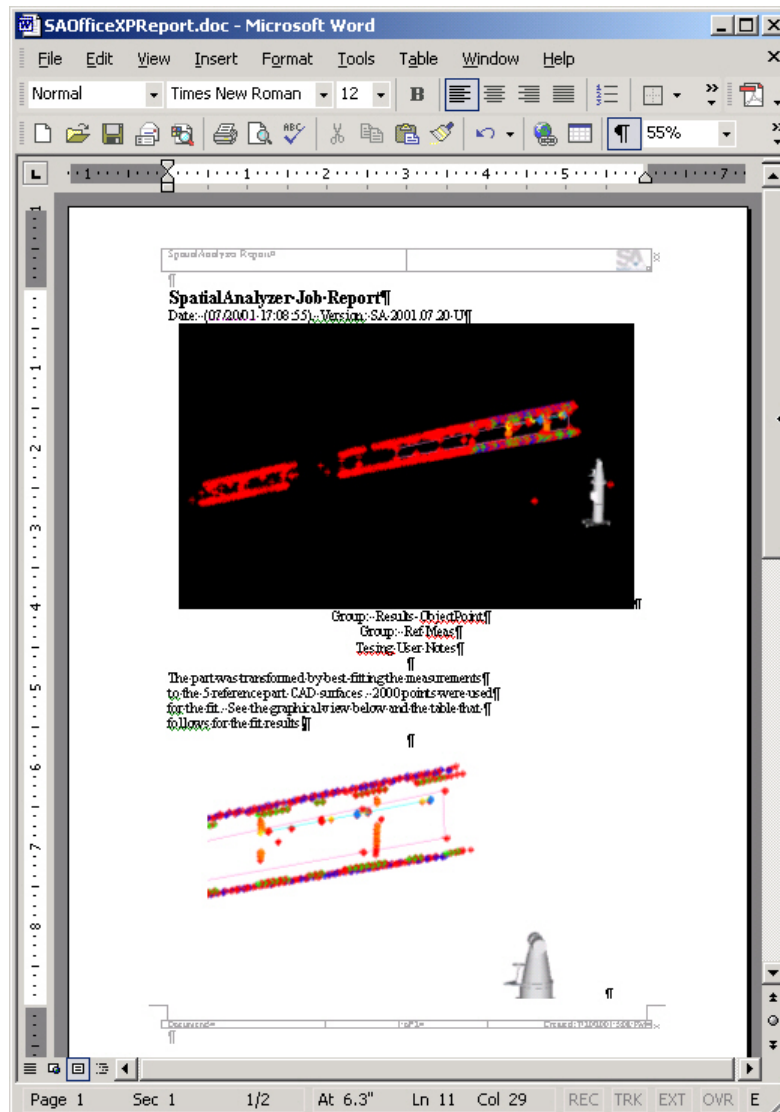
The first menu selection on the Office XP Report Automation menu is Initialize a MS Word Report. A file save dialog prompts you to identify the report document location and name. Once this procedure is completed **SpatialAnalyzer** is ready to send the images, reports, and notes of your measurement and analysis session to the MS Word document.

When you select the menu item Add graphical window from the Reports menu, a copy of **SpatialAnalyzer**'s current graphical view is placed in the MS Word document. The figures below show examples of MS Word documents with graphical windows.



Graphics View from SA in Word XP

An image of object fit results with the color scale provided on the Vector Group Properties dialog along with the instrument's relative position to the object is an effective means of communicating your measurement process and results. The figure above shows Plane Object Fit results in Word XP. It is relatively easy to understand that the 1765 measured points used to compute the best fit plane show the vertical axis bow in the real plane. The image in the report document shows the distortion effectively and presents the specific statistics (shown on the Vector Group Dialog) needed to understand the details.



Graphics View with User Notes from SA in Word XP

When communicating measurement and analysis results showing the relative orientation of the instrument to the object being measured is a good place to start. The figure below shows a MS Word report created by SpatialAnalyzer. The report contains two graphical window images and a note on how the transformation into part coordinates was accomplished. The images provide a macro view and close up view of the instruments relative of the part. When the images are combined with the note, people that were not involve in the data collection or job planning can quickly and confidently understand how the job was oriented into part coordinates. Without the images the potential for miscommunication is greater.

The next step in documenting your work is to include tables of object fit results (e.g., group vector reports, plane and line fits etc.) Selecting the 'Add object report' option from the MS Office XP Automation menu will prompt you to select objects from the job. You can select objects graphically or with the F2 selection tool. The fit results for each

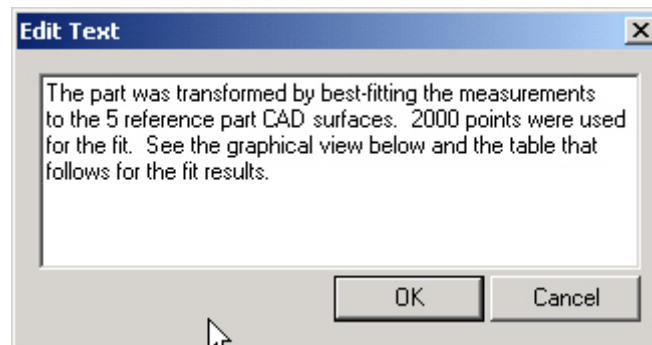
object are then formatted into a table and sent to the report document. The figure below shows an open report document in MS Word with a 'Points to Surface Vector Group' report in a formatted table.

The screenshot shows a Microsoft Word window titled 'SARReportDoc2.doc - Microsoft Word'. The document contains a table with the following data:

Name	X	Y	Z	dx	dy	dz
Final::fincut1	-26.2518	390.7615	-52.4245	.0009	.0000	-.0855
Final::fincut2	7.6537	381.1525	-52.0479	.0010	.0000	-.0897
Final::fincut3	38.9722	371.6244	-51.7000	.0010	.0000	-.0929
Final::fincut4	47.5798	349.1448	-51.6032	.0010	.0000	-.0939
Final::fincut5	20.2816	349.0554	-51.9059	.0011	.0000	-.0947
Final::fincut6	-1.7811	349.9496	-52.1506	.0010	.0000	-.0916
Final::fincut7	-21.8114	346.7640	-52.3726	.0010	.0000	-.0908
Final::fincut8	-20.1043	327.1871	-52.3524	.0010	.0000	-.0901
Final::fincut9	4.5118	328.2944	-52.0795	.0010	.0000	-.0913
Final::fincut10	28.9528	331.2424	-51.8086	.0010	.0000	-.0934
Final::fincut11	49.1103	330.6555	-51.5851	.0010	.0000	-.0931
Final::fincut12	50.7878	309.0110	-51.5651	.0010	.0000	-.0943
Final::fincut13	19.8188	308.4869	-51.9085	.0010	.0000	-.0905
Final::fincut14	-3.4481	307.0126	-52.1665	.0010	.0000	-.0901
Final::fincut15	-16.4993	304.1471	-52.3110	.0010	.0000	-.0895
Final::fincut16	-14.9240	284.5583	-52.2923	.0010	.0000	-.0904

Object Report Tables from SA in Word XP

Adding user note to your report is accomplished with the Add user notes option on the MS Office XP Reports menu. An Edit Text dialog allows you to make detailed comments concerning the measurement process. These are helpful when used in combination with object reports and graphical window images.



User Notes Input Dialog for Word XP

SpatialAnalyzer's Office XP interface provides an easy to use integrated reporting mechanism that can make the job of communicating you results and process to other more effective. This document considers recent enhancements to SA's Quick Reporting mechanism, including tolerancing and formatting options. Also reviewed are Quick Reporting's default report settings and how to customize the report using Template Reporting.

CHARTING, TREND ANALYSIS AND SPC

Charting measurement and analysis results is an important tool when communicating results to customers. When trends in data are presented in clear and concise charts the time spent understanding the data is more productive. Looking at specific attributes over time in charts shows process variation. Combining process variation charts with traditional Statistical Process Control (SPC) analysis helps us separate expected variation from changes that indicate intervention is necessary to bring a process under control. The statistics also help us refine cause vs. effect details which can lead to better decisions.

SA has several integrated charts that make the effort of generating these 2D analysis views of the data relatively straight-forward. Each of the charts types can be created and then updated from a vector group's property dialog. Once created charts and their properties are also accessible from the treeview. Outputting charts is supported with print functions and exports of jpg graphics files.

The first place to start is with the different types of charts supported by SA.

Chart Types

Effectively charting information can take different forms depending on what needs to be communicated. To support analysis and findings typically done with metrology data SA integrated a series of different chart types. Currently chart types include Individual X – Moving Range SPC, Bullseye, and Run charts. When creating a chart, one of the first steps is to select the chart type. The selection dialog for chart types is shown in the figure below.

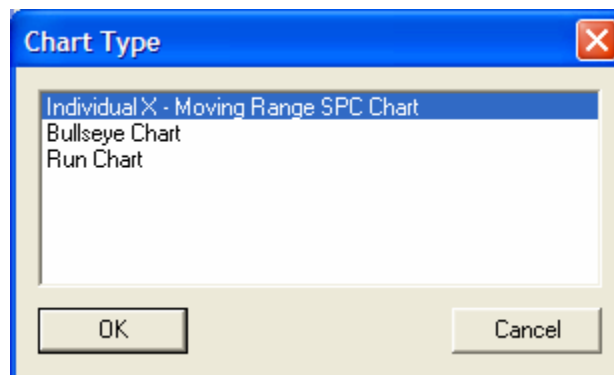
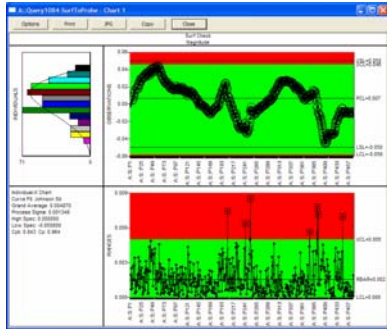
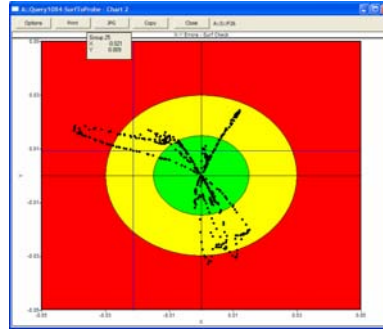


Chart Type Selection Dialog

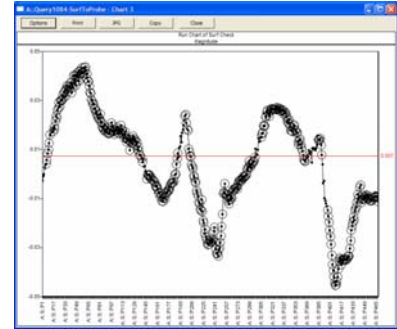
Examples of each chart type are shown in the following figures.



**Individual X –
Moving
Range SPC
Chart**



**Bullseye
Chart**



Run Chart

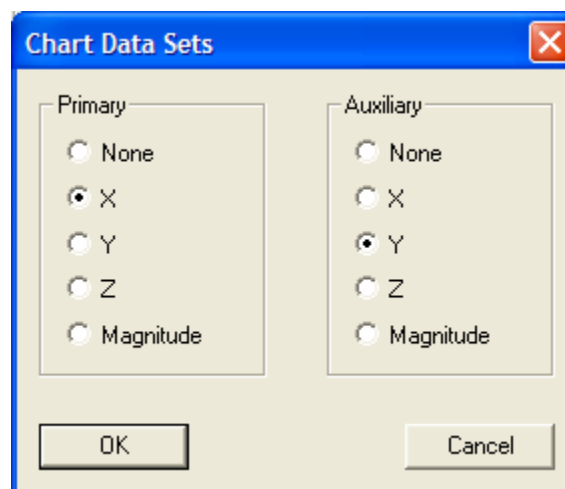
Individual X – Moving Range SPC Chart

Individual-X / Moving Range charts are generally used to monitor actual observations. These charts are efficient at detecting relatively large shifts in the process average, typically shifts of ± 3 sigma or larger. Each subgroup, consisting of a single observation, represents a "snapshot" of the process at a given point in time. The charts' x-axes are based on how the data was sampled (e.g., spatial or temporal).

The charts effectively show process history based on how the data is ordered. In other words it is presented in the sequence that the vector group was created with.

Bullseye Chart

Bullseye Charts are an excellent method to look at vector group data in a 2D projection. This chart type allows you to select a component direction for each axis of the chart. The error vectors are relative to the working frame. The figure below shows the selection dialog for choosing a chart's primary and auxiliary axial components.



Error Vector Component Selection

A good application for this chart type is when deviations from a nominal line or cylinder need to be presented along the object's normal direction. Create a frame on the object and clock it so that the X-Y plane is oriented relative to the datum frame. With the new frame as the working frame create a Bulleye Chart with the X and Y error components as the charts primary and auxiliary axes. The result is an easy to understand view of the error along the line.

Run Chart

Run Charts are employed to visually represent data. They are used to monitor a process to see whether or not the long range average is changing. Run charts are the simplest tool to construct and use. Points are plotted on the graph in the order in which they become available.

Making a Chart from a Vector Group

Creating charts is frequently done by using the options on Vector Group properties dialogs. Using an example to explain the process for creating a specific chart is a effective way to communicate the details. The figure below shows a vector group comparing measured data against a nominal CAD model.

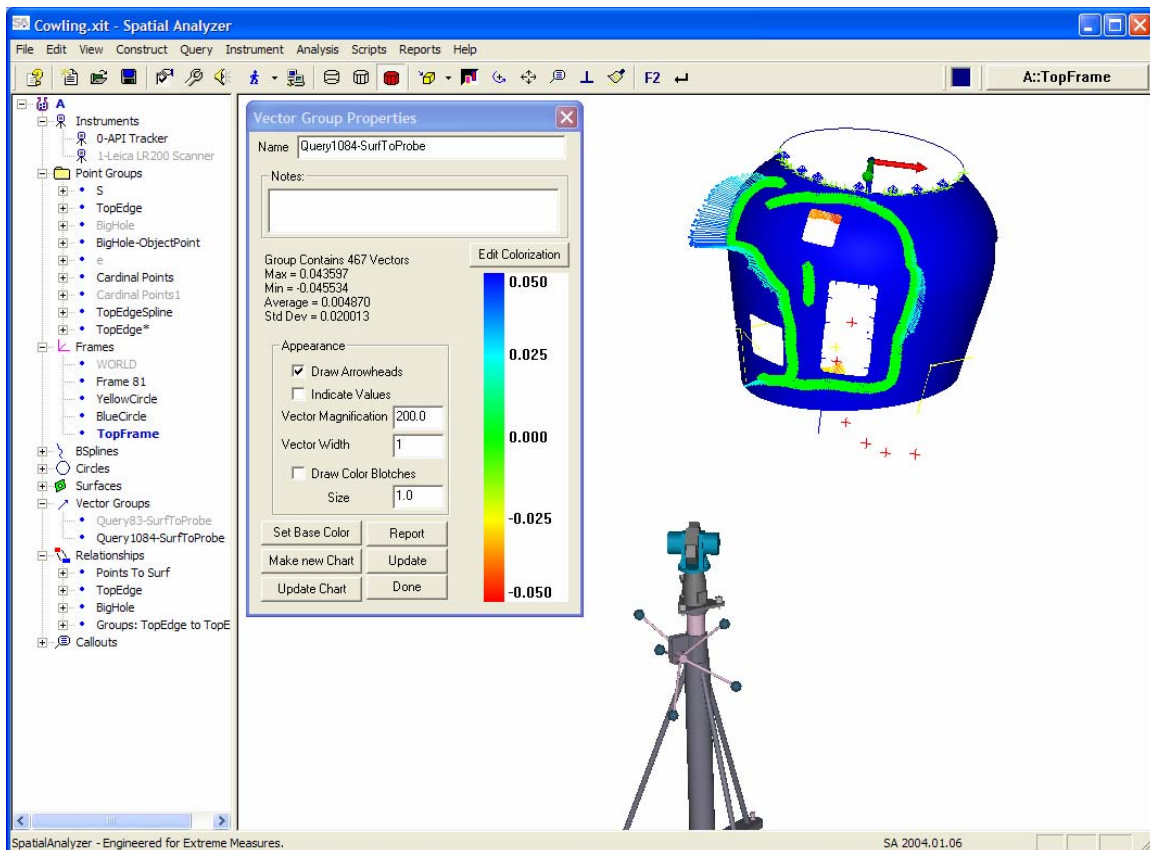


Chart example SA Job

The function 'Make new Chart' supports creating charts from the analysis results in the vector group. The figure below shows a Vector Group properties dialog with the **SpatialAnalyzer User Manual**

functions for working with charts. Select the 'Make new Chart' button to begin the process.

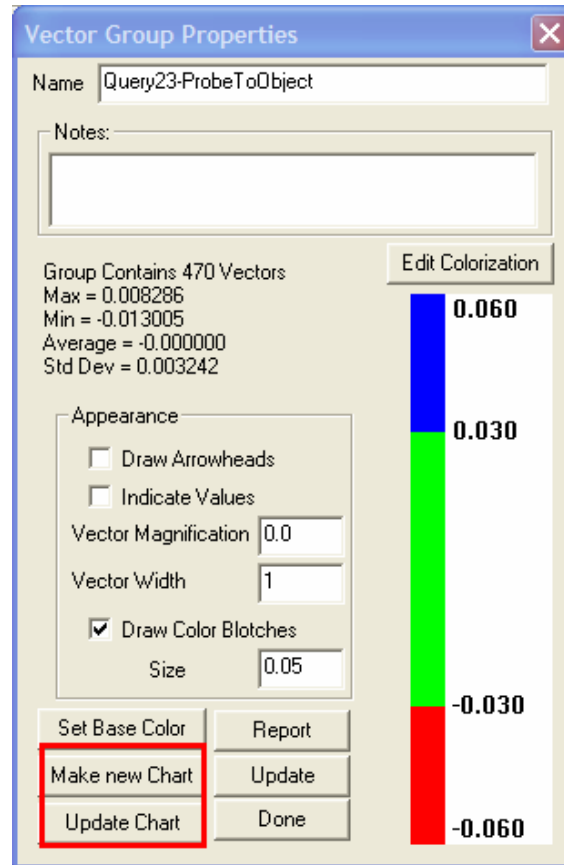


Chart Buttons on Vector Group Properties Dialog

Selecting the chart type and then the primary and auxiliary axes for the chart are the steps required to create a chart. Those selection dialogs are shown below.

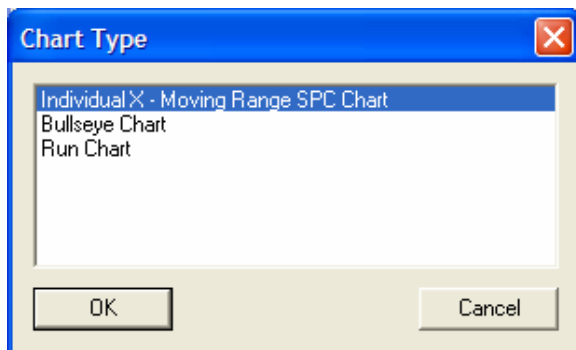
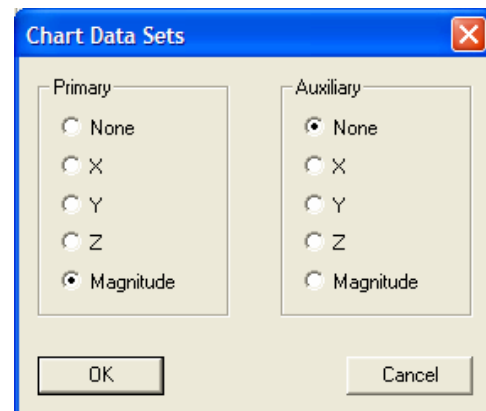


Chart Type Selection Dialog

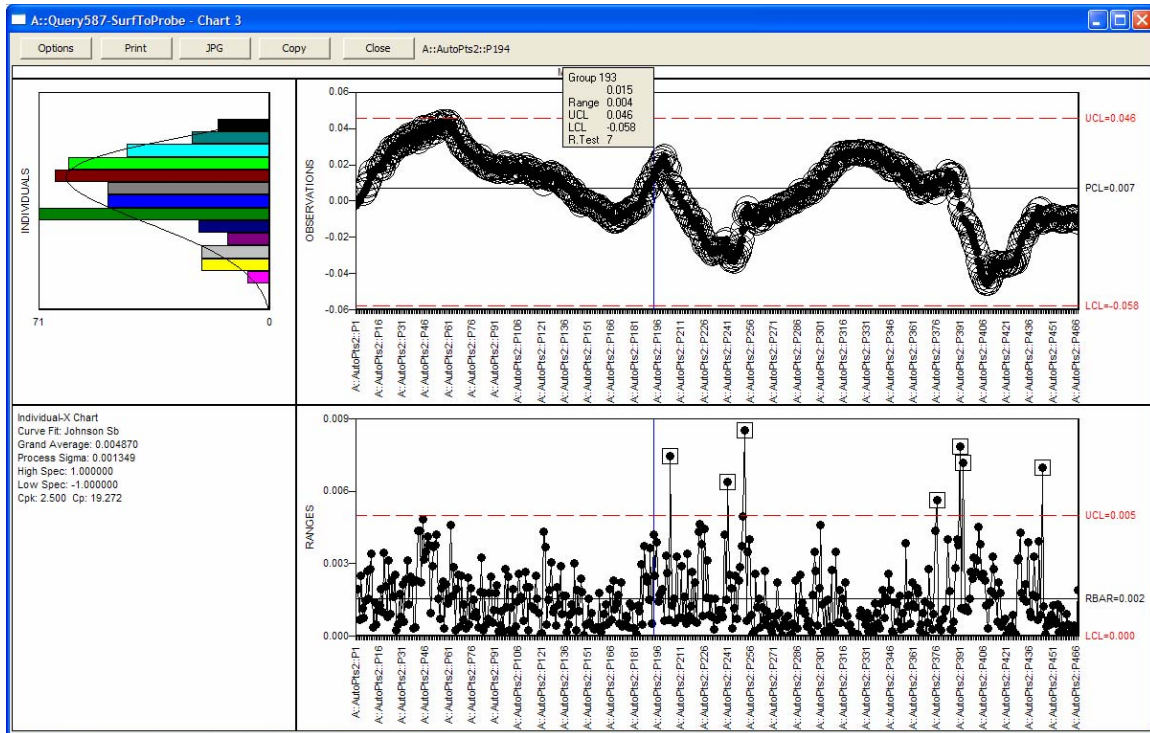


Vector Component

Selection

In this application an Individual X – Moving Range SPC chart was selected and the primary axis was set to show the error magnitude. An auxiliary axis is not required with the Individual X – Moving Range SPC chart. The resulting chart is shown below.

Charting the comparison helps communicate the differences between the part and its CAD definition. Using the 2D chart with the 3D graphics view in concert makes process of communicating the actual vs. nominal surface configuration more effective.



Individual X – Moving Range SPC Chart example

Chart Options

Each chart type has a set of attribute properties that can be configured through a properties dialog. An example of the charts properties dialog is shown below.

Individual X – Moving Range SPC Chart Options Dialog

Set the charts name in the field at the top. Charts each have two lines for title information. The specification limits control where on the chart these limits are shown. Control limits e.g., Upper and Lower Control Limits, are computed with the following formulas.

$$UCL_x = \bar{x} + 3\sigma_x \text{ (Normal Distribution)}$$

$$LCL_x = \bar{x} - 3\sigma_x \text{ (Normal Distribution)}$$

Where x-bar is the average and sigma-x is the process sigma. The 3 in this example is the ordinate value shown on the properties dialog.

Outputting Charts

Charts can be printed with a common printer control or saved into jpg files for use in other reports. Charts are also typically copied to the clipboard with the Copy button and then pasted into SA documents. Use the print, JPG, and Copy buttons on the chart dialog to output the chart into the form and format needed for your reports.

Options for Chart Outputs

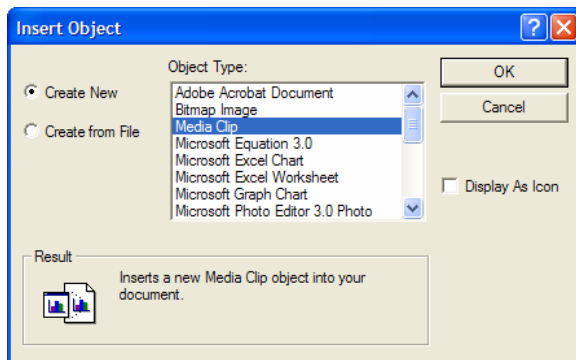
SA DOCUMENTS

SpatialAnalyzer provides a way to create, manage and store documents called SADoc's by embedding the ever present Window Wordpad functionality. Reporting and saving notes about the job are easily supported with this feature.

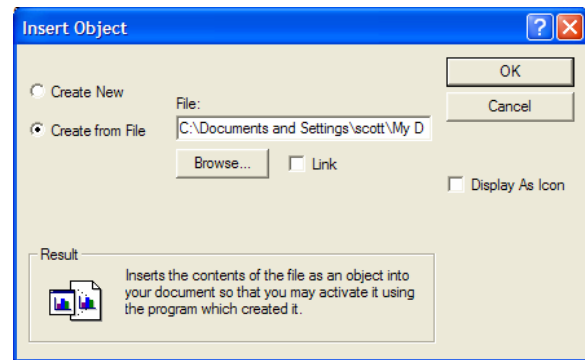
With a SADoc you can insert any object type that supports OLE server access. That includes images, spreadsheets, charts, columns of data, and PowerPoint presentations etc.

Inserting Objects into SADocs

Inserting objects into SA Doc is available on the SADoc menu. Choose the Insert >> Object menu option and you will be shown a dialog like that shown below. The dialog on the left create a new object of a particular type. The selection (on the right) to Create from File helps by letting you browse to the file and insert it based on the file names extension (e.g., .jpg = image, .xls = worksheet, .rtf = Rich Text Format.)



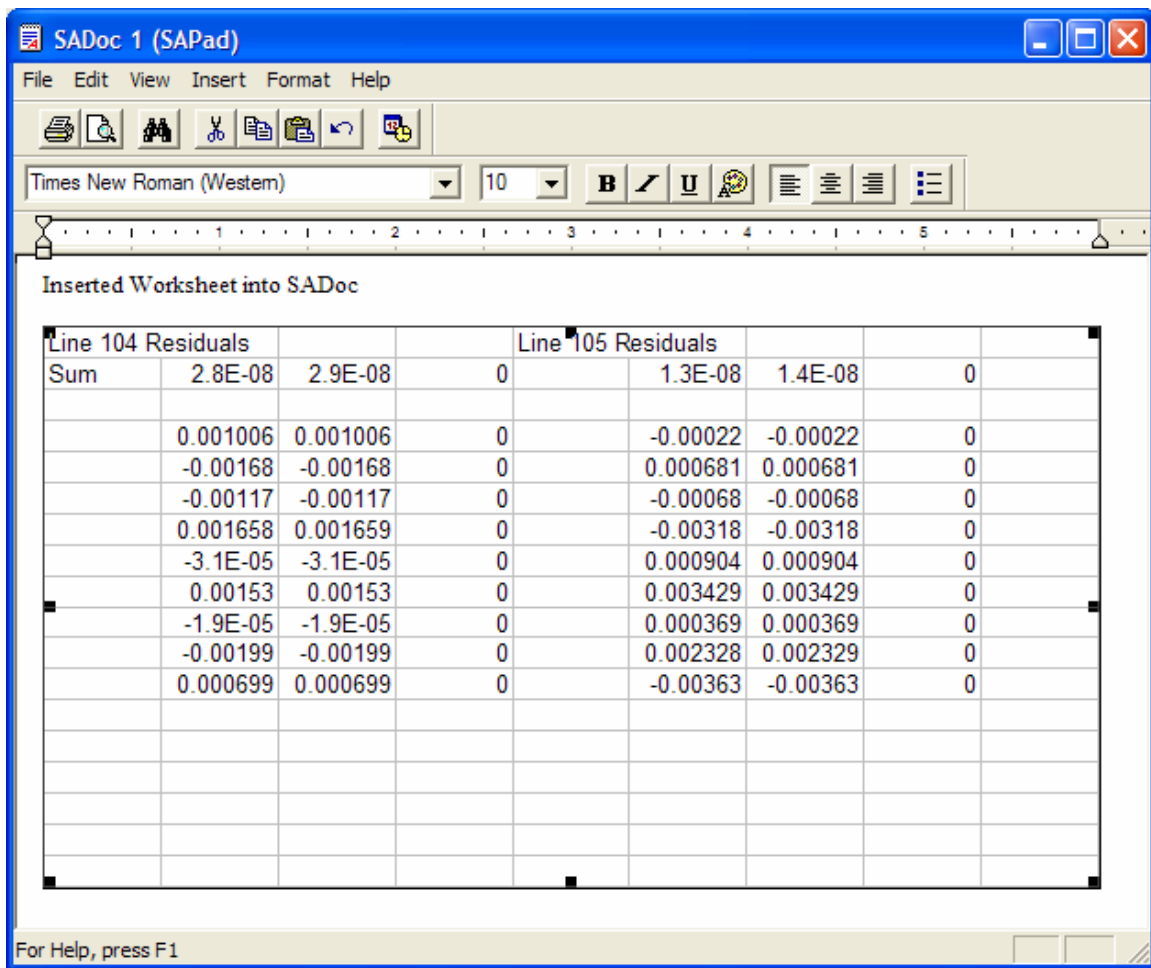
Insert New Object



Insert Object from a File

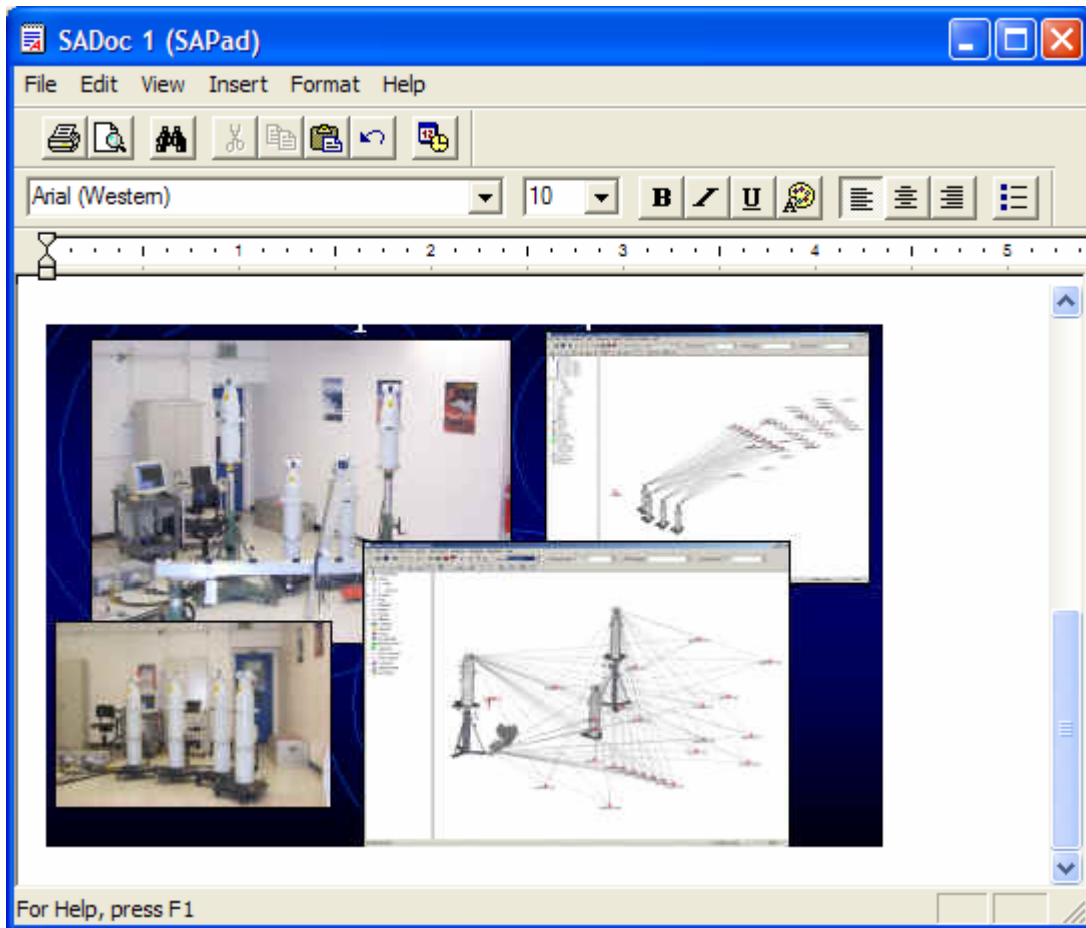
The objects that your system will be able to insert depend on the OLE complainant software that is loaded on the computer. For example all of the components that come with Microsoft Office are able to offer object that can be inserted into SADocs.

As an example of Create New Object, the figure below show a worksheet inserted into the SADoc. The worksheet has a column of data copied and pasted. A simple Sum function was then added in the worksheet.



SADoc Example with a Worksheet

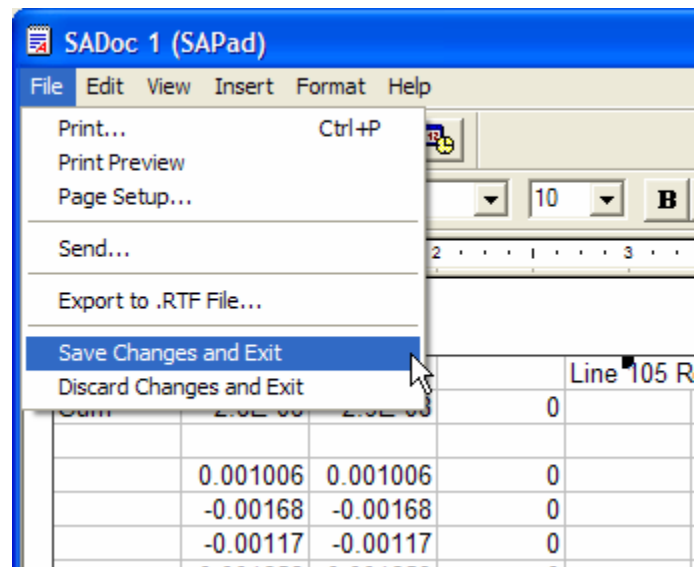
As an example of Create from File, the figure below show a simple image insert from a file.



SADoc Example with an Image Inserted

Saving SADocs

Saving SADoc changes back into the SA job file is done by selecting the File >> Save Changes and Exit menu item. The changes need to be posted with this process for SA to retain the differences. The figure below shows the highlighted menu option.



Saving SADoc changes and exiting

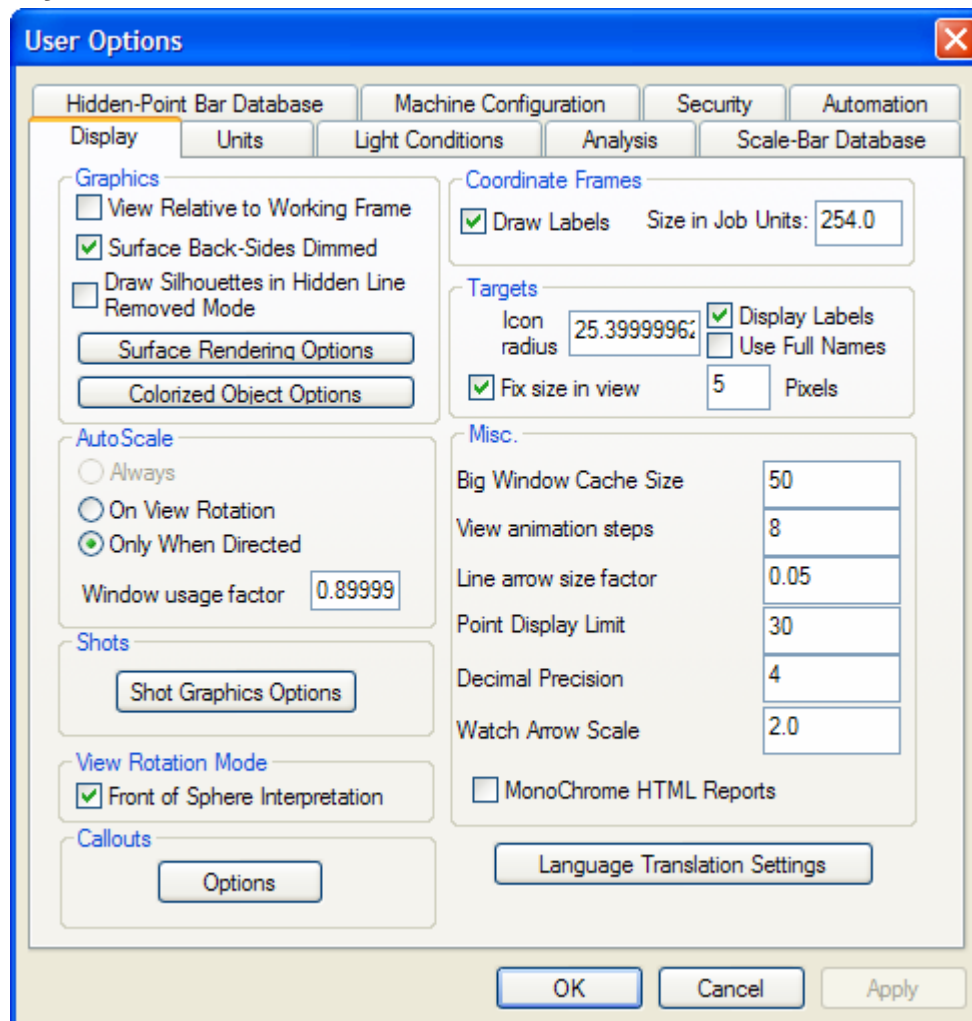
If you need to discard the changes either select the next menu option or simply close the SADoc without saving changes.

USER OPTIONS

The User Options dialog in **SpatialAnalyzer** contains many tabs of settings. All settings are located within this area to organize them as well as to provide the user with a common location for all settings. This chapter will describe the options contained on each tab.

All of the options on these tabs are saved with **SpatialAnalyzer** job file. If you have settings that you would like to use for all jobs you perform, just open a new **SpatialAnalyzer** file, configure the options the way you like them, and save the job file. This file then may be used as a template for other jobs. Just open the file and rename it before beginning a new job.

Display



Display Options

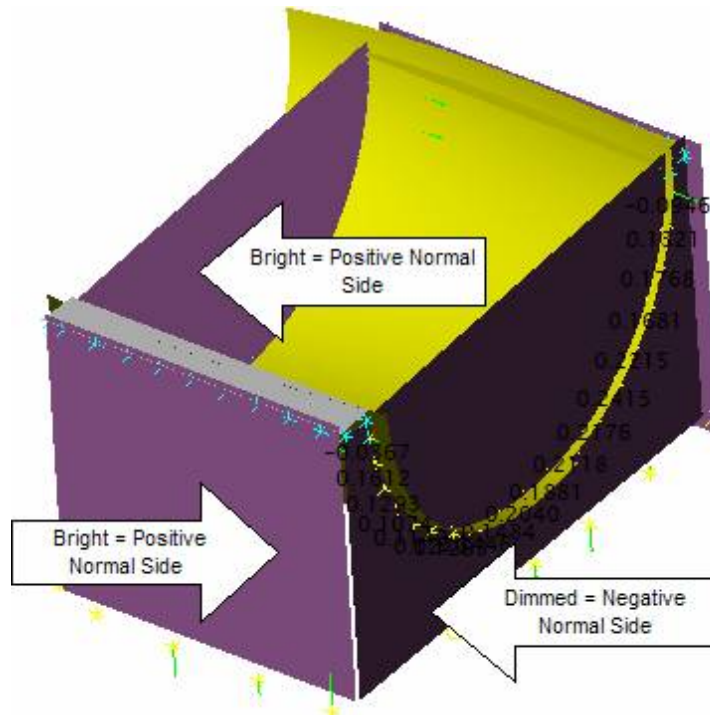
This tab primarily deals with parameters controlling the look and feel of **SpatialAnalyzer**. Each setting is discussed below.

Graphics Mode

The combo box in this area allows you to set the graphics display mode. These options are the same as those represented on the toolbar. In addition, there is a check box to set a white or black background.

Surfaces Back-Sides Dimmed

SpatialAnalyzer shows the sense of the surface's normal direction by presenting the positive side of the surface as Bright and the negative side Dim. There is an example shown in the figure below.



Surface Back-Sides Dimmed

This behavior of showing the positive and negative sense of surface as bright and dim (respectively) can be turned off in the User Options dialog. By turning off the Display option for showing Surface Back-Sides Dimmed all of the surfaces are then shown with the Bright color.

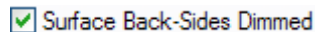


Figure 0-1 User Options >> Display Tab for Surface Back-Sides Dimmed

Autoscale Settings

You can autoscale whenever the view is rotated or only when the button is pressed. In addition, there is an autoscale factor. This factor determines how the image is scaled to fit the window. If the factor is 1.0, the limiting dimension (x or y) will be scaled so that it uses the entire window.

Big Window Cache Size

This setting controls the number of frames of the big data window that are stored in memory for recall. This pertains mostly to theodolite measurement jobs.

View Animation Steps

Whenever you change to a preset view, **SpatialAnalyzer** animates from the current view to the destination view. This is done to help keep the user from getting disoriented. This setting controls the number of intermediate steps that are taken between the current and destination view.

Line Arrow Size Factor

This is the multiplier used to determine the size of the line arrowhead relative to the length of the line. In the line geometry properties, you can control whether the arrow is drawn at the end of the line.

Shots Settings

Default Shot Length controls the length of the line used to represent measurements taken from a theodolite device. The check box for Draw Only Last Shot is used to remove the clutter that may be present in the view when a job contains a large collection of measurements.

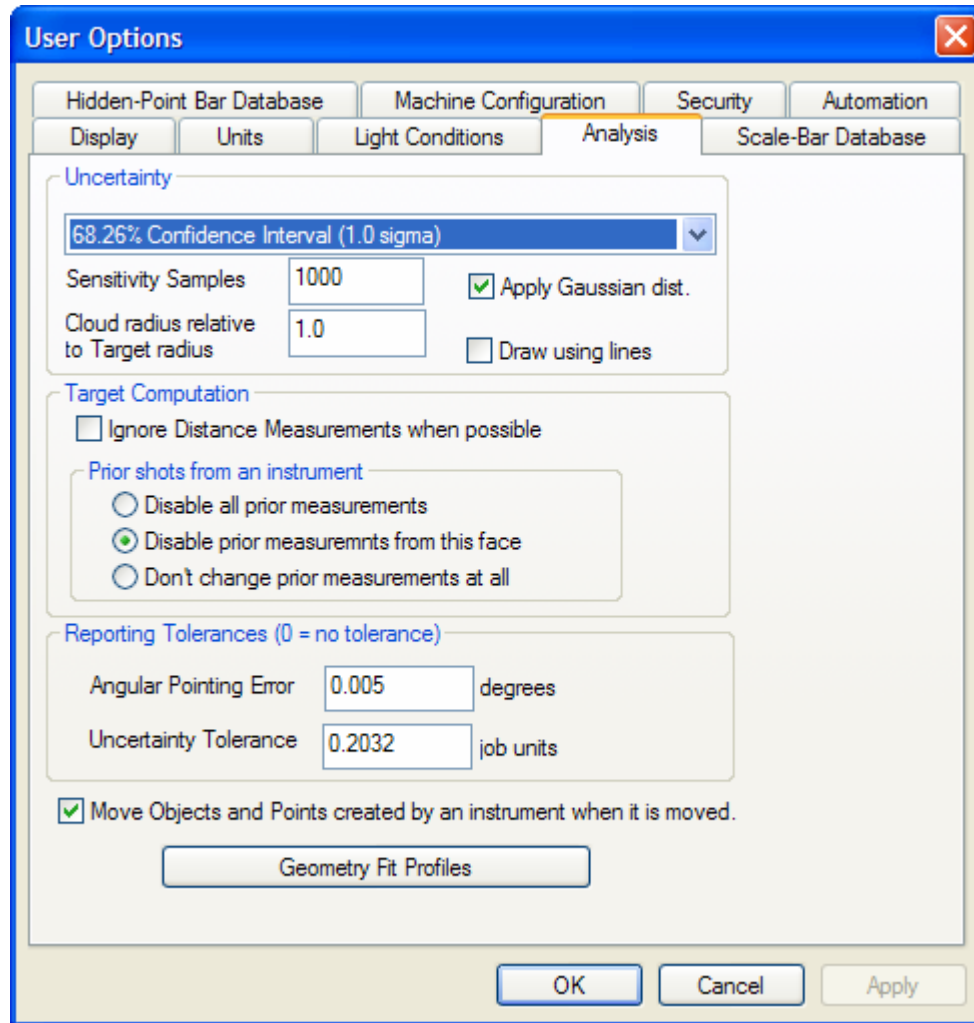
Coordinate Frame Settings

In this area, you can control whether the coordinate frames are labeled when they are displayed. In addition, you can hide or show all coordinate frames.

Target (or Point) Settings

The icon radius controls the size of the point symbol in job units. Below this field is a check box that will fix the size of the icon in the view. In this case, the field immediately to the right will control the number of pixels used. In addition, there are check boxes to determine how points are labeled in the graphical window.

Analysis



The image shows a 'User Options' dialog box with the 'Analysis' tab selected. The dialog has a blue title bar and a close button in the top right. It contains several sections: 'Uncertainty' with a dropdown menu set to '68.26% Confidence Interval (1.0 sigma)', 'Sensitivity Samples' set to '1000', 'Cloud radius relative to Target radius' set to '1.0', and checkboxes for 'Apply Gaussian dist.' (checked) and 'Draw using lines' (unchecked). 'Target Computation' includes an unchecked checkbox for 'Ignore Distance Measurements when possible' and a group box for 'Prior shots from an instrument' with three radio buttons: 'Disable all prior measurements' (unchecked), 'Disable prior measurements from this face' (checked), and 'Don't change prior measurements at all' (unchecked). 'Reporting Tolerances (0 = no tolerance)' has 'Angular Pointing Error' set to '0.005 degrees' and 'Uncertainty Tolerance' set to '0.2032 job units'. At the bottom, there is a checked checkbox for 'Move Objects and Points created by an instrument when it is moved.' and a 'Geometry Fit Profiles' button. The bottom of the dialog has 'OK', 'Cancel', and 'Apply' buttons.

User Options

Hidden-Point Bar Database Machine Configuration Security Automation

Display Units Light Conditions Analysis Scale-Bar Database

Uncertainty

68.26% Confidence Interval (1.0 sigma)

Sensitivity Samples 1000 ☒ Apply Gaussian dist.

Cloud radius relative to Target radius 1.0 ☐ Draw using lines

Target Computation

☐ Ignore Distance Measurements when possible

Prior shots from an instrument

☐ Disable all prior measurements

☒ Disable prior measurements from this face

☐ Don't change prior measurements at all

Reporting Tolerances (0 = no tolerance)

Angular Pointing Error 0.005 degrees

Uncertainty Tolerance 0.2032 job units

☒ Move Objects and Points created by an instrument when it is moved.

Geometry Fit Profiles

OK Cancel Apply

Analysis Options

Uncertainty Options

This area controls **SpatialAnalyzer** uncertainty analysis algorithms. **SpatialAnalyzer** determines the uncertainty of a measured coordinate using a simulation approach. Random error is injected into the measurement values within the known precision of the device. This results in variation in the coordinate values. This variation is processed statistically, and the resulting uncertainty is then determined.

The confidence interval setting determines the value that is multiplied by the standard deviation of each coordinate axis to determine the confidence. The default setting is 95%, but you can change this to the value you are accustomed to working with.

The Sensitivity Samples field controls the number of points that are computed (using the random error injection process described above) to produce a statistical set for analysis.

Setting this to a large number will produce a better result, but will also try your patience when you perform the analysis on a large set of coordinates.

The Cloud Radius Relative to Target Radius simply controls the viewing scale of the sensitivity clouds. In the random error injection process, many points are computed. If desired, you can display these in the graphics. Before drawing the clouds, the largest cloud is located, and the scale for the points is set so that the largest cloud will have an extreme point that is some factor of the target radius.

The Draw Using Lines check box also controls the graphical display of the uncertainty clouds. If checked, the points are not drawn as a cloud, but are instead drawn with a line from each of the actual points to the cloud points.

The Apply Gaussian Distribution check box allows you to control whether a Gaussian or a Rectangular distribution is applied to the noise that is injected into the measurements.

Target Computation

The check box is typically enabled for total station theodolite systems where the distance measurement is less accurate than the angle measurement. This option only has an effect when two different theodolite type (angle, angle, distance) instruments measure the same target. In that case, if this option is enabled, the distance components of the measurements will be ignored, and only the angular measurements will be used. This option **SHOULD NOT** be checked when using laser trackers, as it will actually degrade the accuracy of combined measurements. This is due to the fact that in the case of a laser tracker, the distance component is more accurate than the angular component.

The check box will disable all but the most recent measurement from an instrument to a target. If checked, you can re-measure a target and the previous measurements will be disabled. This allows you to correct an error without going back to the user interface, locating the bad measurement, and disabling it. Front and Back sight measurements are considered different types of measurements, so only multiple front or back sight measurements will be removed. Make sure this box **IS NOT** checked if you want to use multiple measurements from an instrument using the same face of the device in the point computation process.

Reporting Tolerances

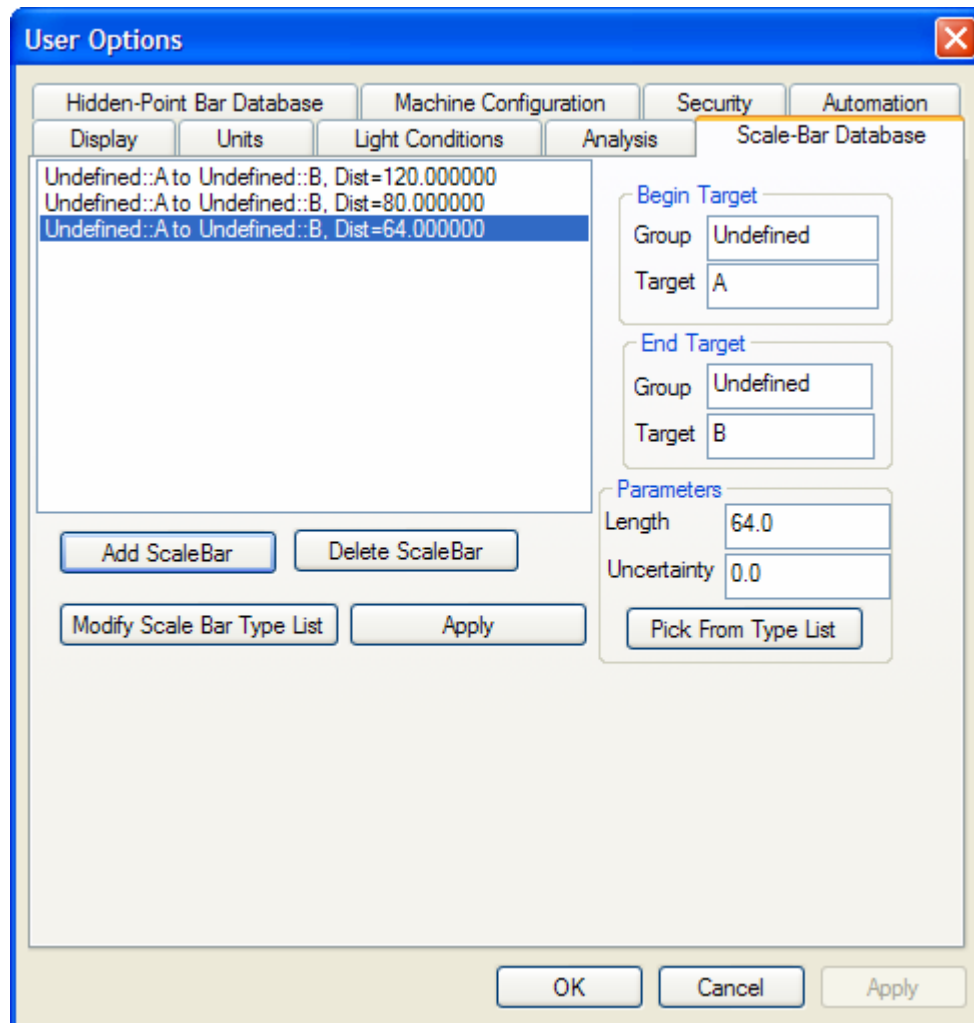
These tolerance values control the level at which the reports will flag pointing errors and uncertainties.

Automatic File Backup

SpatialAnalyzer makes a complete copy of the current job after a certain number of significant operations (a value of zero (0) will disable the auto-backup feature). This field controls the number of significant operations that trigger a backup. Significant operations include removing objects, transforming objects, performing a bundle adjustment, etc. Basically, anything that causes an important change to the internal job database.

Each backup is an entire copy of the job file. It is as if you performed a Save As operation. The file is placed in the Analyzer Data\Backup directory with a filename that is the name of the job appended with the time and date of the backup. This makes it possible to look in the backup directory and recall a job at different stages.

Scale-Bar Database




Scale-Bar Database


How Scale-Bars are Used

Scale-Bars are used primarily in the Bundle Adjustment process. During the bundle, the being and end targets in the scale-bar database that match the points being bundled are used to constrain the optimization. If the target names match the scale bar end names, the optimization will attempt to make the distance between the points match the distance set in the scale bar database.

As with all other target name matching processes in **SpatialAnalyzer**, the points names are case insensitive and space insensitive so that “Point p5 Left” is identical to “pointp5 le ft”.

Adding a Scale-Bar

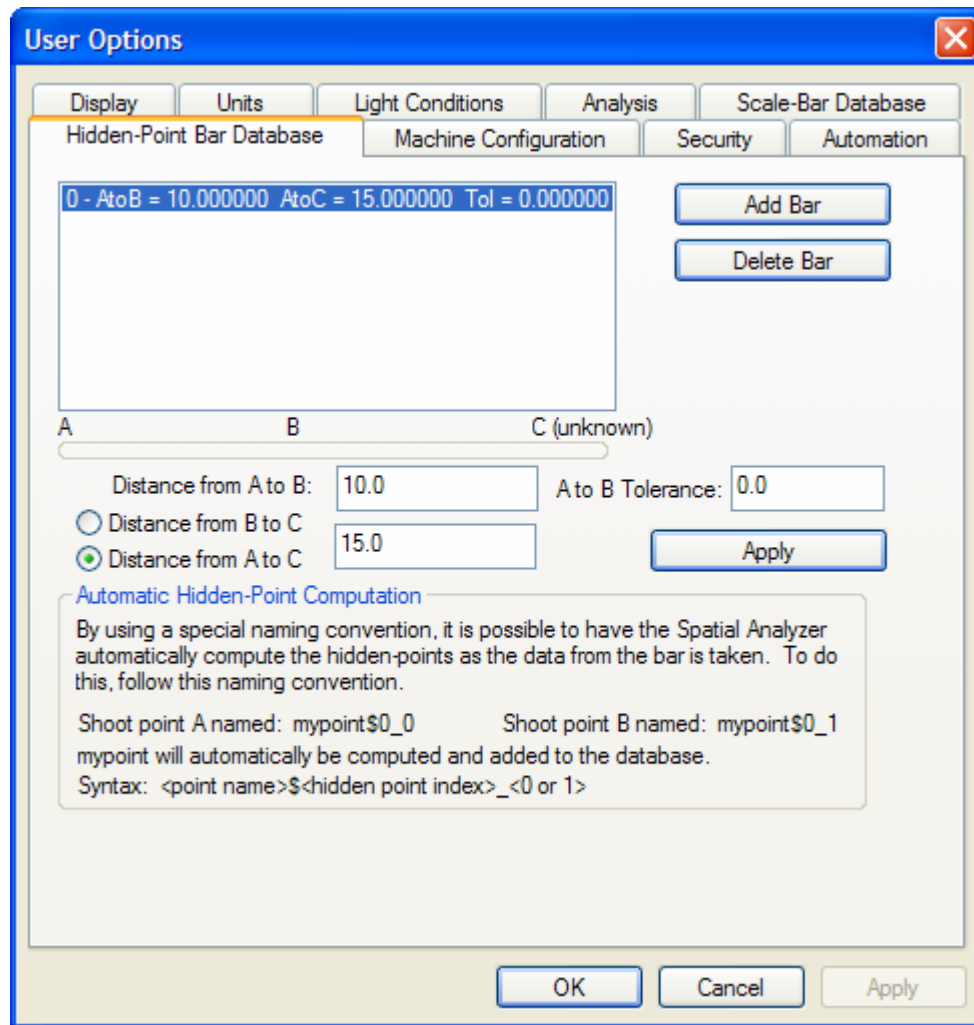
To add a scale-bar, press the  button. A new, undefined scale bar will appear in the scale bar list window. Next, edit the fields defining the scale bar properties. Set the beginning and ending target as well as the length and uncertainty of the bar.

After setting the parameters, press the  button. The Scale Bar list window will update with the new settings.

Modifying a Scale-Bar

To modify or remove a scale bar, highlight the bar of interest in the list window, and edit its properties in the fields to the right of the dialog. When you are done, press Apply, and the database will be updated.

Hidden-Point Bar Database



Hidden-Point Bar Database

The Hidden-Point Bar Database allows you to set the characteristics of the hidden point rods you wish to work with. Once configured, you can measure hidden points using one of two methods. The first mode, automatic mode, will determine that points are on a hidden point bar using a special naming convention described in this dialog. It will compute the location of the hidden point and add it to the model. The second mode, manual mode, requires you to construct a point using the option. You will be prompted to select the type of bar to use and the points that are measured on the bar. The hidden point will then be created for you in a group you can select.

Adding a Hidden-Point Bar

Press the button. A new default bar will be added for you. Next, select the way you wish to define the bar and enter the reference and projection length. Once the settings are properly configured, press the button to save your changes.

There are two different methods for defining the distance to the hidden point. You can dimension it from the first point, or the second. You must, however, supply the distance between the two reference points on the bar.

Modifying a Hidden-Point Bar

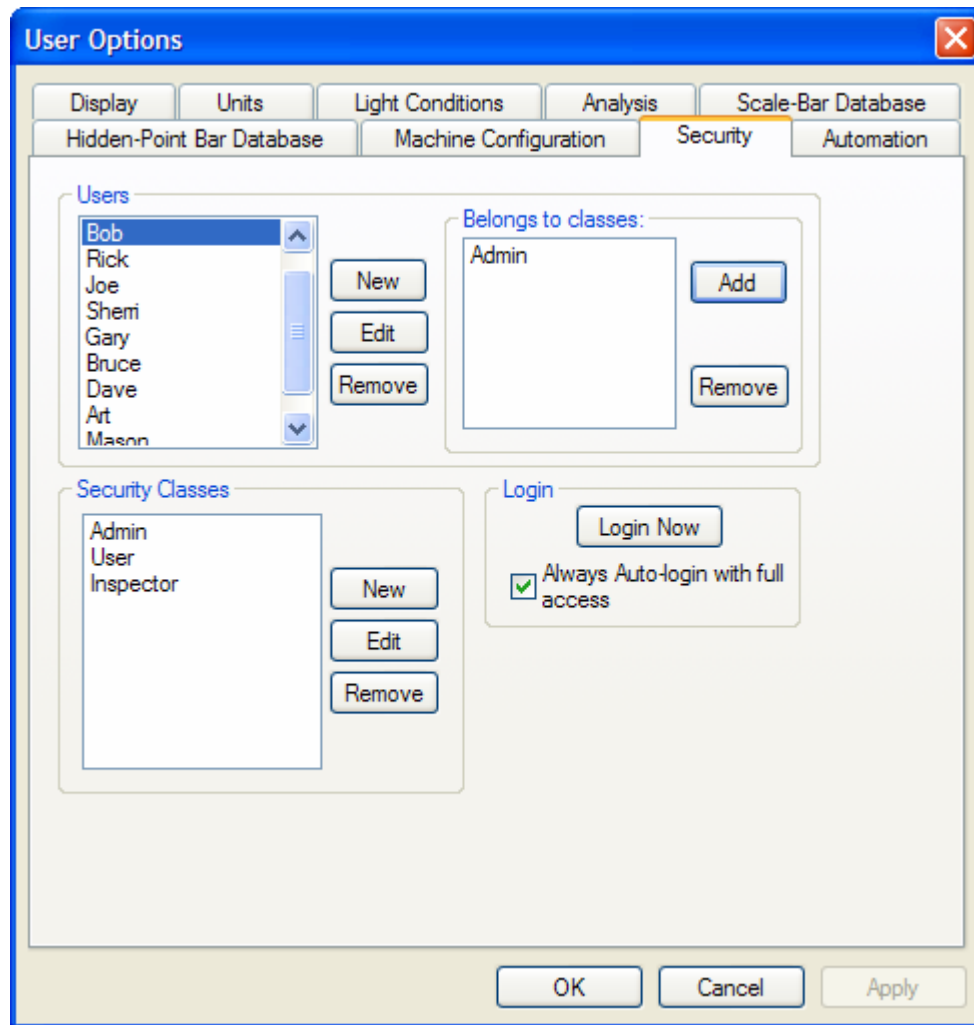
Select the bar you wish to modify or remove. Edit the fields and properties. Press **Enter** to save your changes.

Automatic Hidden-Point Computation

As mentioned previously, **SpatialAnalyzer** will automatically construct hidden points if you follow the naming convention described in the dialog. There are several issues you must be aware of pertaining to this automatic computation.

- The hidden points are computed as soon as both points (for A and B) are measured.
- If you bundle-adjust, or perform another operation that moves these points, you must go to the **Tools** menu and select the **Regenerate Hidden Points** option to regenerate the points.
- The hidden-points are treated as points, not targets. This is due to the fact that an instrument does not directly measure them. In order to look at the measurement information used to create the point, you must look at the targets that measured points A and B.

Security



Security Options Tab

SpatialAnalyzer security system follows the same organization structure as the Windows NT security system. **SpatialAnalyzer** has security classes. These classes have specific privileges controlling the user's access to the program's functionality.

Each user is a member of one or more security classes. In the case where a user is a member of multiple security classes, the attributes are combined in a logical OR sense. This means that if a user is a member of two groups where group 1 CAN perform function A but NOT B and where group 2 CANNOT perform function A, but CAN perform B, the combination will yield a user who CAN perform both A and B.

Login Settings

By default, the check box is checked. This means **SpatialAnalyzer** security is completely disabled. If you uncheck this box, security will be activated. This means the next user to run **SpatialAnalyzer** will be required to pass through the login process.

DANGER: Make sure you add an Administrator level user with all permissions activated so you can log back into the system!

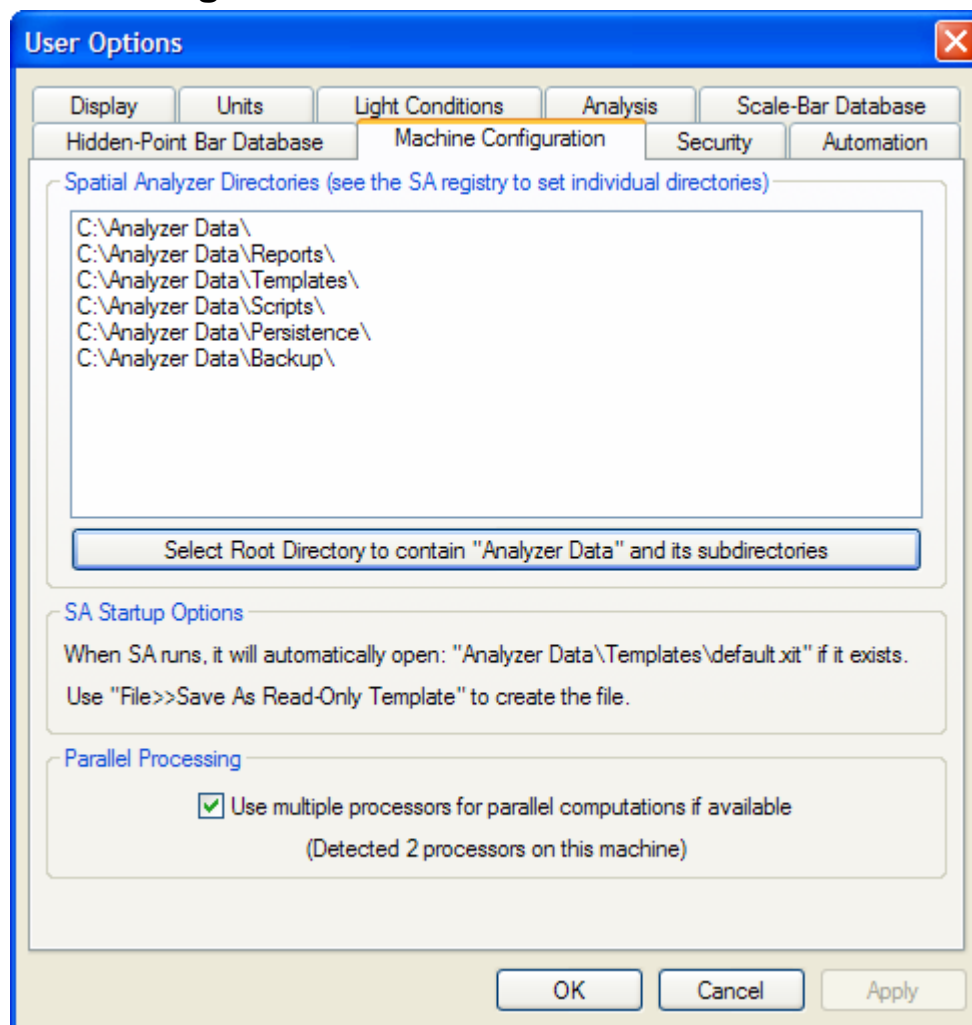
The button allows you to re-login as another user. This is useful for verifying the security configuration settings.

Configuring Security

Begin by adding several security classes. The figure above shows an example where the classes consist of an administrator, a user, and an analyst.

Next, add several users. For each user, assign a class (or classes) to which they belong. Make sure to create at least one user with administrator level clearance.

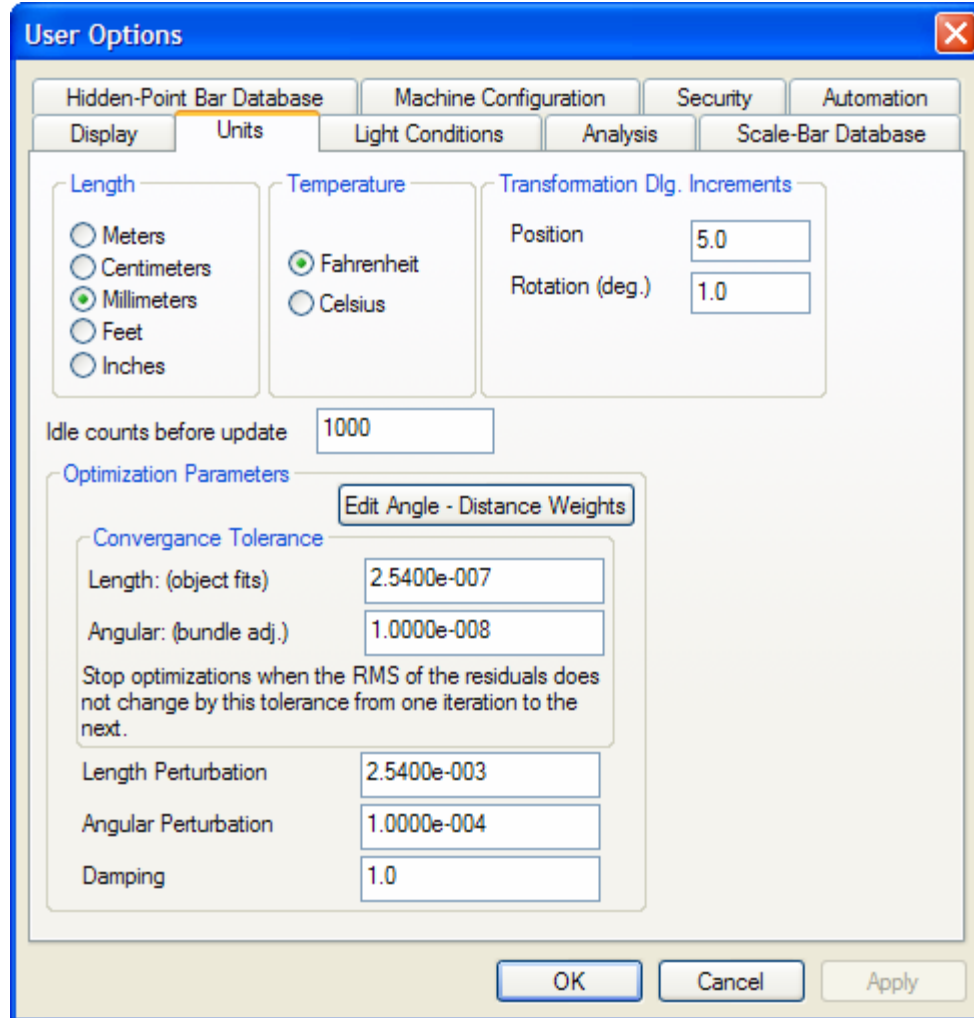
Machine Configuration



Machine Configuration Tab

This dialog lists the directories where **SpatialAnalyzer** will store its data files. Currently, these settings may not be changed from within the program. Instead, you must edit these through the system registry.

Units



The image shows the 'User Options' dialog box with the 'Units' tab selected. The dialog has a blue title bar and a standard Windows-style window with a close button. The 'Units' tab is highlighted in the tab bar at the top. The main content area is divided into several sections. The 'Length' section has five radio buttons: 'Meters', 'Centimeters', 'Millimeters' (which is selected), 'Feet', and 'Inches'. The 'Temperature' section has two radio buttons: 'Fahrenheit' (selected) and 'Celsius'. The 'Transformation Dlg. Increments' section has two text input fields: 'Position' with the value '5.0' and 'Rotation (deg.)' with the value '1.0'. Below these is a text input field for 'Idle counts before update' with the value '1000'. The 'Optimization Parameters' section is expanded, showing a button 'Edit Angle - Distance Weights'. Below this is a 'Convergence Tolerance' section with three text input fields: 'Length: (object fits)' with '2.5400e-007', 'Angular: (bundle adj.)' with '1.0000e-008', and a descriptive text 'Stop optimizations when the RMS of the residuals does not change by this tolerance from one iteration to the next.' Below this are three more text input fields: 'Length Perturbation' with '2.5400e-003', 'Angular Perturbation' with '1.0000e-004', and 'Damping' with '1.0'. At the bottom right are three buttons: 'OK', 'Cancel', and 'Apply'.

Section	Option	Value
Length	Meters	<input type="radio"/>
	Centimeters	<input type="radio"/>
	Millimeters	<input checked="" type="radio"/>
	Feet	<input type="radio"/>
	Inches	<input type="radio"/>
Temperature	Fahrenheit	<input checked="" type="radio"/>
	Celsius	<input type="radio"/>
Transformation Dlg. Increments	Position	5.0
	Rotation (deg.)	1.0
Idle counts before update		1000
Optimization Parameters	Edit Angle - Distance Weights	
	Convergence Tolerance	
	Length: (object fits)	2.5400e-007
	Angular: (bundle adj.)	1.0000e-008
	Stop optimizations when the RMS of the residuals does not change by this tolerance from one iteration to the next.	
	Length Perturbation	2.5400e-003
	Angular Perturbation	1.0000e-004
Damping	1.0	

Units Options

Length Area

The length area contains a series of radio buttons that set the current units of the job. The default units are inches. If you change the units, all of the data in the job will be scaled so that it reflects the change in units. For this reason, it is strongly recommended that you set the units properly in a template, or at least before working with the job. If you wish to change units on a job already in progress, make sure to back the job up before performing this operation.

Increments

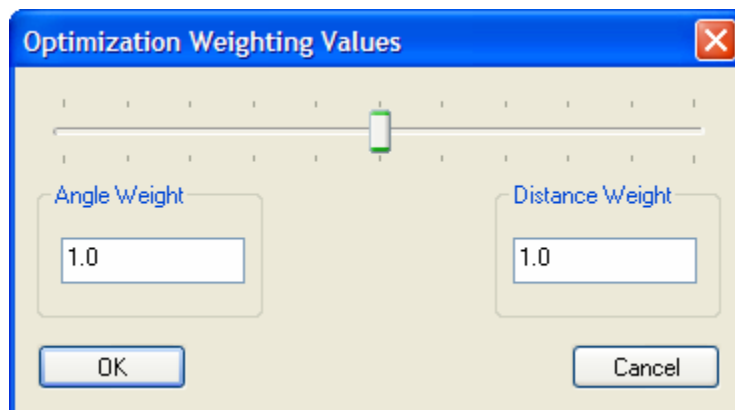
This area controls the degree to which the spin (or arrow) controls on the edit fields within the dialogs effect the numerical value. By default, position is 3. This means that if you open the transformation dialog by editing the properties of an object, and press the up arrow for the X position value, the object will move 3 job units (inches by default) in the X axis.

Idle Counts Before Update

This setting controls the number of idle counts that must expire before the graphics will be updated in **SpatialAnalyzer**. The meaning of this value is somewhat obscure. It is based on the Windows idle time processing loop. If the computer is not busy, an idle count will occur within the program. The rate at which the machine increments the idle count is not directly convertible to seconds.

Edit Angle-Distance Weights

This button controls the relative weighting between angular and distance values in optimizations. This is most applicable to Bundle Adjustment operations with devices that measure angles and distances. Pressing this button will allow you to set the values using the dialog shown in below.



Angle-Distance Weighting Dialog

In this case, angular values are deemed to be $\frac{1}{2}$ as important as distance values during the optimization process. This makes sense for a laser tracker measurement system since the distance values are more accuracy that the angular ones.

Convergence Tolerance

Throughout **SpatialAnalyzer**, numerical optimization techniques are applied to solve complex geometrical problems. These settings allow you to control to what extent the optimization attempts to minimize the objective function. Putting a smaller tolerance in these fields will cause the optimization to run longer, work harder, and theoretically reach a better solution. The default values have been found to work well for the current array of measurement devices and geometrical optimizations. The length tolerance is used for most of the object fitting operations and best-fit transformations. The angular

convergence tolerance applies mostly to Bundle Adjustment operations since the residual errors are tabulated in terms of angular values.

Length and Angle Perturbations

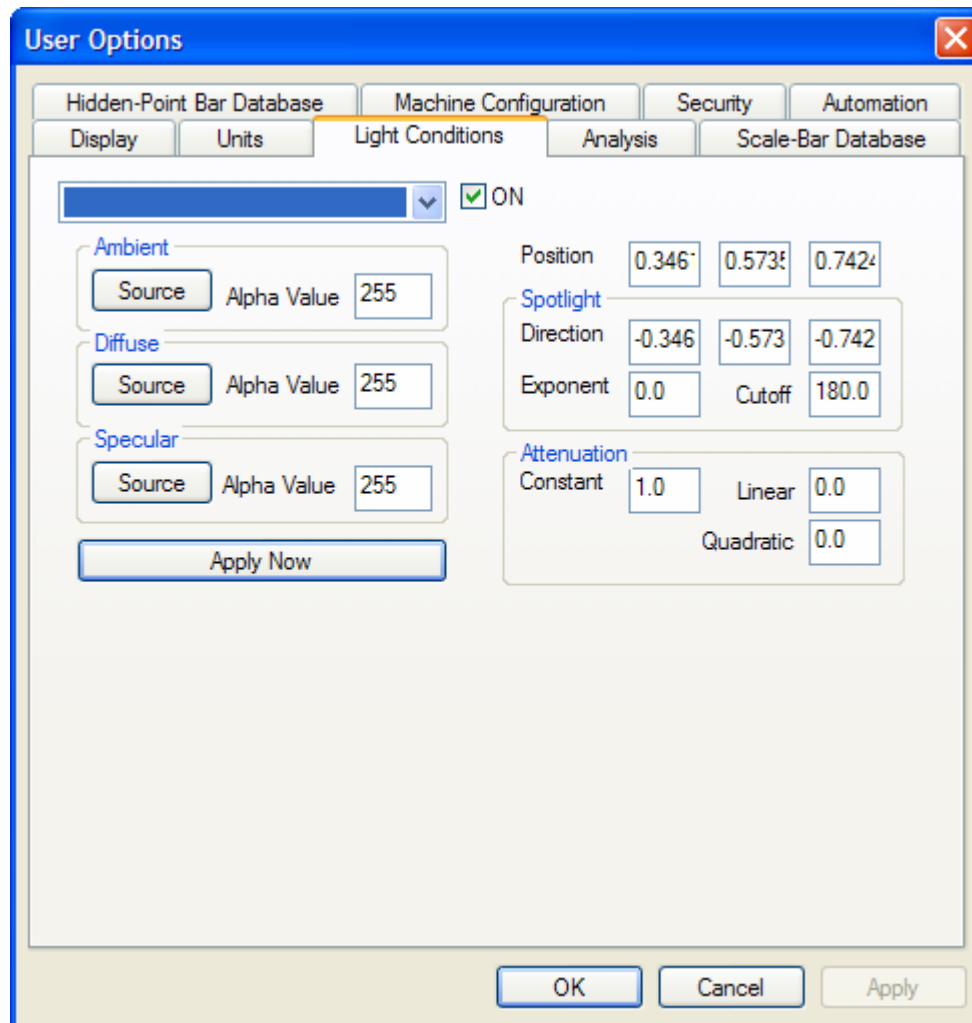
For many of the numeric optimizations within **SpatialAnalyzer**, the partials matrix is computed using numerical differentiation. A value is changed by a delta, and the result is computed. The deltas used to offset the values in an optimization are set here. In the case of a best-fit transformation, for example, the x value is moved by the length perturbation. The objective function is recomputed, next the y then the z. After that, the Rx value is perturbed by the angular perturbation, then Ry, then Rz. At each step, one portion of the partials matrix is populated.

If these values are too small, you can experience sensitivity problems. The perturbation may be so small that the change in the objective is negligible and therefore the optimization proceeds without accurate direction. If the values are too large, it is possible to miss nuances in the solution space by moving over the ideal solutions.

Damping

Once the partials matrix is computed for a given step in an optimization, the matrix is inverted, and multiplied by the residual errors. This results in a vector describing how to move the input variables to minimize the objective if the system were truly linear. Since the systems we deal with in coordinate metrology are usually non-linear, the delta vector will most likely not move to the optimal solution. For this reason, we scale the vector by the damping factor to retard its effect. After making the move, we then repeat the partial determination process and so on.

Light Conditions



Light Conditions

All objects drawn in the Spatial Analyzer model are rendered based on various lighting and Color properties. For a more detailed description of various lighting and Color models, refer to the book, "Computer Graphics: Principles and Practice", Foley & VanDamm 1996.

The Lighting Conditions dialog will allow you to change the properties of any of the 8 lights in the model. The default condition is to have only light 0 enabled. This is usually sufficient for most models, though additional lights may be used to add a more realistic feel to the environment. It is important to note, however, that additional lights substantially increase the computational burden and will slow the graphics down.

Ambient, Diffuse, and Specular Light Settings

There are three components of light available in the environment: ambient, specular, and diffuse. Ambient light is the light level that is present ewhere in the environment and

impinges on e part of the model. Specular light is the light component that only becomes visible when a surface on the object reflects light directly back toward the user's point of view. Diffuse light emanates from the location of the light source. This light will reflect off the object based on the orientation of the surfaces to the light source.

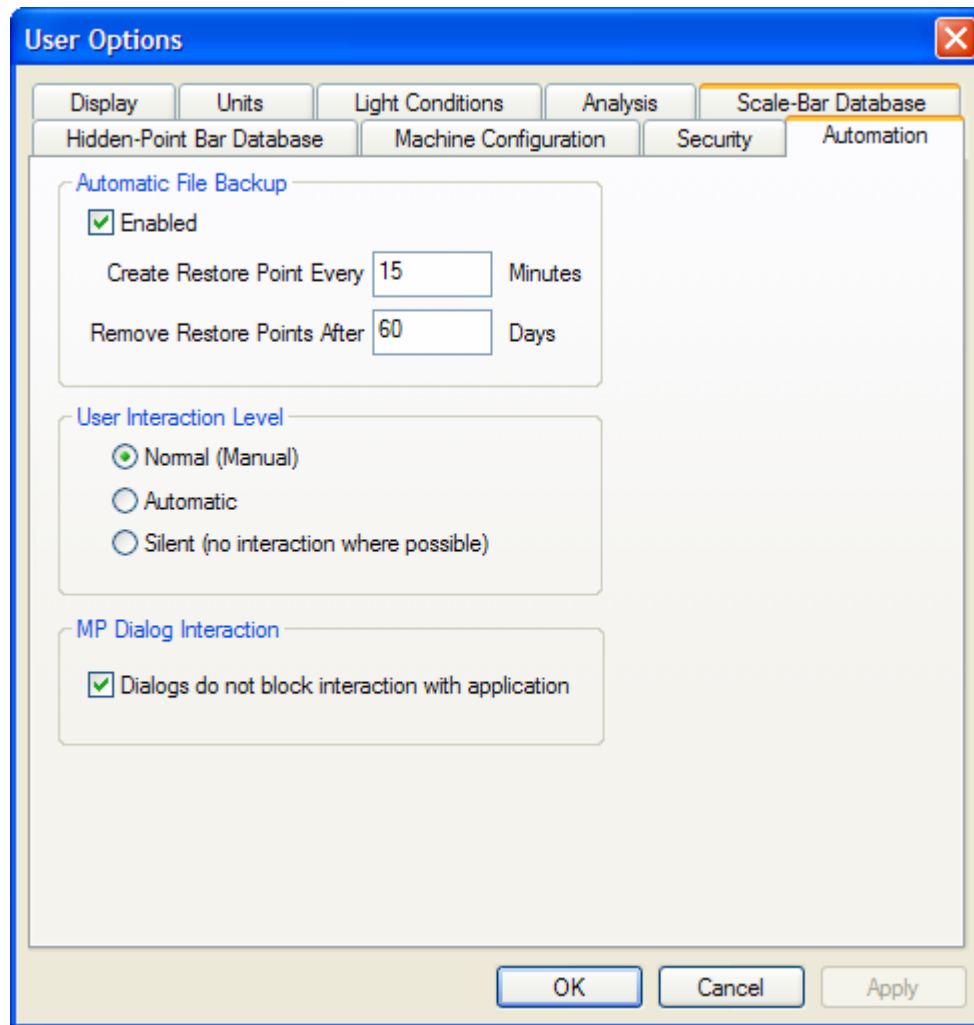
Pressing the associated button may configure each component of light. This will cause the color dialog to appear. Using this dialog, select the lighting values you want, and press ok. Once you then press Apply from the Lighting Conditions tab, the change will be visible in the model environment.

There are several other more advanced lighting configuration options available such as position, attenuation, and spotlight effects. These are described well in the Foley VanDamm text or other basic computer graphics references.

Each object contained in a **SpatialAnalyzer** model file has an associated Color. This Color determines how light from the environment is reflected toward the viewer for any given configuration.

Each Color has three separate color components and four additional parameters. Each will be described below. It is important to note that not all of the properties will be used on all of the objects in the model. Anything drawn using lines, for example, will be drawn with the color specified by the Ambient and Diffuse reflectivity.

Automation



Automation Settings

Automation settings determine how much or little interaction you want from SA. These settings are typically changed when SA is used with an MP where the user does not interact with SA.

The Normal setting provides the highest level of interaction. Automatic user interaction will run without interrupting the user with results if there are no errors. If an error condition occurs, SA will present a dialog to the user asking for action. Silent interaction turns off all possible reporting.

MEASUREMENT PLAN

Measurement Planning (MP) is a powerful process control and programming language for **SpatialAnalyzer**. It allows administrators and users to program or script complex operations for production use.

Measurement Plan

Measurement Planning (MP) is a high-level (i.e., easy to use) powerful process control and programming language for **SpatialAnalyzer**. It allows administrators and users to program or script complex operations for repetitive tasks and production use. MP exposes an advanced metrology suite of tools with an interface that makes it easy to capture complex measurement processes into a succinct program. The programming process and environment is developed with the experience metrology system user as the target programmer. Only minimal programming experience is required before a user can craft and deliver automated metrology solutions to your customers.

A list of some of the key features:

- File Operations (Import/Export of CAD, Templating, Backups...)
- Load/Create Job Templates
- HTML Interface Support
- Add Instruments → Start Instrument Interface
- Data Collection → Drive Instrument and external devices (e.g., turn-table, translation stage)
- Auto-Alignment to CAD
- Feature Detection → Extraction → Geometry Creation
- Feature Analysis to Tolerances
- Automatically Generate Formatted Reports (MS Office)
- Save and Archive Inspection
- Multi-Run if desired
- Branching capabilities based on logical comparisons
- Referencing results and inputs
- Run any instrument (or multiple instruments) automatically
- Instrument Operations (automated tie-in, drift check, Op checks...)
- View Control with animated sequencing
- Standard SA Analysis Functions
- Utility Operations (guide users, issue speech notices, control NC hardware)

MP Benefits

The concepts in this process yield significant savings in tooling, programming, configuration management, measurement, analysis, data management and in process product verification. The system as a whole can be applied to other manufactured parts to gain further savings without non-recurring development costs.

Daily system/cell setup and checks can also be automated, utilizing the same programming concepts and user interface. This allows for secure operator system checks and documented corrections to keep product flowing.

A list of some of the key benefits:

- Survey Variability Reduction
- Enhanced Training and Minimized Learning Curve for Users
- Consistent, and Customizable Reporting
- Simple User Interface
- Applicable to any Spatial Metrology Instrument

Overview of MP Architecture

By using **SpatialAnalyzer's** Measurement Plan (MP's), operators can script repetitive or automated metrology tasks. These plans can vary from simple utilities constructed in just a few minutes to comprehensive inspection plans that provide go/nogo type automation and drive other machinery such as robots or CNC machines.

The next section will introduce a simple process for structuring a template and a measurement plan to complete a general scripted measurement task.

Measurement Plan Components: Template, Script and Interface

A template job, MP script, and a simple machine interface are three basic components for a scripted measurement process.

- Template
- MP Script
- Simple Interface

Measurement tasks typically have some components that are consistent each time the process is run and some that vary each time. Saving the components that are consistent each time into a template saves a significant amount time and ensures consistency between runs. For the components that change each time a script is created that initially loads the template and then uses the elements from the template to drive the measurement process. The interface between the script and user is the other element in the process. Having a straight-forward easy-to-understand interface ensures the process is repeatable.

Structuring the measurement automation components (template, script, and interface) in this way allows you to create a simple template for each part and a general script that is able to run different parts. The programming for a particular process is then re-useable.

SA Template

A template job is typically created for each part that runs through a factory process. It normally begins with the designed CAD model and roughly positioned measurement instruments. The model is exported from the design package in a number of different formats e.g., STEP (STandard for Exchange of Product) model data and then imported into SA. These SpatialAnalyzer (.xit) files become templates for each part and contain the specific attributes that need to be inspected, checked, or monitored.

The next elements in a template file are the instruments and their respective settings. The specific instruments that will be used in the process are added to template and located (roughly) in relation to the part geometry. Rough locating the instruments in the template is a good way to help guide the user during the setup process. The measurement acquisition settings are then configured for each type of measurement needed for the job, e.g., Tooling Ball Measurement Profile, Special Plane Measurement Modes, Turn-Table positions, Reflector Definitions etc.

MP Script

The second component is the MP script. It contains the process specific elements e.g., template loading, data acquisition, orientation, feature extraction, tolerances, comparison to nominal or between attributes, report generation, and inspection archival. These plans or scripts can be general enough to run a whole class of parts (e.g., 200 different parts) which eliminates the need for individual programs for each part.

MP's can typically be thought of as a Script of Measurement Process. Typical steps in that process involve.

- Load Template
- Start Instrument Interface
- Data Collection → Drive Scanner & Turn Table
- Close the Instrument Interface
- Auto-Alignment to CAD
 - Move Objects
 - Move Instrument(s)
- Feature Detection → Extraction
- Feature Analysis
- Generate Report
- Save and Archive Inspection

Measurement Plan contains a comprehensive set of process control tools that enable you to control and define the measurement process. As the script is executed the results of each step in the process can be check and verified. If a step is not completed successfully your script can detect the discrepancy and then branch into your error handling processes that are under your control.

Simple Interface

The third element is a simple interface for the operator. Its function is to wrap the process into simple inputs vs. outputs. Inputs include information such as, part number, lot quantity, and operator name. After each part inspection the simple interface shows either a red or green result window indicating pass or fail. A summary of the important attributes is also available to provide feedback on which features passed and failed.

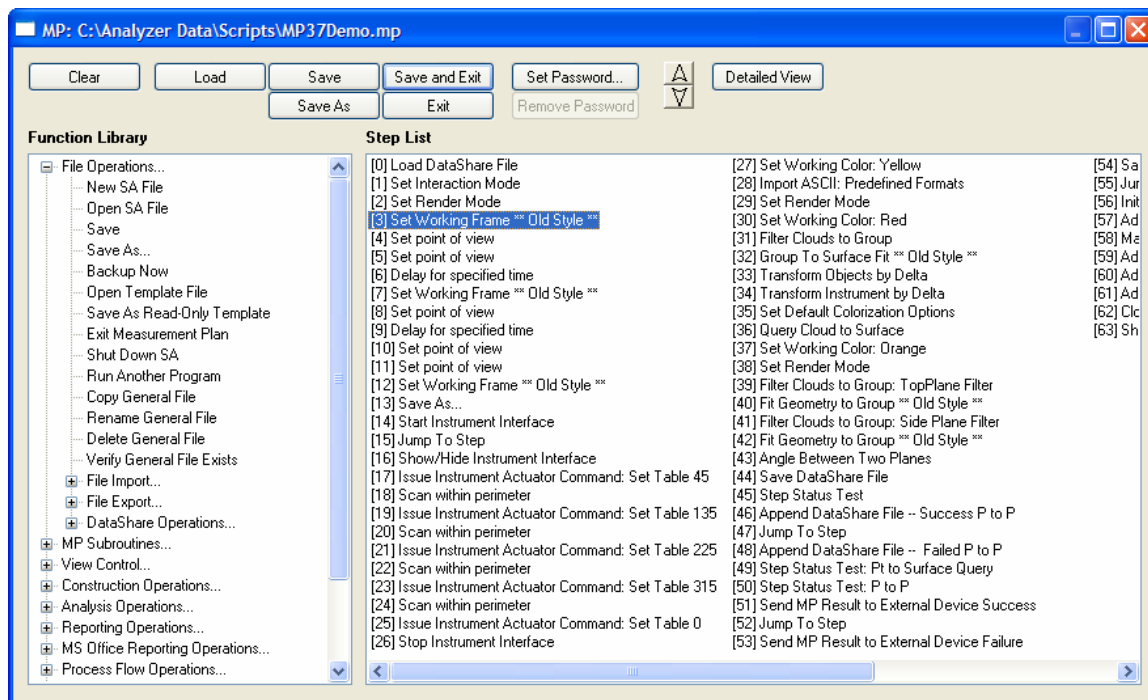
MP Editor

The **SpatialAnalyzer** MP Editor allows you to view and edit MP steps within tree and list controls in the editor. You can also add and delete MP steps, such as another instrument, CAD export, change in point of view, and delete temporary construction geometry.

The MP Editor supports a number of key features that make an MP easier to develop and document. It is a visual programming tool in that all of the functions are listed in a treeview and once inserted into a script the arguments can be configured by entering the data directly or can reference arguments anywhere in the MP. A list of some of the features of the editor is:

- Treeview of functions
- Each function has a simple list of parameters
- Assemble sequences of functions, looping, and conditional testing that capture the measurement process.
- Arguments can reference values from other MP steps for error trapping, customizable reporting, and process control.
- Subroutines functionality helps modularize and re-use script components.
- Detailed view of the Measurement Plan supports documentation and printing.

An example editor session is shown below. Note the treeview on the left and the active script on the right.



MP Editor Environment

When you are finished modifying plan steps in the **SpatialAnalyzer** MP Editor and you return to **SpatialAnalyzer** application where the job is open, you can run the MP script on the job to apply the processes you made need to the elements in the job. With the **SpatialAnalyzer** MP Editor open, you can create new MP script files and save them directly to any available location, and you can open existing MP scripts files from any available location to edit or debug.

Detailed View

The Detailed View button on the MP Editor opens a printable version of the MP script that shows each step and the individual arguments. Printing this view is help track references and to document your Measurement Plan. An example Detailed View is shown below.

StartEngine.mp

0-Set (or construct) default collection Step Type: Set (or construct) default collection

Arg	Type	Description	Method	Value
0	Collection Name	Collection Name	Enter Value	B

1-Set Working Frame Step Type: Set Working Frame

Arg	Type	Description	Method	Value
0	Frame Name	New Working Frame Name	Enter Value	Airplane

2-Create Counter Step Type: Create Counter

Arg	Type	Description	Method	Value
0	Counter	Counter Value	Enter Value	0

3-Integer Comparison Step Type: Integer Comparison

Arg	Type	Description	Method	Value
0	Integer	Integer A	Reference	Ref {S2 A0}
1	Comparison Type	Comparison Type	Pick	<
2	Integer	Integer B	Enter Value	1200
3	Step ID	Step if TRUE	Reference	Ref {S4}
4	Step ID	Step if FALSE	Reference	Ref {S7}

4-Transform Objects - Frame To Frame Step Type: Transform Objects - Frame To Frame

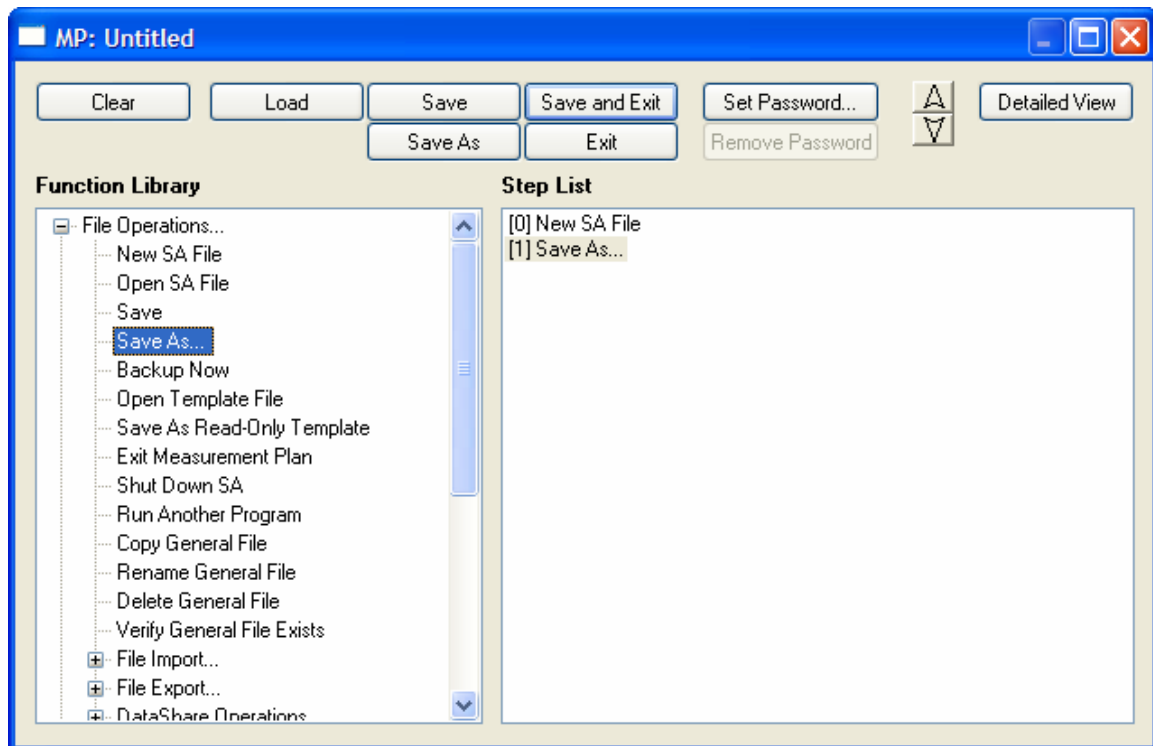
Arg	Type	Description	Method	Value
0	Object Name Ref	Object Name List	Enter	PropS Nut Prop PerimeterProp
1	Frame Name	Initial Frame Name	Enter Value	Airplane
2	Frame Name	Destination Frame Name	Enter Value	DeltaMove

MP Details View

An MP Step

The MP architecture ensures each MP step is able to function autonomously or in other words it can be the one and only step in a measurement plan or it can contribute within a sequence of steps to accomplish more complex tasks. To understand MP programming we need to understand what an individual step requires verse what it can yield. To begin lets look at the pieces and parts of a simple and a complex MP step.

Go to the Scripts menu in SA and select [Create/Edit Measurement Plan...] which will start the MP Editor environment. Open the [File Operations...] branch of MP functions and double left click on the functions called [New SA File...] and [Save As...]. The figure below shows the MP Editor with the File Operations branch open and the two MP steps added into the active script.



Steps with and without Arguments

MP Step without Arguments

A good place to start with MP is to look at a simple MP step or function. The basic structure of each step is the same. Beneath each step are a set of properties that you can configure.

Each MP step is able to accomplish a task; in this case the task is to open a new SA file. By double clicking on the function in the active script pane the function's properties page is displayed. The figure below shows the New SA File properties page.

Step Number 0

Step Type = New SA File

Step Title:

Comment:

Arguments				
Arg#	Type	Description	Method	Value

OK Cancel

Properties page of MP Step with Arguments

An MP step has a number of attributes that need to be described.

MP Step with Arguments

Step Number 1

Step Type = Save As...

Step Title:

Comment:

Arguments				
Arg#	Type	Description	Method	Value
0	File Path	File Name	Enter Path	
1	Boolean	Add Serial Number?	Enter Value	FALSE
2	Integer	Optional Number	Enter Value	0

OK Cancel

Properties page of MP Step with Arguments

Data Types

There are several different types of data that you need to be aware of when programming with MP.

Standard Data Types

The message and response packets are composed of the following standard data types:

Data Type	Range
Bool	true or false
Int	-2,147,683,438 to 2,148,386,437
double	approx. +/- (3.0E-308 to 3.0E+308)
String	Alpha-Numeric Characters

Standard Data Types

Special MP Data Types

There are several special data types that you can use in MP when programming measurement and analysis processes. They include transformation objects, Object Name Lists, Frame Names, and Vectors. Since MP creates and manages these special data types programming becomes quite a bit easier. Understanding what they are and how they are used is an important component of successfully automating a factory process. These data types are typically created by one MP step and then use as a referenced in a subsequent step.

A list of special MP data types follows with a description of each type.

Class	Data Type	Description
Naming		
	Object Name	A string that is used to retrieve and object or it will become the object's name. If the name already exists in the database for the specific type of object an asterisk (*) will automatically be appended to the name.
	Object Name List	List of strings used to access one or more objects. The database is search for each name in the list. An example for this data type is when projecting points to multiple objects e.g., surfaces.
	Point Name	Special name type that contains both Group and Point name fields. The two are separated by two colons, e.g., [Ref::Pt1]

	Group Name	A string used to access a group of points or to name a group of points.
	Group Name List	List of strings used to access one or more groups of points.
	Cloud Name	A string used to create or identify a cloud object.
	Vector Group Name	A string used to create or get a Vector Group.
	Frame Name	A string used to either get a frame or to name a frame that is created during the measurement plan.
	View Name	A string that defines an existing graphical view port.
	Perimeter Name	A string that corresponds to a perimeter in the current SA job.
Instruments		
	Inst ID	An unsigned integer value that indicates an instrument's reference number within the SA job. Instrument numbers start at 0 and sequentially increase as each new instrument is added to the job.
	Instrument ID List	A list of unsigned integers that are used to identify instruments. They start at 0 and continue to the number of instruments in the job.
	Instrument Type	A selected item from a list of all the available instrument interfaces. This data type makes it easy to select the instrument type when inserting it into a job.
Step/Counters		
	Step ID	An integer value used to indicate a step's position within the MP. Note step ID's start at 0.
	Counter	An integer value used to control loop iteration. Tests of the counter's value are done to determine when to stop iterative processes.
	Counter Reference	A reference to a counter. This data type is typically used to either increment or decrement a counter's value.
Math		
	Math Operator	A selected math operation e.g., addition, subtraction,

		multiplication, or division.
File Operations and System Variables		
	File Path	A string with a valid file path.
	File Format	A selected format from a set of supported file formats e.g., Group::Point Name, X Y, Z
	System String	A string type used to hold information collected from the computer's operating system.
Units		
	Units	A conversion factor, 25.4, 12, etc. Typically used to adjust the length component of point coordinates when they are imported.
	Angular Units	A selected item from a list of support angular units, e.g., decimal degrees, radians, gon, deg:min:sec. It is used to convert from one angle unit to another, e.g., radians to deg:min:sec
	Time Units	A selected item from a list of supported time units, e.g., seconds and minutes. The time value is then scaled to match the selected choice for the specific use at run time.
SA Data Types		
	Color	A color based on the standard R-G-B selection dialog. This data type is typically used to set the current working color in SA.
	Colorization Object	A data type to control the colorization of Vector Groups. Once set in an MP subsequent vector groups within the MP are colorized with the settings in this object. There can be many colorization objects in a single MP.
	Projection Options	An object that sets Make Vector Group, Make Point Group, and Offset projection options. Projection direction and offsets can be configured for a particular analysis step. These settings generally do not persist between uses and therefore must be set for each use.
	Vector	3 doubles that define a direction, e.g., plane normal, point on a plane etc.

	Vector Tolerance	This data type holds high and low tolerances for each X, Y, and Z vector component as well as its magnitude, i.e., 8 values ... hi/low for X Y Z and RSS. The data type also retains a flag for each of the 8 tolerances. A specific tolerance flag is only used for acceptance when its flag is on.
	Double List	A list of doubles. One typical use for this data type is to set a whole series of instrument axes positions, e.g., set all 7 angle positions for a Portable CMM.
	Geometry Type	Geometry object type that is typically picked from the following list: Line; Plane; Circle; Sphere; and Cylinder. This data type controls which geometry type is computed in a geometry fit.
Transformation		
	World Transform Operator	An object containing rotation and translation coefficients that operates consistently regardless of the current working frame. This object is typically computed in a best-fit transformation and then applied to both objects and then instruments.
	Transform	An object containing rotation and translation coefficients that operates relative to the current working frame. This is used to apply a delta transform about the current working frame.
	3pt Frame Construction Method	Picked item that includes Origin, X XY; Origin, X, XZ; Origin, Y YX; Origin, Y YZ; Origin, Z ZX; and Origin, X ZY;
MP and SA Execution Mode Control		
	MP Step Mode	This data type sets the step execution mode to either single step or auto-run.
	SA Interaction Mode	This data type sets the SA inaction mode to Manual, Automatic, or Silent. Manual mode is the default. In Automatic mode SA with attempt to run through each MP process without presenting acknowledgement, notification, and response dialogs to the user. Silent mode suppresses warnings and all response dialogs during MP execution.
	Measurement Plan Interaction Mode	The measurement plan execution mode is set with this data type. The options include Halt on Failure Only, Halt on Failure or Partial Success, and Never Halt. Halt on



		Failure or Partial Success is the default mode were if any individual step fails any part then the MP will stop and allow you to acknowledge the conditions. The MP can be re-started from the current step. When in Halt on Failure mode the MP will stop when any step fails. Never Halt mode is used in production systems when error trapping and acknowledgement are handle within your MP script.
	Measurement Plan Result	Success, Partial Success, or Failure status flag. This flag can be used to check the results of a subroutine.
	Window State	A picked item that set the SA window state from the following list. Maximize; Minimize; Restore; Show; and Hide.
Reporting and Export		
	Word Headings	Selected item from list of options including: Heading1; Heading2; Heading3; and Heading4. The options correspond to MS Word style setting for text. They allow you to control text that is inserted into an MS Word report.
	Edit Text	A general multi-line text object. One typical use is for text that is to be inserted as into an open MS Word document. The MS Word document session must have been initialized with the Initialize Office Report MP step.
	Report Table	A configurable table to insert objects or computed results into an open MS Word document. One common use of this data type is for outputting formatted tables of statistical results that had been computed in prior MP steps.
	HTML Keyword Association List	A configurable list of keyword to references of data within the MP. This object is used to assign a mapping between specific keyword strings and references to data within an MP. When a document is opened each instance of the keyword is searched for and once found is replaced by the data value referenced in the list.

MP Data Types

Selection Methods

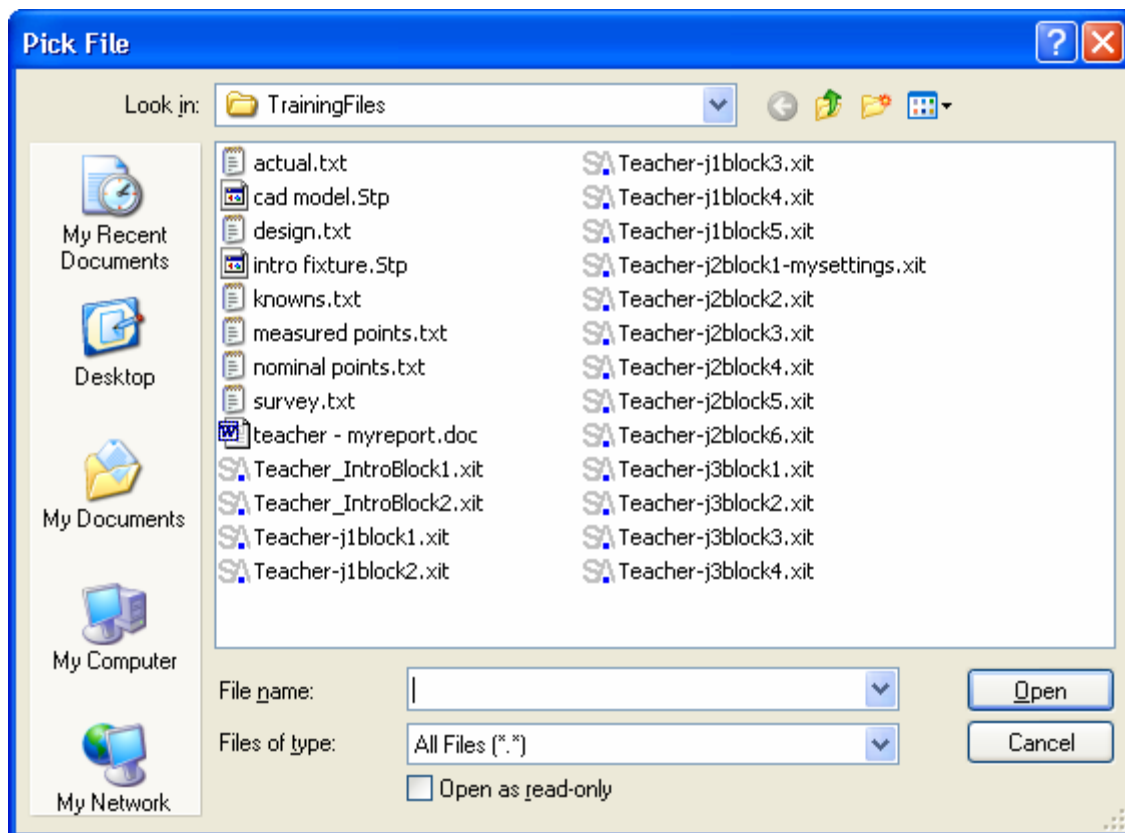
Within most MP steps there are a number of arguments. Some of the arguments are used to pass data into the function and other are used to pass computed information back out. For the arguments that pass information into a function a number of different selection methods are provided for each argument type. The different selection methods

allow you to control how information is entered into to each argument in a measurement process step. Available selection methods include:

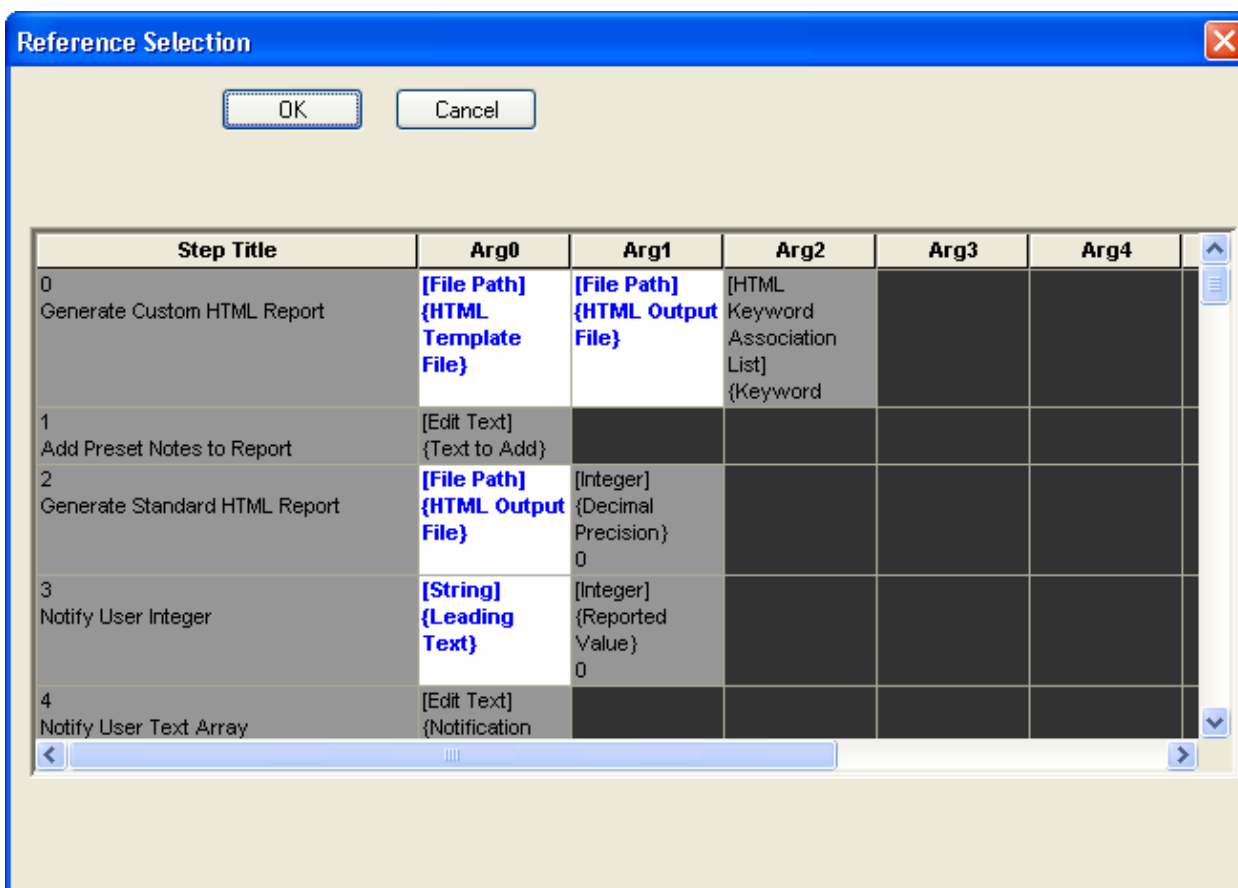
Selection Method	Description
Enter Value(s)	Enter values directly into the step's argument field.
Enter Path	Enter the path to the file directly into the step's argument field.
Pick	Select from the set of support options in the drop down list.
Browse	Select the  button in the Value column to open a Pick File dialog. Use the controls in the dialog to browse to the file you need to select. When you close the dialog the fully qualified path and file name are entered into the Value field for you.
Reference	Select the  button in the Value column to open a Reference Selection dialog. Navigate through the MP and select from the arguments that match the data type required by the argument.

\Selection Methods

An example Pick File dialog is shown below.



Pick File Dialog when using the Browse selection method



Reference grid selection method dialog

An example Reference Selection dialog is shown above.

Each argument automatically makes the selection methods available that make sense for the argument's data type. By automatically setting up the selection methods the ones that don't apply are not shown which help make MP scripting easier to consistently setup.

Result Arguments

Arguments that pass information out for use on subsequent steps are called Results arguments. These arguments are computed during the execution of the MP. The results argument is created when the step is added to script which allows you to reference it in subsequent steps. Since the value is not known when editing an MP the results value is green and is not editable on a step's properties dialog.

An example MP step with a Results argument for a Transform data type is shown below. For this step the instrument transform is returned in argument 2. The transform will be relative the Nacelle Reference Frame.

Step Number 29

Step Type = Get Instrument Transform

Step Title:

Comment:

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument ID	Enter Value	0
1	Frame Name	Reference Frame	Enter Value	Nacelle
2	Transform	Transform	Result Only	Result Only

OK Cancel

MP Step Properties page with a Results Only Argument

The Instrument Transform is available for subsequent operations in the MP. A reference selection method from another MP step is used to get at or use the values in the transform.

Running a MP

To run a MP, you first need to load it into **SpatialAnalyzer** with the menu command under the Script menu → Run MP. Navigate to the particular MP file and select load in the file selection dialog. The plan is initialed with the arrow button on the right of the plan step list control.

Single Step
Automatic
'Halt On' Behavior

Applications

The number and types of applications were MP technology has delivered solutions includes the list below. The project solutions that MP delivers on are always expanding.

- Automated Inspection
- Build and Assembly
- Advanced Analysis
- Auto-Feature Extraction
- Machine Control
- Simplify Operator Training
- Training

MP concepts for 3D metrology inspection cells enable parts to be inspected directly against their CAD definition, delivering easy to understand results, and minimize operator training. Utilizing SpatialAnalyzer's real-time graphics, Template files, and MPs you are able to utilize flexible tooling while automating product measurement with an a suite of 3D optical measurement system, as well as CAD model analysis and reporting.

Viewing and Running a Sample MP

The outline for an example MP is shown below to help your familiarize yourself with the MP Editor and to see the results.

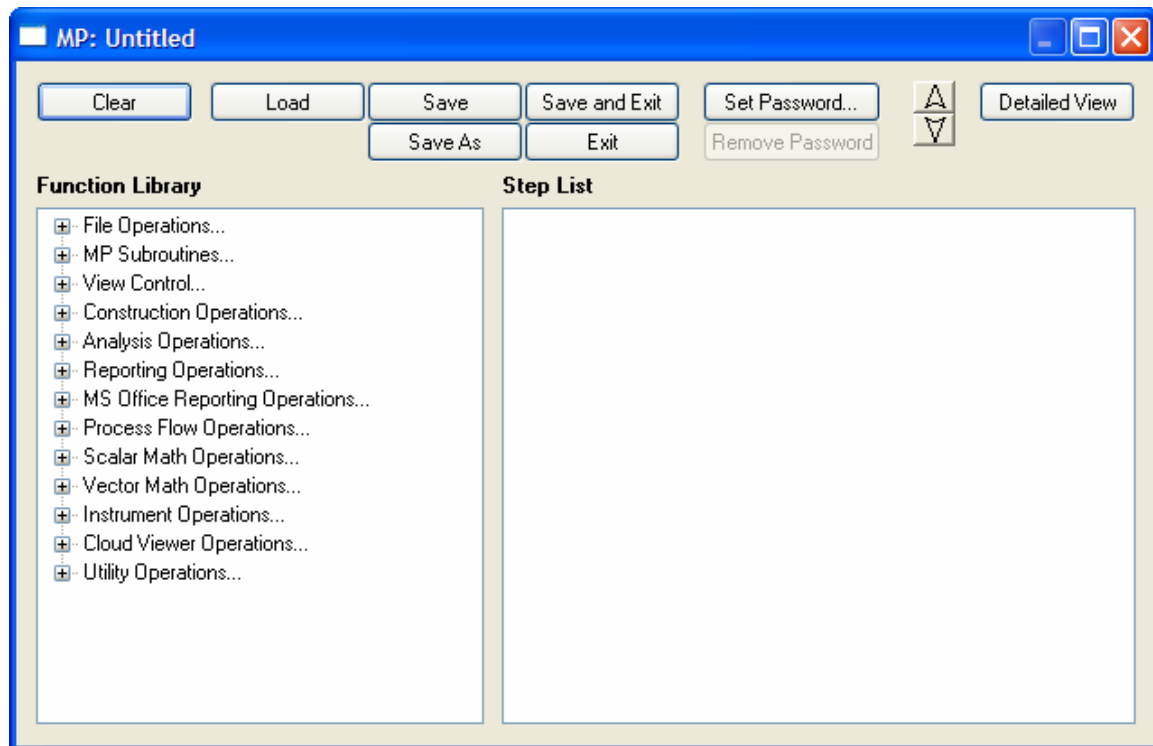
1. Start SA
2. Open a sample MP in the Editor
3. Editor Components
4. File Buttons (e.g., Load, Clear, Save, Exit)
5. Buttons to Move Step Up or Down
6. Treeview of Functions
7. Plan Listing
8. Right Click Menu
9. Double Clicking to Add or Goto Properties Dialog
10. Printing
11. Step Properties
12. Step Type
13. Title
14. Arguments
15. Type
16. Description
17. Method
18. Value
19. Save and Exit the MP Editor
20. Load and Run the MP
21. MP Dialog
22. Navigation Buttons
23. Start and Stop Buttons
24. Exit MP Button
25. Step Results Icons
26. Single Step Through the MP
27. Resize the Plan Window
28. MP ends
29. Review the results
30. Save the Job and Close SA

Measurement Planning (MP) is a powerful process control and programming language for **SpatialAnalyzer**. It allows administrators and users to program or script complex operations for production use.

Constructing a basic MP

This section is intended to provide a basic tutorial on how to construct a basic MP. It is not intended to be a comprehensive reference on all functions support by **SpatialAnalyzer**'s MP capabilities. To construct your first MP begin by starting the MP

editing dialog. This is done by selecting File->Create/Edit MP from the main menu. You will see a dialog like the one presented below:



MP Environment

The dialog is divided into two panes. The left pane presents a tree view containing all of the possible MP functions available to the operator. At the highest level of the tree structure the functions are divided into several broad categories to facilitate organization of these concepts. There are for example, file operations and instrument operations as category headings. The right pane will be an area where the user may construct individual MP steps into a plan.

To begin the designing process it is usually best to consider what exact steps need to be accomplished. There are usually many different ways of accomplishing the same task. For example, suppose we wished to inspect several points on a part as called out in the CAD STEP file Part.stp. Either one of the following approaches will work, but one may be more advantageous than the other depending on the constraints of what you're trying to accomplish. These are just two options, in reality this could be accomplished in about a dozen or more different ways.

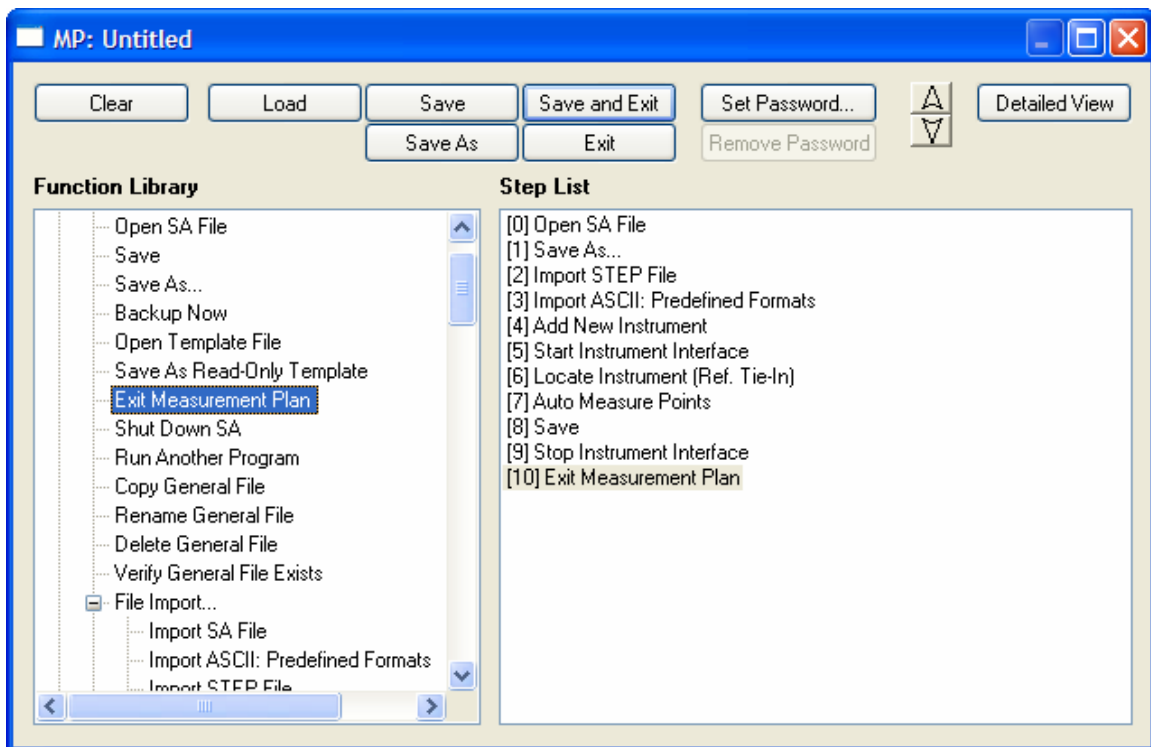
Option A (all MP)

Open an MP edit session and add the following functions from the left tree view to the script.

Operation	Function Branch	MP Function
Open a new blank file	File Operations	New SA File
Rename it so it gets saved as MyFile.xit	File Operations	Save As...
Import the CAD file [cad model.Stp]	File Operations	Import STEP File
Import the “nominal” point group from the [nominal points.txt] file.	File Operations	Import ASCII: Predefined Formats
Add an SMX tracker to the Job	Instruments	Add New Instrument
Start the tracker interface	Instruments	Start Instrument Interface
Locate the instrument relative to the part by measuring some known points on the part	Instruments	Locate Instrument (Ref. Tie-In)
Measure the remaining points to be inspected	Instruments	Auto Measure Points
Save the file	File Operations	Save
Close the tracker interface	Instruments	Stop Instrument Interface
Close the MP	File Operations	Exit Measurement Plan

MP Steps and the Branch for Script

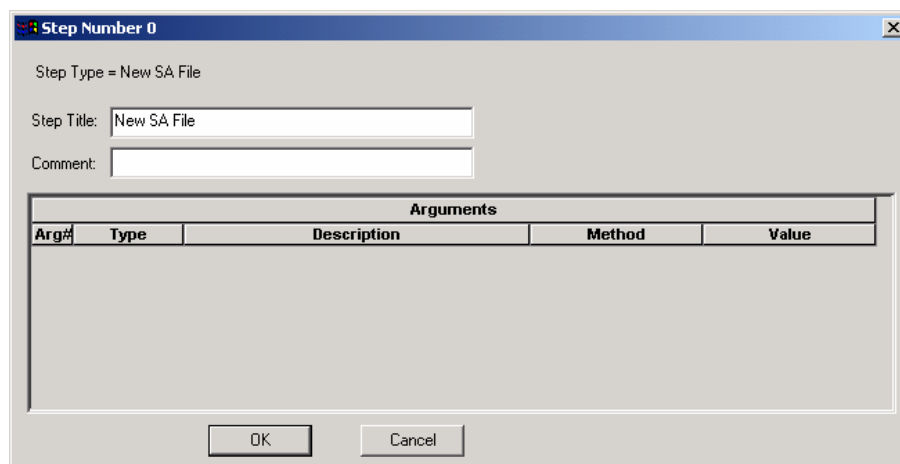
Save the MP command list at [MP_A.mp] in the [Analyzer Data\Script] directory. The figure below show the initial script as it should appear.




MP Editor Environment with the Script Steps

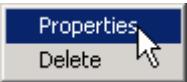
Before this script can execute plan we need to specify the specific elements within the steps. To set the step properties you can either double click on the step or right click on each step and the select the properties options.

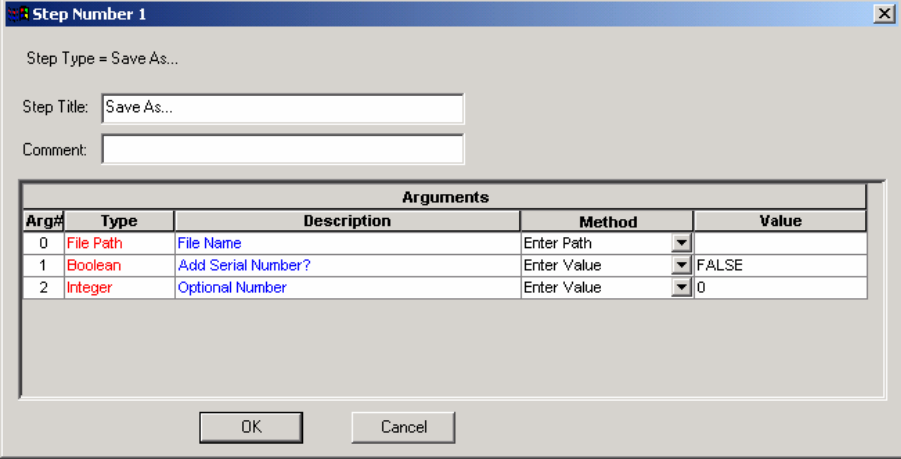
Open the properties page for step 0 by double clicking on the [0] New SA File step within the plan. The properties page for this step has the basic components. The title bar shows the Step Number of 0. The Step Title shows the default title but this attribute can be edited to show the specific purpose for the step. A comment field for the step is also provided. These two fields appear on the details view to be printed.



First MP Step Properties Dialog

This particular step does not have any optional arguments. The next step in the MP has a few arguments ... let's look at it next. Right-click on the  [1] Save As... and select the

[Properties] option. . The properties dialog for the Save As step shows up as below.



Step Type = Save As...

Step Title:

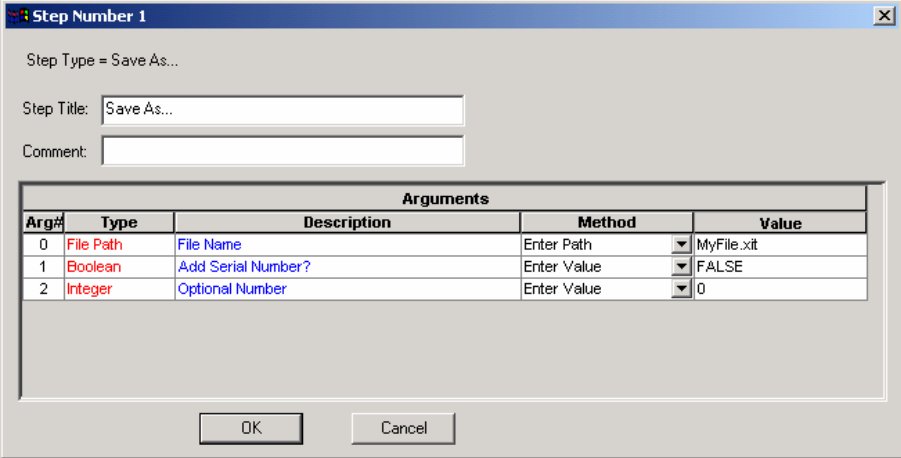
Comment:

Arg#	Type	Description	Method	Value
0	File Path	File Name	Enter Path	
1	Boolean	Add Serial Number?	Enter Value	FALSE
2	Integer	Optional Number	Enter Value	0

OK Cancel

Empty 'Save As' MP Step Properties Page

The three items in the arguments area of the dialog need to be completed before the Save As step will execute and save the job with the name that we need. Enter a file name of MyFile.xit. In this case the serial number is an option that can be left at FALSE and with that the Optional Number can also be left at 0. The completed properties dialog look like the following.



Step Type = Save As...

Step Title:

Comment:

Arg#	Type	Description	Method	Value
0	File Path	File Name	Enter Path	MyFile.xit
1	Boolean	Add Serial Number?	Enter Value	FALSE
2	Integer	Optional Number	Enter Value	0

OK Cancel

Enter Path selection method for 'Save As' MP Step Properties Page

The completed steps 2 through 10 follow.

Step Number 2

Step Type = Import STEP File


Step Title:

Comment:

Arg#	Type	Description	Method	Value
0	File Path	STEP File Path	Enter Path	..\cad model.Stp
1	Boolean	Display Entity Filters	Enter Value	FALSE

OK Cancel

Default 'Import STEP File' MP Step Properties Page


Step 2's argument 0 above shows one option to specify file path to the [cad model.stp] file. A better Method for you to use is to specifically locate the file on your computer. To help you accomplish that task choose the Browse method as shown below. Select Browse from the pull down options in the Method column and then select the option button  that appears in the Value column for argument 0. The arrow in the figure below is pointing to the option button.

Step Number 2

Step Type = Import STEP File

Step Title:

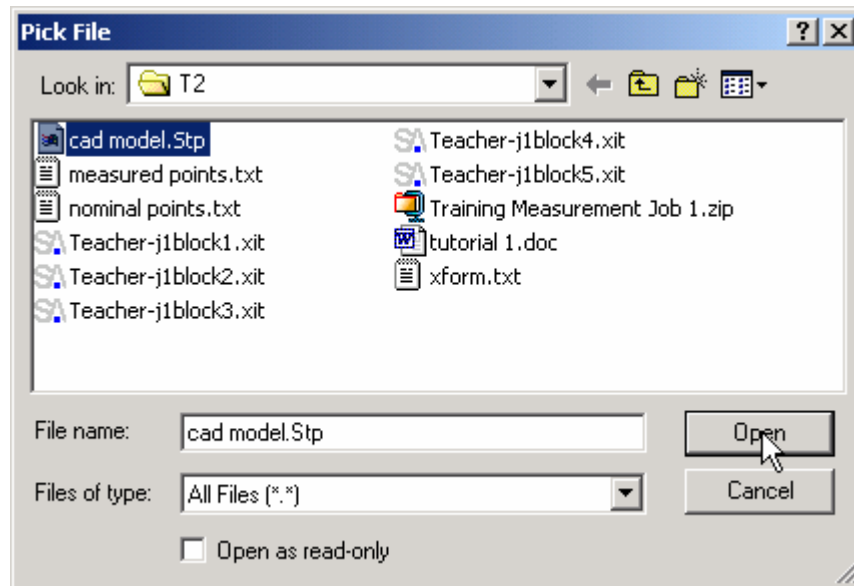
Comment:

Arg#	Type	Description	Method	Value
0	File Path	STEP File Path	Browse	
1	Boolean	Display Entity Filters	Enter Value	FALSE

OK Cancel

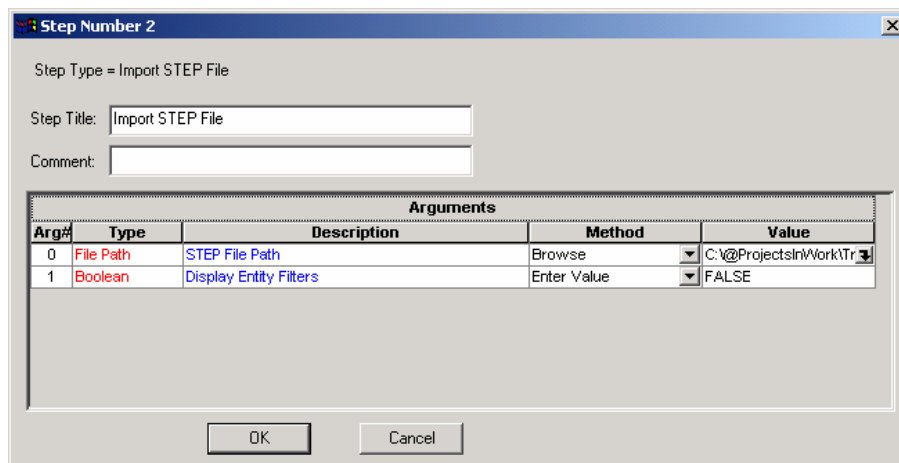
Completed 'Import STEP File' MP Step Properties Page

After selecting the option button a classic open file dialog opens and it helps you to navigate your way to the file on your computer. Select the Open button when you have navigated and selected the [cad model.Stp] file as shown below.



Pick File dialog for 'Import STEP File' Browse selection method

The completed step 2 properties dialog appears as below.



Completed 'Import Step File' Properties Page

The complete step 3 properties dialog is shown below. Note the Browse method is used to locate the ASCII file on the computer.

Step Type = Import ASCII: Predefined Formats

Step Title:

Comment:

Arg#	Type	Description	Method	Value
0	File Path	ASCII File Path	Browse	C:\@Projects\InWork\Tr...
1	File Format	File Format	Pick	PointName X Y Z
2	Units	Units	Pick	Inches
3	Group Name	Group Name	Enter Value	nominals
4	Boolean	Import as Cloud	Enter Value	FALSE

OK Cancel

Completed 'Import ASCII Predefined Formats' Properties Page

Step 3 has 5 arguments (numbered 0 through 4). The File Format is selected by picking from a list of supported formats. Having the actual list of all the supported formats makes it easy to associate the file being imported with the format that will be used to interpret it. For the [nominal points.txt] file select the [PointName X Y Z] format from the list of available options. Argument 2 allows you to specify the units that the incoming file is in so that it can be scaled to match the job's current unit selection.

The step then lets you specify with argument 3 what group that you would like to put the incoming points into. Enter [nominals] as the Group Name for this argument. This value you enter into the Group Name argument will be referenced in subsequent steps. Referencing the group name from other step means that the group name will only have to enter it once and then just referred too many times later in the process. That helps keep the typing issues to a minimum.

The last argument allows you import the points into an SA cloud object. Since we aren't working with cloud objects in this project argument 4 can be left with the default value of FALSE.

In step 4 we add an instrument into the job. Argument 0 lets you specify which instrument type to add to the job. Select the 3D instrument that you are most familiar with from the list provided in the value field for argument 0. Argument 1 is grayed out because it is a "Return Only" value. This argument will actually have information put into it by SA, so that subsequent steps can refer back to it. In this case the instrument that is added is assigned a number by SA. By using that number in subsequent steps SA will know that we mean that specific instrument.

Step Number 4

Step Type = Add New Instrument


Step Title:

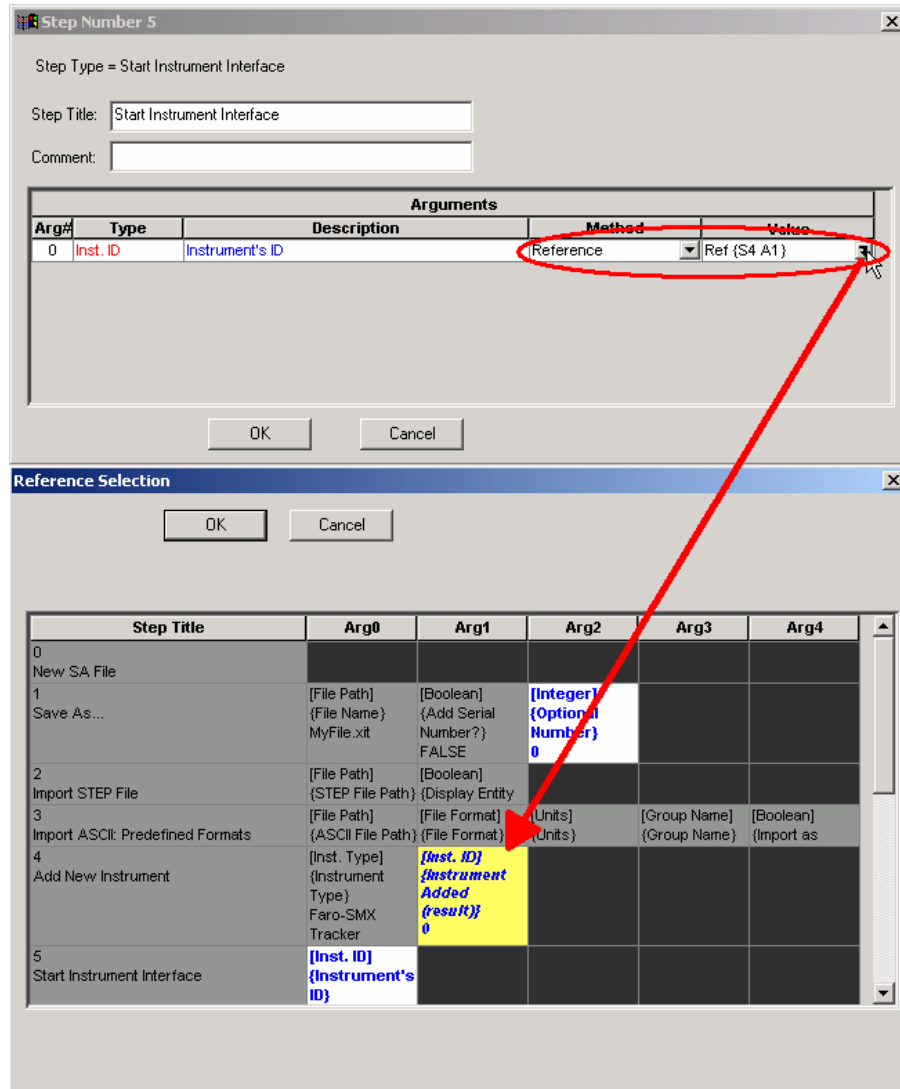
Comment:

Arg#	Type	Description	Method	Value
0	Inst. Type	Instrument Type	Current List	Faro-SMX Tracker
1	Inst. ID	Instrument Added (result)	Result Only	Result Only

OK Cancel

Completed 'Add New Instrument' Properties Page

In step 5 we start the instrument interface. In order to know which instrument to start we will reference the instrument number that SA assigned to instrument added in step 4. In argument 0 select the Reference Method. Then select the  button to bring up the Reference grid of all the available steps and there arguments as shown in the figure below. Navigate to Step 4's argument 1 and select it. By referencing this cell, the Start Instrument Interface step will always start the instrument added in step 4 regardless of how many other instruments had already been added into the job. The figure below shows the proper cell to select.



Start Instrument Interface by referencing the instrument number

Please note that when using a reference method the value show as a Ref {Step number Argument number}.

Step 6 is a function that accomplished quite a bit in one step. It directs the instrument added in step 4 to measure and transform on the points added in step 3. See the figure below to help coordinate the proper references. The step then asks for the measured groups name and whether the points should be auto-measured.

Step Number 7

Step Type = Auto Measure Points

Step Title:

Comment:

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument ID	Reference	Ref {S4 A1}
1	Group Name	Reference Group Name	Reference	Ref {S3 A3}
2	Group Name	Actuals Group Name (to be measured)	Enter Value	meas2

OK Cancel

Auto-measure Points MP Step Properties Page

Step 8 is a simple step that saves the job with the current job name.

Step Number 8

Step Type = Save

Step Title:

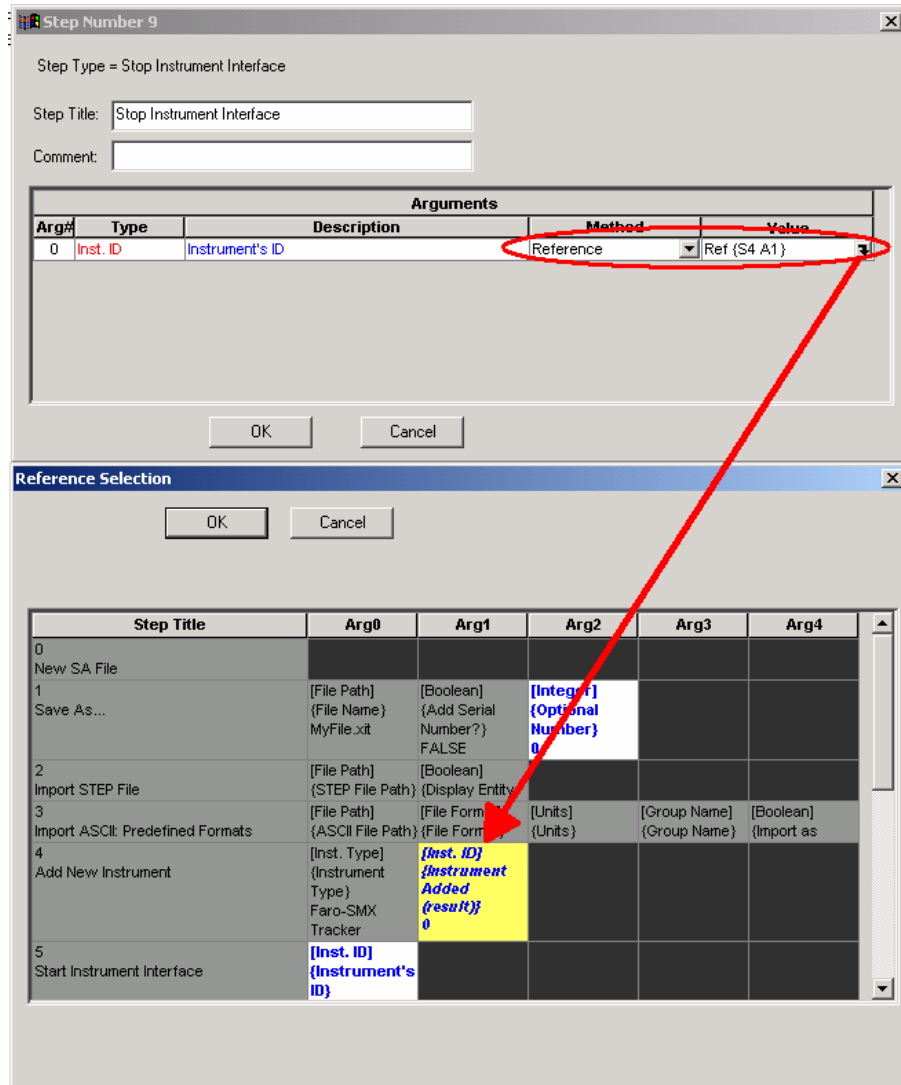
Comment:

Arguments				
Arg#	Type	Description	Method	Value

OK Cancel

Save MP Step Properties Page

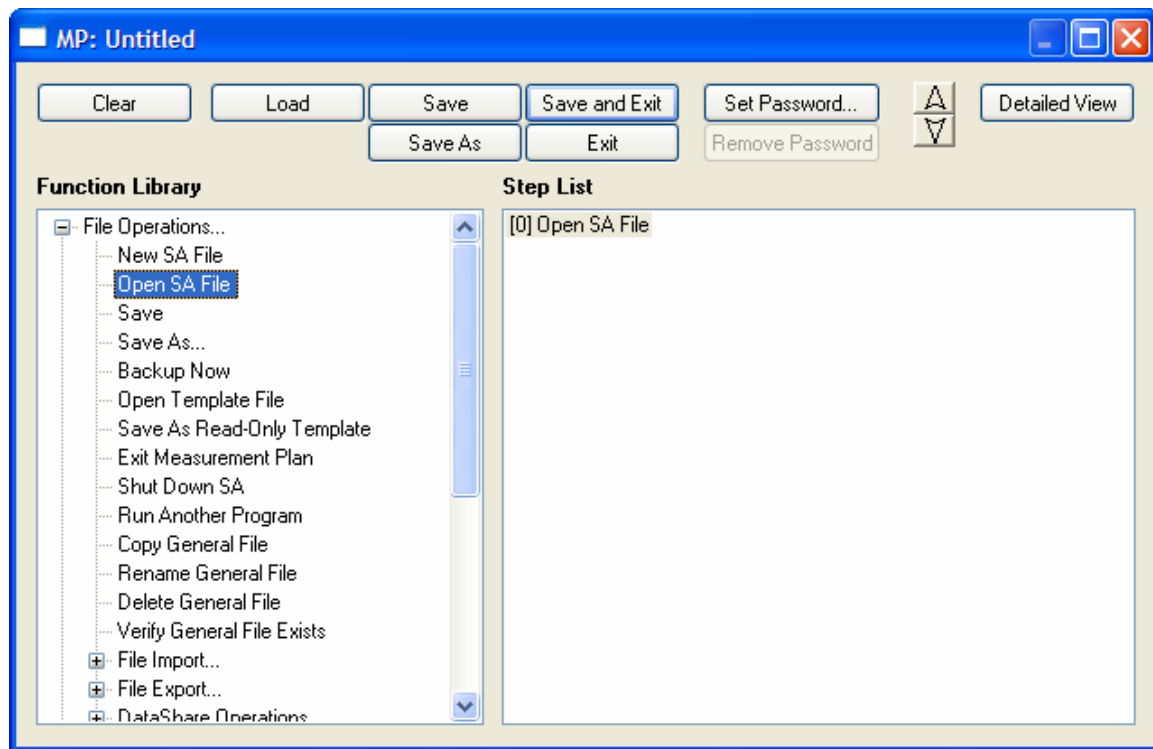
Stopping the instrument interface is accomplished with step 9. The instrument ID is referenced all the way back to when it was created in the Add Instrument step (number 4). The figure below shows the reference method and the Reference selection grid.



Stop Instrument Interface MP Step Properties Page

The last step in this MP actually shuts down the measurement plan.

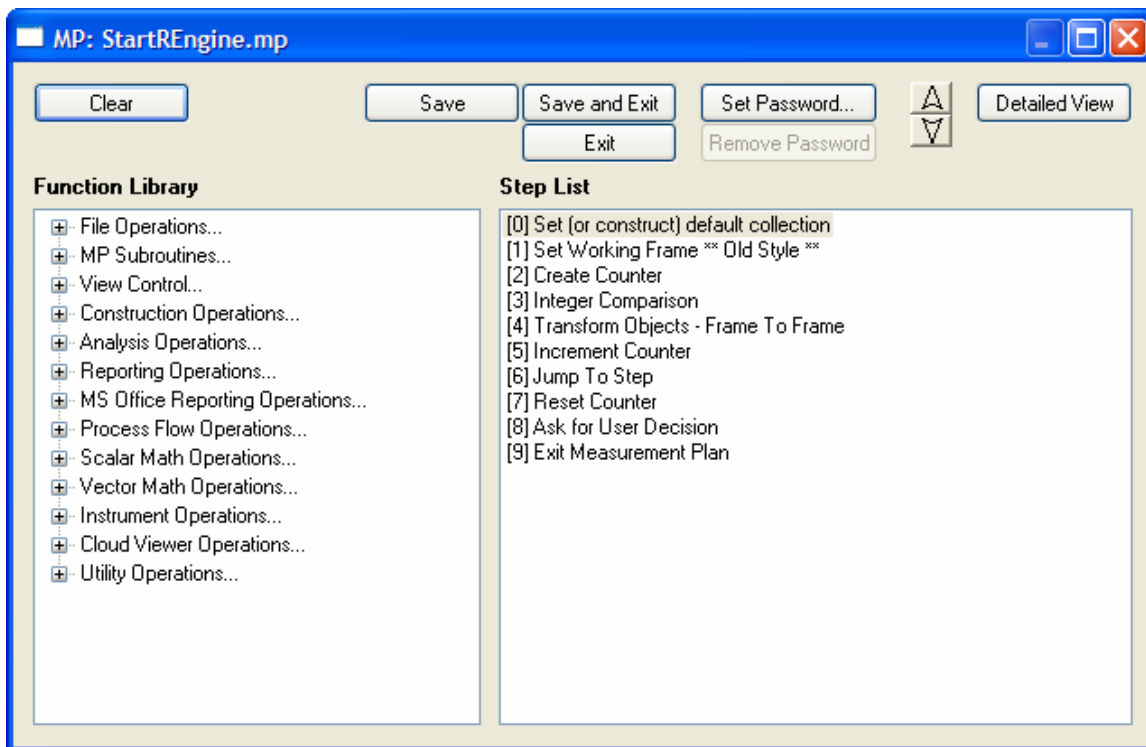
We begin by finding the Open **SpatialAnalyzer** file item in the file operations area of the left pane. If you double click on that entry, it will get added to a list in the right pane. See below:



Open SA File MP Step

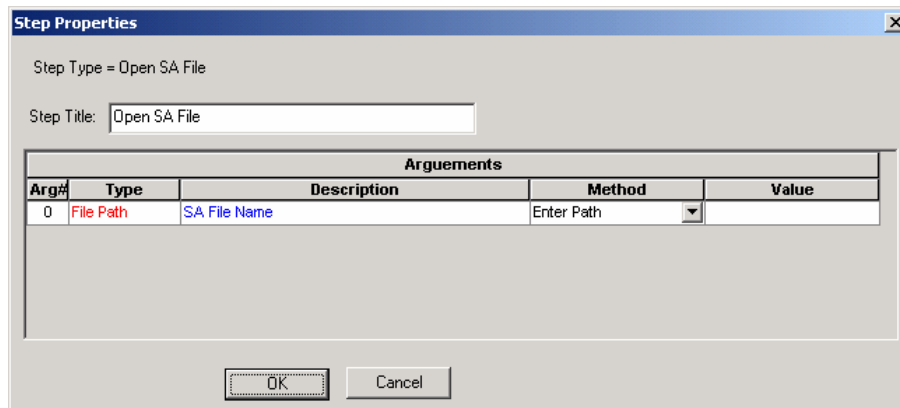
It is usually best just to outline the measurement process with a list of steps like we did under the heading Option B. With that in mind we can just add each step to the list in the right pane. If you need to insert a step, just right click on the item instead of double clicking and you'll see options to insert before or after the highlighted step in the right pane.

At the completion of this process we have an outline that looks like the following:



MP Editor

Now we're ready to start filling in more of the details for our MP. Let's begin with the first step and go sequentially through the program. If you double click the first item in the right pane list view, you'll see the following step properties dialog.



MP Step Properties Dialog

This dialog provides us with general information about the step, the type of arguments it needs and allows us to set the values of those arguments. For the open **SpatialAnalyzer** file step we see there is only one argument that we can change - the name of the file we wish to open. To do this we can choose one of several methods for assigning a value. We can enter a path just by typing it in the value column, or we could browse for a path and select it with a typical windows file open dialog, or we could refer to another step

argument. We will discuss references in greater detail later on in an advanced MP section. We should also note that we have the ability to change the step title as well. This controls what text the operator will see when executing the MP. Getting back to our example, the step property dialog might look like this when we're done editing.

Step Type = Open SA File

Step Title:

Arguments				
Arg#	Type	Description	Method	Value
0	File Path	SA File Name	Browse	C:\Analyzer Data\MyFile.xit

OK Cancel

Open SA File Step with a Path

Notice that when you accept these options the text in the list view gets modified to reflect the new step title.

Proceeding to the next step – Start instrument interface, we see the following arguments.

Step Type = Start Instrument Interface

Step Title:

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument's ID	Enter Value	0

OK Cancel

Starting an Instrument Interface Step

This is indicating that the only argument for the start interface command is the **SpatialAnalyzer**. Instrument ID. In this case, we had a .xit file where only one instrument was added an SMX as instrument 0. If we had two or three instruments in the job, we would need to identify which instrument's interface we wished to start. Notice that we did not have to identify the type of instrument. This is because we started with a file that already had the tracker added to the job. **SpatialAnalyzer** already knows that it's an SMX tracker.

Continuing to the next step, locate the instrument, we see a step property dialog with five arguments. We need to identify the instrument to be located, the name of the group containing the reference nominals from the file, the name of the group to be created that will contain the actual measured points, a tolerance to flag errors if deviations are greater than this amount, and a flag to indicate whether the interface should attempt to automatically survey the points. As in the first dialog, we can change the title of the step and specify arguments through any number of different methods. Different argument types will have different possible methods for assigning values.

The 'Step Properties' dialog box shows the following configuration:

- Step Type = Locate Instrument (Ref. Tie-In)
- Step Title: Locate Instrument (Ref. Tie-In)

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument to Locate	Enter Value	0
1	Group Name	Reference Group Name	Enter Value	
2	Group Name	Actuals Group Name (to be measured)	Enter Value	
3	Double	Tolerance	Enter Value	0.000000
4	Boolean	Auto Survey	Enter Value	FALSE

Buttons: OK, Cancel

Default Locate Instrument MP Step Properties

If we complete this dialog according to our stated objectives, we might end up with something like the following:

The 'Step Properties' dialog box shows the following configuration:

- Step Type = Locate Instrument (Ref. Tie-In)
- Step Title: Locate SMX Tracker

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument to Locate	Enter Value	0
1	Group Name	Reference Group Name	Enter Value	CAD Nominals
2	Group Name	Actuals Group Name (to be measured)	Enter Value	Actuals
3	Double	Tolerance	Enter Value	0.030000
4	Boolean	Auto Survey	Enter Value	FALSE

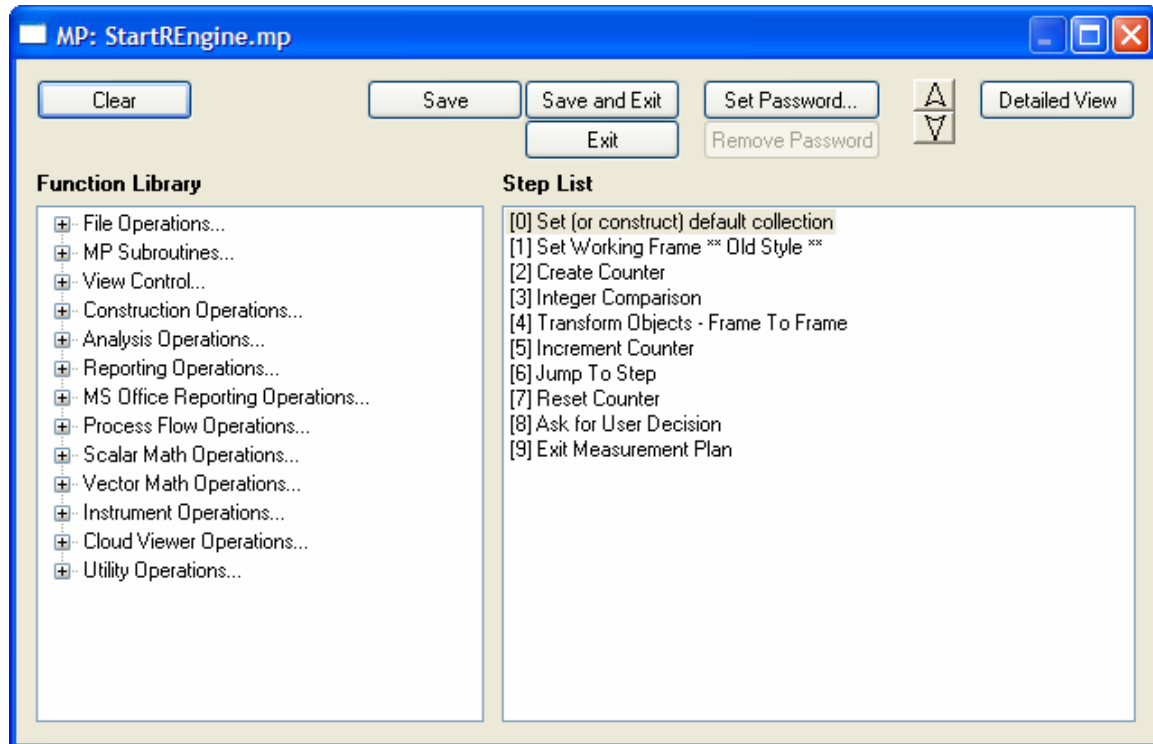
Buttons: OK, Cancel

Completed Locate Instrument MP Step Properties

In the interest of brevity, we'll skip delving into each of the remaining steps. It's sufficient to say that all steps are populated in a similar manner.

Note that from the main MP editor, if you need to change the position of step within the MP you can highlight a step and hit the up or down arrows to promote or demote the step. Also highlighted steps can be deleted using the delete button.

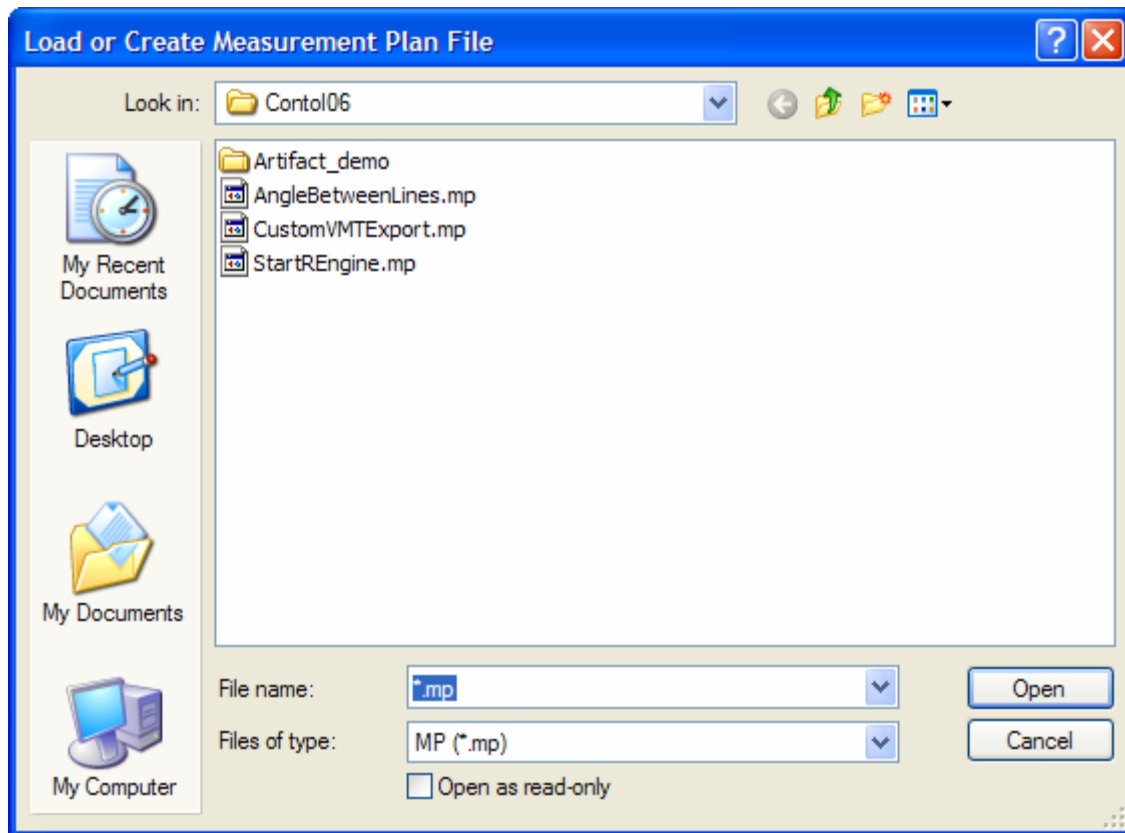
When you're done editing a plan as shown below simply save the MP file as Test.mp and exit the editor.



Completed Basic MP

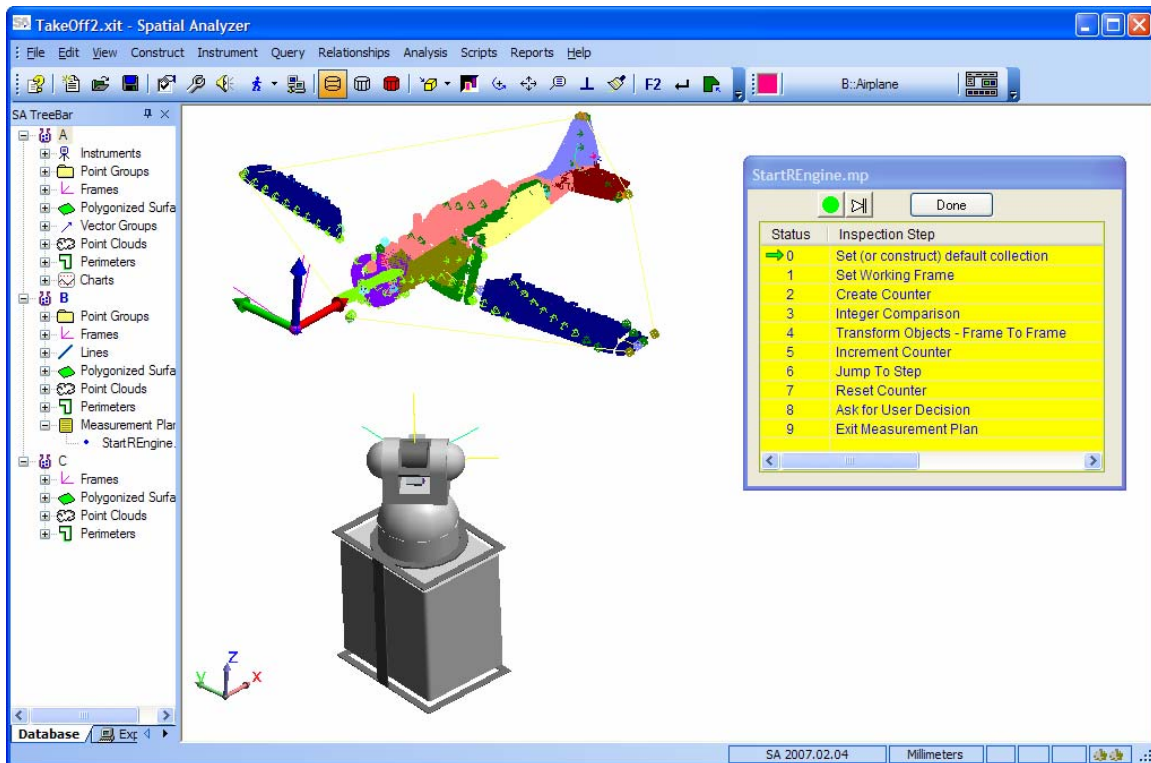
Executing the basic MP

Now that we have constructed a basic plan (Test.mp), let's look at what a user sees when they execute this plan. To start the process the user needs to run the MP. This is done by selecting the File->Run MP option from the main menu. After the menu selection, the user will be prompted for what plan to execute with the typical windows file open dialog shown below.





File Open Dialog to Load an MP

After selecting the appropriate file the user will see the MP interface along with the regular **SpatialAnalyzer** environment.



MP Loaded into the MP Interface

To begin execution of a plan, users can single-step through the plan by hitting the single step  button. Or alternatively, users could hit the go button  to just run the plan without stopping. Depending on options set by the MP, the continuous execution of the program will either halt or continue at the user's or program designer's discretion.

Advanced MP concepts

This section will present several advanced topics for using MPs within **SpatialAnalyzer**. None of these topics are necessarily complicated but familiarity with MP basics is essential before attempting these concepts.

Executing from a command Line

We saw earlier how MPs can be started by a user. Sometimes, it's desirable to even bypass this phase and have **SpatialAnalyzer** start-up and begin running an MP immediately. This is done by starting **SpatialAnalyzer** with command line arguments. This is easiest to do in windows by either by making a shortcut to **SpatialAnalyzer** on the desktop and altering the default options for that shortcut or by making a single line batch file that starts **SpatialAnalyzer**. We will discuss both here briefly.

To make a shortcut simply right click on the **SpatialAnalyzer** program in the start bar and drag the icon to the machine desktop. When you release the mouse button, select create shortcut. You will see the shortcut appear. If you right click on the desktop icon and select properties you will see that the target path should be something like

"C:\Program Files\New River Kinematics\SpatialAnalyzer 2001.10.11\Spatial Analyzer.exe"

To add command line arguments just add the following arguments to the end of the string as follows:

/MP "C:\Analyzer Data\Test.mp" "C:\Analyzer data\testfit.xit"

The /MP argument tells **SpatialAnalyzer** to load and start executing the named MP file, and the following .xit path tells **SpatialAnalyzer** to load this .xit when starting SA.

Be careful with this method to update your shortcuts when you install a new **SpatialAnalyzer** version (if you wish to upgrade the MP as well).

To make a single line batch program just use the Notepad editor to enter the same strings presented above.

MP Process Flow Checks

The simple MPs that we have looked at thus far have all started on step 0 and proceeded through until the end of the plan. In many instances we need the MP to react to certain conditions and branch to alternate steps. These functions are handled in MP by the Process Flow functions. Process flow checks can be as simple as looking at the status condition of another step. For example we might make an MP that checks for drift points to see if the instrument has been inadvertently moved. We may want the program to continue measuring if the drift check didn't have errors and we may want to halt or jump to another step if it did have errors. Another typical example of process flow control is to branch based on logical comparisons. We could for instance, compare the RMS residuals from a fit to an input value. We may want to go to alternate steps depending on whether the RMS value was greater than, or less than the specified amount. Other process flow step types include branching to other MP programs, exiting the current MP, or shutting down **SpatialAnalyzer** altogether.

References to other arguments

We mentioned earlier that it is possible to not only specify an argument to an MP step, but also that it is possible to refer to the argument of another step. This gives MP an extremely powerful way of adapting to varying runtime conditions. You could for example have an MP that creates a group as part of a measurement process. If we wished to perform an analysis on this group later on in the MP we could refer to the group name from the prior step. This makes it easy to ask a user for a simple group name like "flange" and perform an analysis on that group later by name, perhaps a circle fit to construct "flange circle". The next time through the program the operator might want to name the group "left-flange", then the resulting circle would be named "left-flange circle". References can do far more than just simple string manipulation, but the concepts are the same regardless of the data type. To assign a reference just select "Reference" in the method pull-down for an argument. You will see a hot spot control appear in the values column for that argument. When you hit the hot spot control you will see an argument browser that allows you to select arguments from other steps that are of compatible data types. As an example if we select the instrument ID in step 5 of our sample program by reference, we see the following dialog. The yellow cell indicates the currently referenced cell, and white cells are legitimate choices. Invalid choices are grayed out and not selectable.

Reference Selection					
<div>OK</div> <div>Cancel</div>					
Step Title	Arg0	Arg1	Arg2	Arg3	Arg4
0 Open MyFile.xit	[File Path] C:\Analyzer				
1 Start SMX Tracker Interface	[Inst. ID] 0				
2 Locate SMX Tracker	[Inst. ID] 0	[Group Name] CAD Nominals	[Group Name] Actuals	[Double] 0.030000	[Boolean] FALSE
3 Auto Measure Points on Flange	[Inst. ID] 0	[Group Name] Design	[Group Name] Flange Pts		
4 Save File					
5 Shut Down SMX Interface	[Inst. ID]				
6 Exit Measurement Plan					

File MP Step Reference Matrix

Sharing data between plans and with other applications

Many tasks are sufficiently accomplished by a single, self contained MP. However, there are circumstances where it is desirable to pass information into or out of a single MP or, to allow multiple MP's to share information between them. This is accomplished through the use of DataShare files.

By using the MP function an MP designer can now create a step in which data external to the MP can be referenced in future MP steps. As an example consider the case where a specified CAD file may change from run to run of a complicated MP, perhaps also the resulting **SpatialAnalyzer** filepath for an MP job is to be drawn from a database or excel file. Lets further suppose that we wish to have tolerances also change depending on the part being inspected. These paths and tolerances have to be somehow passed to the MP. To accomplish this, a DataShare file is created that contains, among other items, the two filepaths as text strings and the desired tolerance as a double precision number. By loading the DataShare file early in the MP, the designer may in a later step use the function with a reference to the field created by the DataShare. Likewise a fit function could use a reference to the tolerance double of the Datashare file for the fit tolerance.

DataShare files can also be used to export data to other MPs or other processes. We could for example write the RMS residuals from a fit to a datashare file so that it can be used in some other process external to **SpatialAnalyzer**. To write a datashare file a MP should include a command that allows the user to write user specified data to share with other applications or other MPs.

Loading a DataShare File into MP

To load a DataShare file into memory simply insert the into an MP. It's argument options appear as follows:

The screenshot shows the 'Step Properties' dialog box. At the top, it says 'Step Type = Load DataShare File'. Below that, 'Step Title:' is followed by a text box containing 'Load DataShare File'. To the right of this text box is a button labeled 'Refresh Arguments'. Below these elements is a table with the heading 'Arguments'. The table has five columns: 'Arg#', 'Type', 'Description', 'Method', and 'Value'. There is one row in the table with 'Arg#' 0, 'Type' 'File Path', 'Description' 'DataShare File Path', 'Method' 'Enter Path', and 'Value' is empty. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Arg#	Type	Description	Method	Value
0	File Path	DataShare File Path	Enter Path	

Default Load DataShare Step Properties Dialog

The MP designer may now type in a path for the filename, assign a reference to another path, or browse for a desired DataShare file for argument 0. Once a valid path is entered, the designer should hit the Refresh Arguments buttons. The Step Properties Dialog will then appear as follows:

The screenshot shows the 'Step Properties' dialog box after the 'Refresh Arguments' button has been clicked. The 'Step Type' and 'Step Title' remain the same. The 'Arguments' table now has four rows. The first row is the same as before. The second row has 'Arg#' 1, 'Type' 'Double', 'Description' 'TestDouble', 'Method' 'Enter Value', and 'Value' '5.000000'. The third row has 'Arg#' 2, 'Type' 'Integer', 'Description' 'TestInt', 'Method' 'Enter Value', and 'Value' '8'. The fourth row has 'Arg#' 3, 'Type' 'String', 'Description' 'TestString', 'Method' 'Enter Value', and 'Value' 'TEST'. The 'Refresh Arguments' button is now disabled. 'OK' and 'Cancel' buttons are at the bottom.

Arg#	Type	Description	Method	Value
0	File Path	DataShare File Path	Browse	C:\@junk\test.bds
1	Double	TestDouble	Enter Value	5.000000
2	Integer	TestInt	Enter Value	8
3	String	TestString	Enter Value	TEST

Load DataShare Step Properties with a valid Path

All the values presented are read directly from the specified DataShare file.

Caution: The DataShareFile will be re-read with the execution of the MP. This is necessary to facilitate sharing dynamic data. For example, Argument 2 might have a value of 8 for one run and a value of 9 for the next. **MPs that use a specified DataShare file must use the same format file (order of arguments) for each run.** Thus we could not make an MP where the double of Argument 1 and the Integer of argument 2 swap positions within the file. Future references would point to the wrong data type!

Writing a DataShare File from MP

To write a DataShare file to disk simply insert the into an MP. Its argument options appear as follows:

The 'Step Properties' dialog box shows the configuration for a 'Save DataShare File' step. It includes a 'Step Title' field, 'Add Argument' and 'Remove Argument' buttons, and a table of arguments.

Arg#	Type	Description	Method	Value
0	File Path	DataShare File Path	Enter Path	
1	Boolean	Save in Binary Format?	Enter Value	FALSE

Save DataShare Step Properties

The MP designer may now type in a path for the filename, assign a reference to another path, or browse for a desired DataShare file for argument 0. This is the location the file will be written to. Additionally, the designer must specify whether the file to be written will be in Ascii or Binary formats. Now the designer needs to determine exactly what they wish to export from the MP. To add a field, simply hit the Add Argument button. The following dialog will appear:

The 'Add Argument' dialog box allows the user to specify the type and descriptor of a new argument.

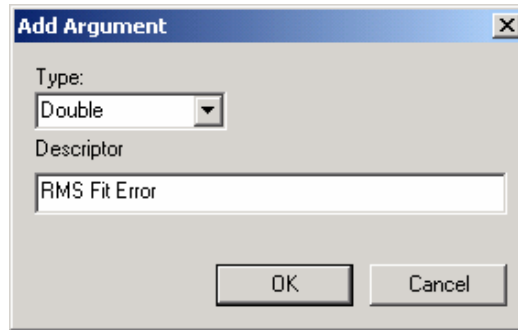
Type:

Descriptor:

Buttons: OK, Cancel

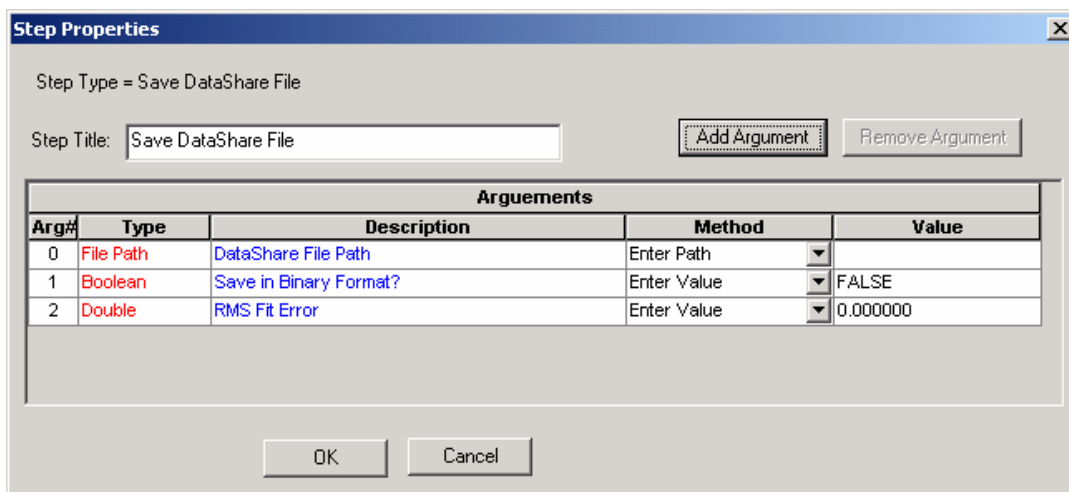
Default Add an Argument to a DataShare Dialog

The Type pull-down allows the designer to specify what type of argument they wish to add to the export list. Valid types are currently Integer, Double, String, IntegerArray, DoubleArray, and String Array. The designer then must enter a Descriptor for the argument added. For example, we could add a double to represent the RMS fit error as follows:



Add a double to a DataShare Dialog

After accepting this dialog, the Step Properties dialog appears as follows:



Save DataShare Step Properties with an additional Field

Notice that Argument 2 was added. The designer may then assign a value to that argument of reference another double within the MP. To remove an argument, highlight the desired argument by clicking on its row and hit the Remove Argument button. The first two arguments are protected and cannot be removed.

Database (ODBC) Functions with MP

MP Database (ODBC) functions allow putting data to and getting data from any ODBC-capable database. They allow the MP programmer to specify an ODBC connection string (which identifies the database driver, the specific database, and any login info), table name, and a set of columns with values to be put to the database or a set of columns to be retrieved from the database. An SQL WHERE clause can also be used which allows updating existing records in the database and which is used when retrieving data from the database.

This guide was written using the Firebird database v2.0 (<http://www.firebirdsql.org/>) and includes installation and setup instructions for that database product. A section below

shows how to download and install an example database. The next section assumes that operation has been done for the Firebird database or something similar

MP Creation

Start SA and create a new embedded MP (Scripts>>Embedded Measurement Plans>>Create New, enter a name and click OK). This will create an MP embedded in the current job file and will then open the new MP for editing.

From the Function Library, double-click on File Operations...>>Database Operations...>>Put to ODBC Database to add the function to the MP as a new step. Double-click on the newly added MP step to open the Step properties window.

Note that this guide uses the interactive picking mode for setting up the MP arguments. This mode requires having access to the database at the time of MP creation/editing. If the database is not available at the time of MP creation/editing, you can instead just type the values in directly, however, you'll want to make sure the values match what will be present for the database.

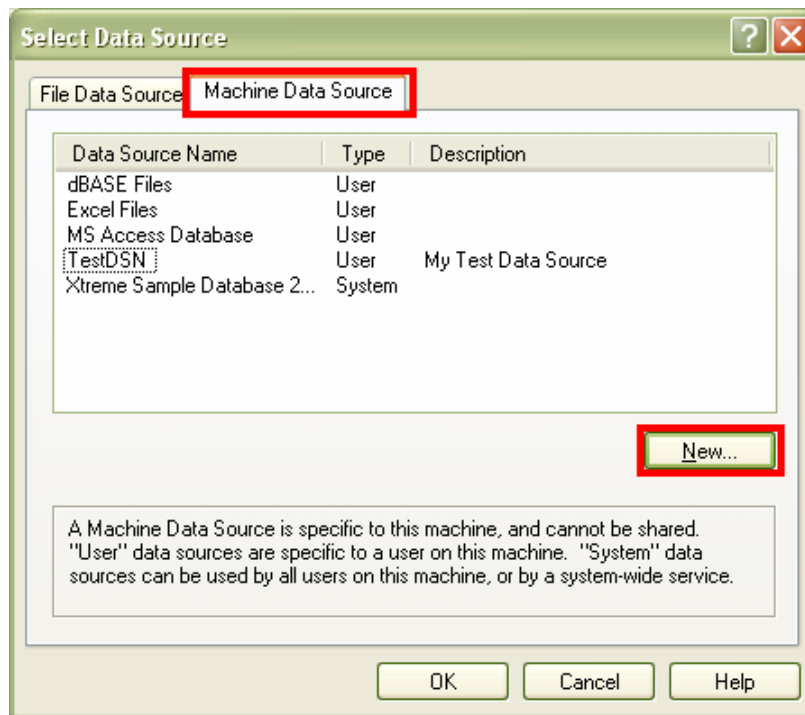
For the connection string argument, change the Method to Browse and then click the hotspot button which should appear.

The screenshot shows a dialog box titled "Step Number 0" with a close button in the top right corner. Inside the dialog, the "Step Type" is set to "Put to ODBC Database". Below this, there are input fields for "Step Title" (containing "Put to ODBC Database") and "Comment". To the right of these fields are buttons for "Add", "Remove", "Edit", and "Add using Column Picker...". Below the input fields is a table with the following structure:

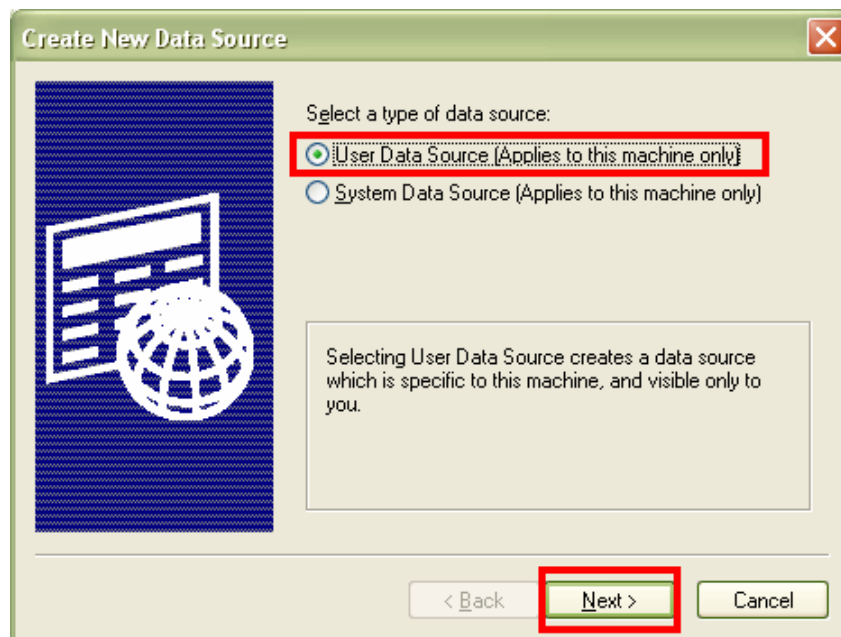
Arguments				
Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	
1	Table Name	Table Name	Enter Table Name	

Red boxes highlight the "Browse" method for the first argument and a small icon in the "Value" column of the first row. At the bottom of the dialog are "OK" and "Cancel" buttons.

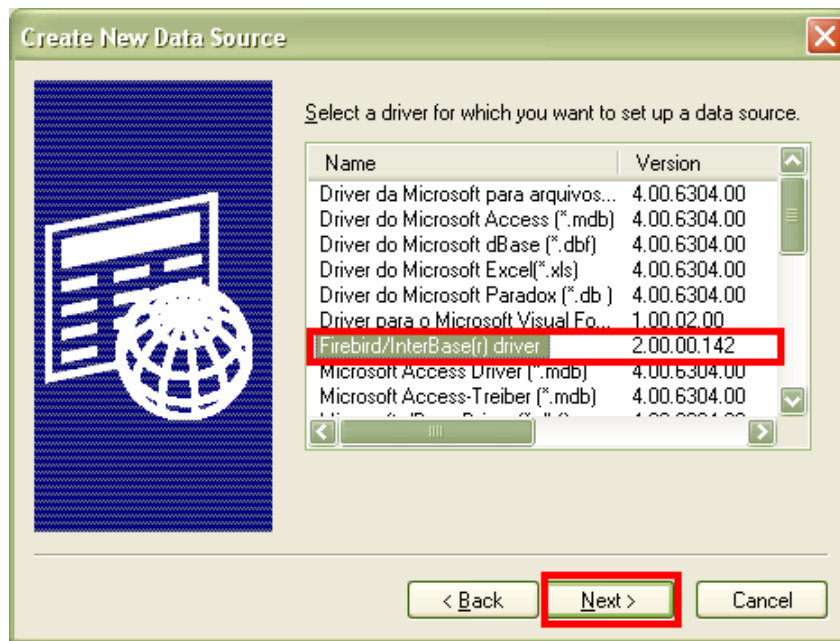
This should open up the Windows ODBC data source picker. Alternatively, you could manually type in a connection string. In the Select Data Source window, click on the Machine Data Source tab, then click the New button.



This will open the Create New Data Source wizard. In the first screen, select User Data Source, and then click Next.



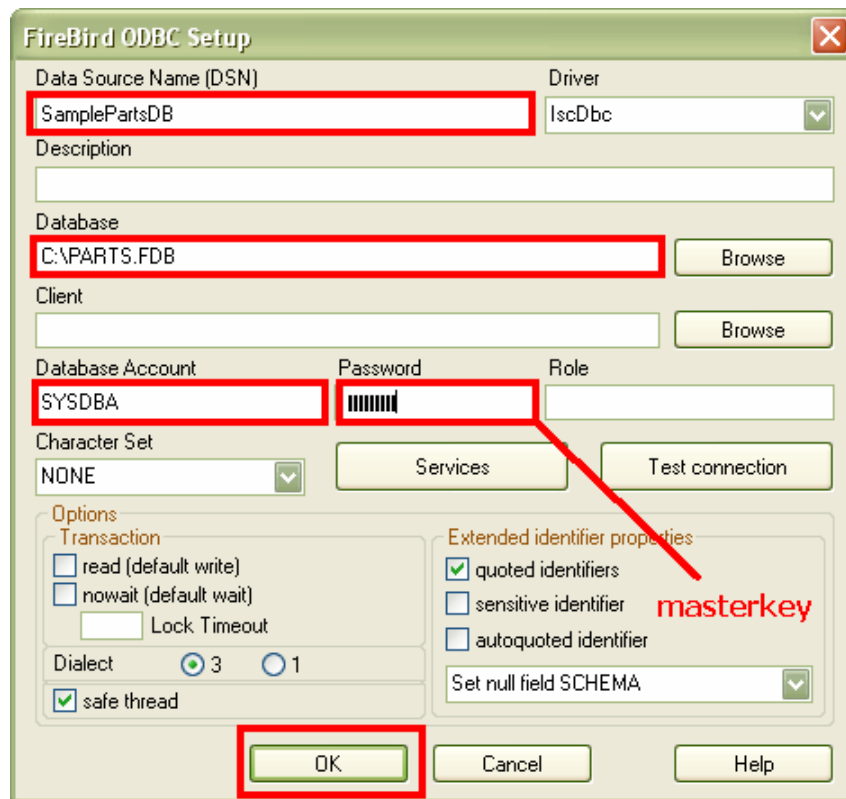
Select the "Firebird/Interbase® driver" driver and then click Next.



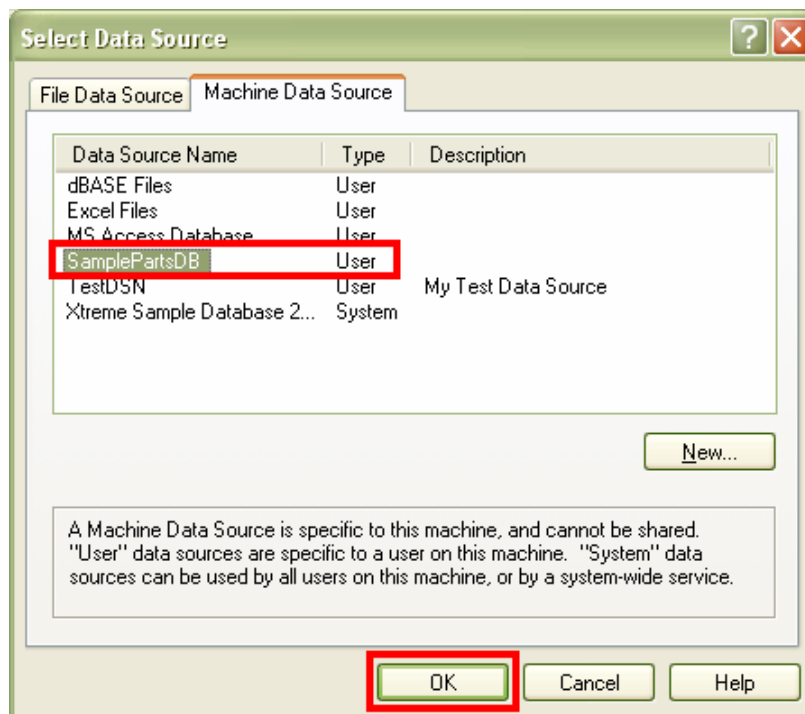
This will display the final page in the wizard. Review the information and then click Finish.



The FireBird ODBC Setup window will now be displayed, which allows you to enter the connection information for the database. Enter a data source a name, and provide the connection info as shown below and then click OK.



The newly added data source should now be shown in the Select Data Source window. Now that the data source has been created, you can just pick it from this list for future MP editing. Select the SamplePartsDB data source and click OK.



Verify that the connection string argument has been set.

Step Number 0

Step Type = Put to ODBC Database

Step Title: Put to ODBC Database

Comment:

Arguments

Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Enter Table Name	

OK Cancel

The complete connection string is shown below. As is evident from the below, it is usually much easier to browse for the string than to manually enter the string.

```
DSN=SamplePartsDB;Driver=Firebird/InterBase(r)
driver;Dname=C:\PARTS.FDB;CHARSET=NONE;PWD=masterkey;
UID=SYSDBA
```

For the Table Name argument, change the Method to Browse and then click the hotspot which should appear.

Step Number 0

Step Type = Put to ODBC Database

Step Title: Put to ODBC Database

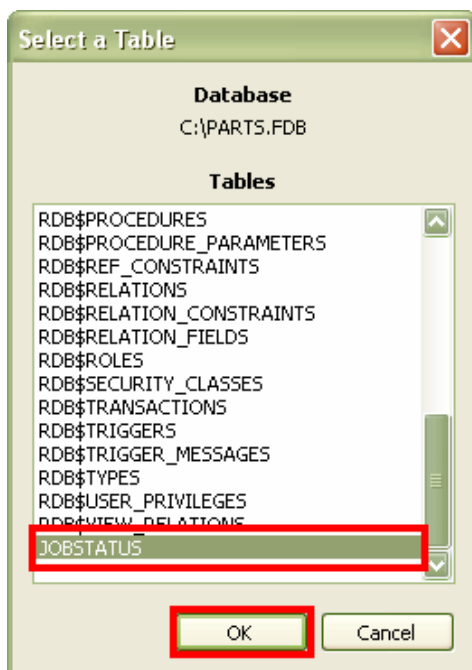
Comment:

Arguments

Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Browse	

OK Cancel

This will open the Table Selection window. Select the sample table created earlier and then click OK. You can ignore the RDB\$ tables. These are system tables created by Firebird.



Verify that the Table Name argument has been set.



Inserting New Records

At this point, we have enough information in this step to connect to the database and direct our attention to a particular table in that database. It is now necessary to add arguments for each column we wish to update in the database. When adding arguments, the argument Description should be the column name and the argument Value will be what gets written to the column in the database.

Click on the Add using Column Picker... button. This will open the column picker window which will allow you to interactively pick a column to add. When using this method, the data type is also automatically selected based on the column data type in the database. If the database is available at the time of MP creation/editing, this is the best approach to ensure there are no typos in the column name or the incorrect data

type is chosen. If the database is not available, the column arguments can also be manually added using the Add button.

Note that when manually adding column arguments, if the argument type does not match the database column type, the data will be converted to the database column type if possible.

Step Number 0

Step Type = Put to ODBC Database

Step Title: Put to ODBC Database

Comment:

Add Remove Edit

Add using Column Picker...

Arg#	Type	Description	Method	Value
0	Connection String	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Browse	JOBSTATUS

OK Cancel

Select a column(s)

Database
C:\PARTS.FDB

Table
JOBSTATUS

Columns

JOBID (INTEGER)

JOBNOTES (VARCHAR)

OK Cancel

Select the JOBID column and click OK. Verify that an argument for this column has been added to the MP step argument list.

Step Number 0

Step Type = Put to ODBC Database

Step Title:

Comment:

Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Browse	JOBSTATUS
2	Integer	JOBID	Enter Value	0

Now repeat the above to add the JOBNOTES field as another argument. When done, you should see both columns listed in the MP argument list. Enter some sample values for both the JOBID and JOBNOTES fields. Note that these could also be set to reference other arguments in the MP. Click OK.

Step Number 0

Step Type = Put to ODBC Database

Step Title:

Comment:

Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Browse	JOBSTATUS
2	Integer	JOBID	Enter Value	1024
3	String	JOBNOTES	Enter Value	Passed all Required Tests

Click the Save and Exit button in the MP Editor window.

MP: Database Test

Function Library

- File Operations...
 - New SA File
 - Open SA File
 - Save
 - Save As...
 - Backup Now

Step List

[0] Put to ODBC Database

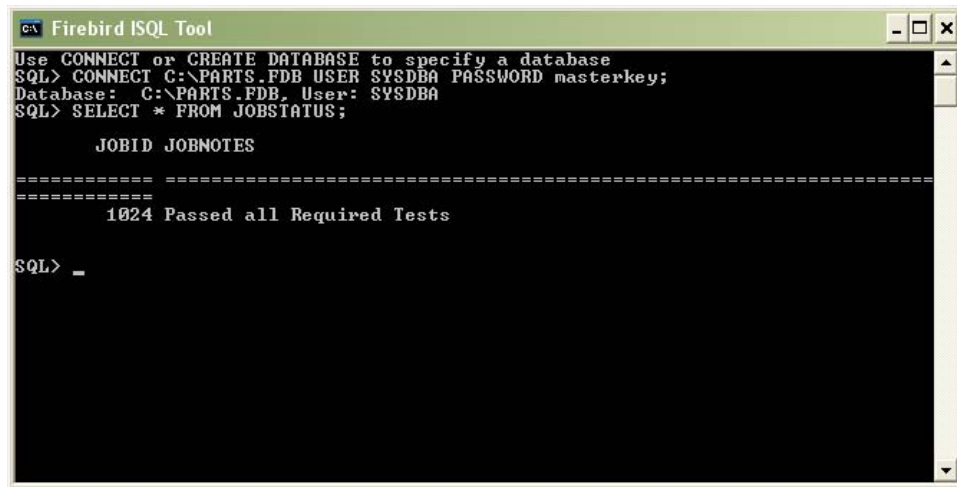
Double-click on the embedded MP which has been created and then click the green circle button to run the MP. Verify there were no errors reported.

Verify Record Insertion

Run the Firebird ISQL Tool, connect to the database, and query for all records in the sample table.

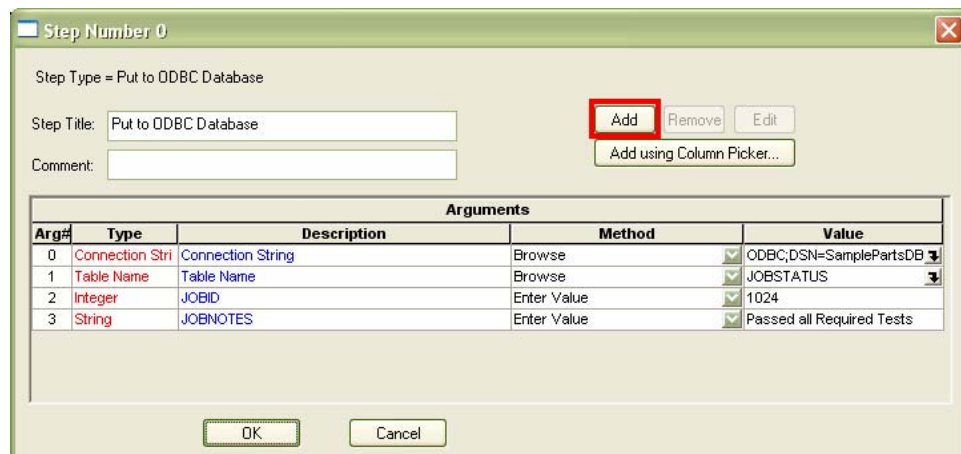
```
CONNECT C:\PARTS.FDB USER SYSDBA PASSWORD masterkey;  
SELECT * FROM JOBSTATUS;
```

You should see the new record added by the MP.

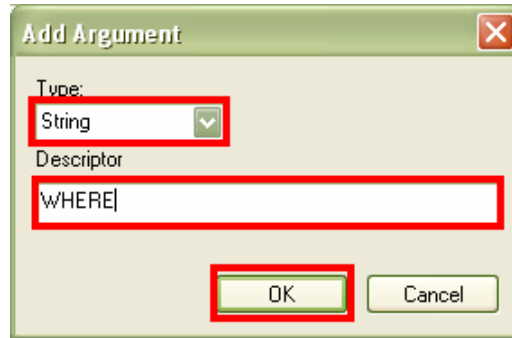


Updating Existing Records

Right-click the embedded MP and select Edit. Double-click the Put to ODBC Database step to open the step properties window. The step is currently configured to insert a new record into the database. In order to update an existing record a WHERE clause must be added to tell the MP step which record to update. To add a WHERE clause, click the Add button.

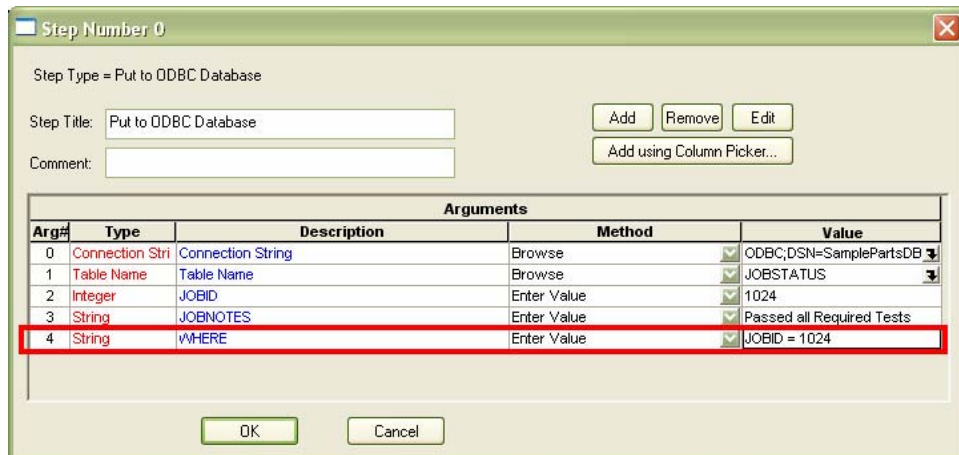


Select string for the type and enter "WHERE" (without the double-quotes) for the descriptor and then click OK.



The 'Add Argument' dialog box has a title bar with a close button. It contains a 'Type' dropdown menu with 'String' selected, a 'Descriptor' text field containing 'WHERE', and 'OK' and 'Cancel' buttons at the bottom.

For the WHERE argument, enter a value which will specify which record to update. The example shown updates the record we previously inserted.

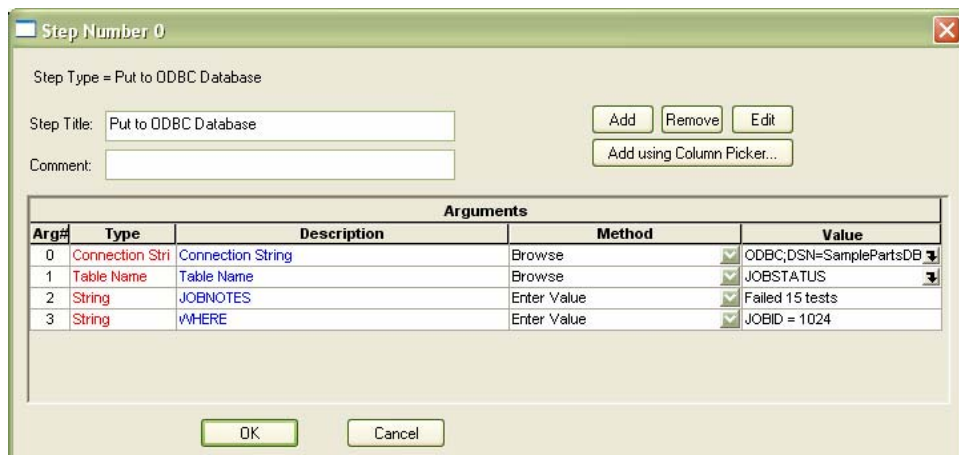


The 'Step Number 0' dialog box shows a step configuration for 'Put to ODBC Database'. It includes fields for 'Step Title' and 'Comment', and buttons for 'Add', 'Remove', 'Edit', and 'Add using Column Picker...'. Below is a table of arguments:

Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Browse	JOBSTATUS
2	Integer	JOBID	Enter Value	1024
3	String	JOBNOTES	Enter Value	Passed all Required Tests
4	String	WHERE	Enter Value	JOBID = 1024

The row for Arg# 4 is highlighted with a red border. 'OK' and 'Cancel' buttons are at the bottom.

Now, select the JOBID argument and click the Remove button, since we're not updating this column. Also, change the JOBNOTES value to be anything different than what it currently is to demonstrate that we've updated the record. The result should look like the below. Click OK.



The 'Step Number 0' dialog box after modifications. The 'JOBID' argument (Arg# 2) has been removed, and the 'JOBNOTES' value has been changed to 'Failed 15 tests'.

Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Browse	JOBSTATUS
2	String	JOBNOTES	Enter Value	Failed 15 tests
3	String	WHERE	Enter Value	JOBID = 1024

'OK' and 'Cancel' buttons are at the bottom.

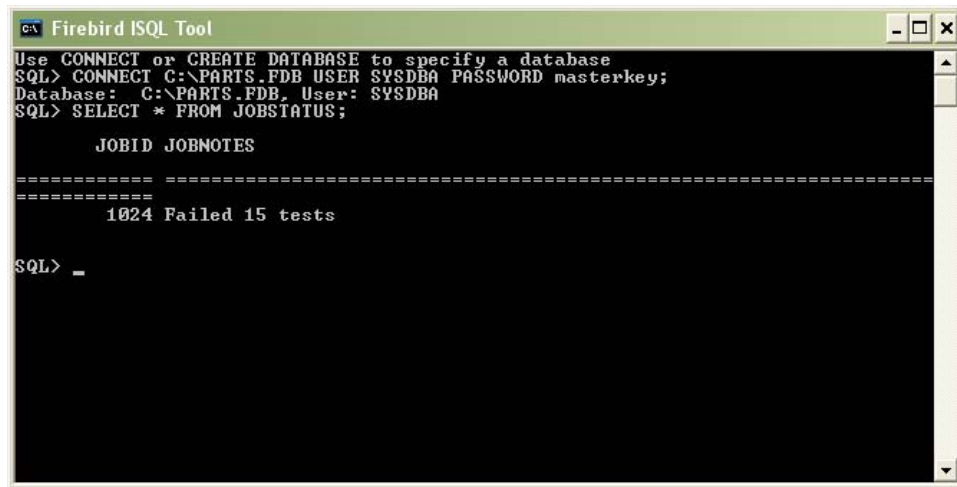
Click Save and Exit and run the MP again. Verify there were no errors reported.

Verify Record Update

Run the Firebird ISQL Tool, connect to the database, and query for all records in the sample table.

```
CONNECT C:\PARTS.FDB USER SYSDBA PASSWORD masterkey;  
SELECT * FROM JOBSTATUS;
```

You should see that the record has been updated by the MP.



Retrieving Data from the Database

Open the MP in the editor and add a new step by double-clicking File Operations...>>Database Operations...>>Get from ODBC Database. Double-click on the new step to open the properties window. Use the same procedure as before to setup the connection string and table name.

When putting data to the database, the WHERE clause is optional and controls whether a new record is inserted or an existing record is updated. When retrieving data from the database, however, the WHERE clause is required. If the WHERE clause results in more than one record being returned from the database, only the first record is considered.

Columns are added as arguments just as before and when run, the values will be set using the data returned from the database. These values can then be referenced in later MP steps. Note that if using the Column picker, the argument types will be automatically determined by the column types in the database. If you manually add column arguments, you can pick the desired type and the value from database will be converted to your chosen type if possible.

Enter a WHERE clause which will retrieve the record created and updated earlier. Use the Column picker to add both fields to the argument list. When done the MP step should look like the below.

Step Number 1

Step Type = Get from ODBC Database

Step Title:

Comment:

Arg#	Type	Description	Method	Value
0	Connection Stri	Connection String	Browse	ODBC;DSN=SamplePartsDB
1	Table Name	Table Name	Browse	JOBSTATUS
2	String	WHERE	Enter Value	JOBID = 1024
3	Integer	JOBID	Enter Value	0
4	String	JOBNOTES	Enter Value	

In order to verify that the values were loaded from the database, it is necessary to add some diagnostic output steps to the MP.

Double-click on Reporting Options>>Notify User Integer and then double-click on the new step to edit the properties as shown below. The leading text explains the value and the value itself is set to reference the loaded data in the previous MP step.

Step Number 2

Step Type = Notify User Integer

Step Title:

Comment:

Arg#	Type	Description	Method	Value
0	String	Leading Text	Enter Value	The JOBID is
1	Integer	Reported Value	Reference	Ref (S1 A3)

Also add a string notification step setup as shown below to display the JOBNOTES text field.

Step Number 3

Step Type = Notify User String

Step Title:

Comment:

Arguments				
Arg#	Type	Description	Method	Value
0	String	Notification Text	Reference	Ref (S1 A4)

OK Cancel

The resultant MP step list should look like the below.

Save Save and Exit Set Password... Detailed View

Exit Remove Password

Step List

- [0] Put to ODBC Database
- [1] Get from ODBC Database
- [2] Notify User Integer
- [3] Notify User String

ations...
is...
ons...

Template
date Templated Report
Graphical View Options

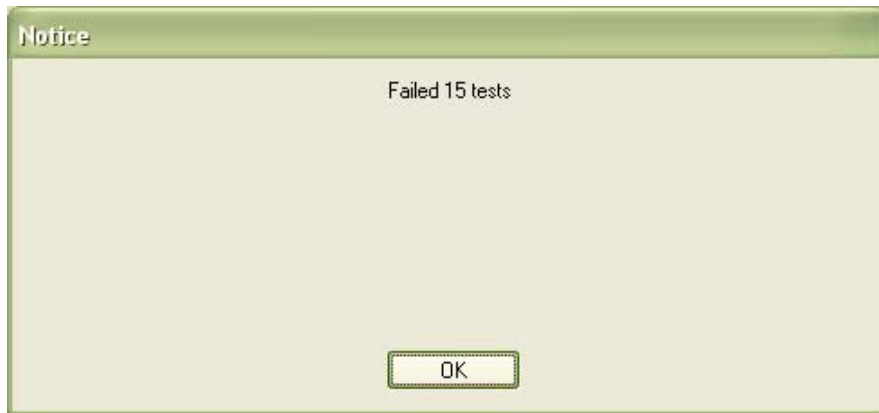
Verify Record Retrieval

Click the Save and Exit button and then run the MP. The MP will update the record, read the values back from the record and then display 2 notifications which should contain the values from the database as shown below.

Notice

The JOBID is 1024

OK



Database Install and Setup

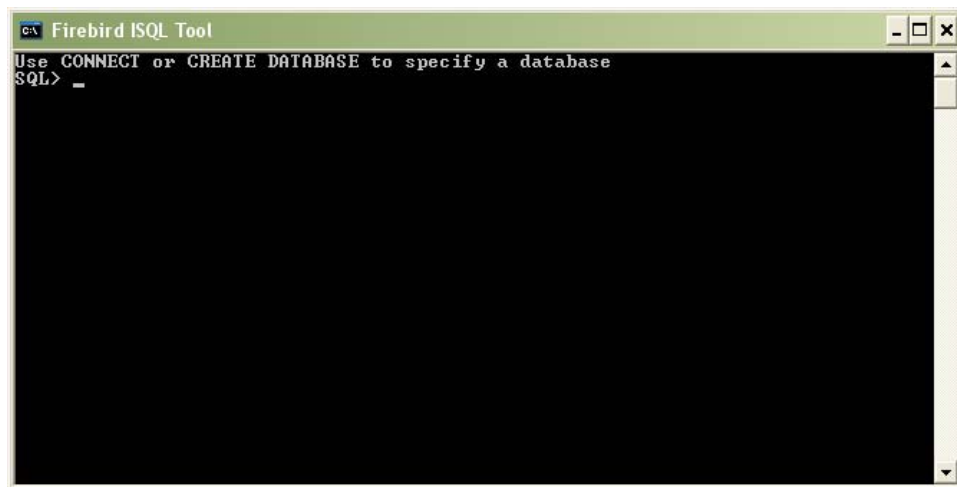
If you already have an installed ODBC-capable database, you can skip this section, however, the examples were created using the sample database created in this section.

Download and Installation

Download and install Firebird 2.0 (SuperServer) from the above link, accepting the default options. This should install the database server as a Windows service and start it running. Next, download and install the ODBC drivers for Firebird from the same web site.

Test the Connection and Create the Sample DB

Select Start Menu>>All Programs>>Firebird 2.0>>Firebird ISQL Tool. This should open the interactive SQL prompt in a DOS window. From this window you can enter database commands (terminated by a semicolon) to be issued to the database server.



Create a sample database.

```
CREATE DATABASE 'C:\PARTS.FDB' PAGE_SIZE 8192 USER  
'SYSDBA' PASSWORD 'masterkey';
```

Create a table in the new database

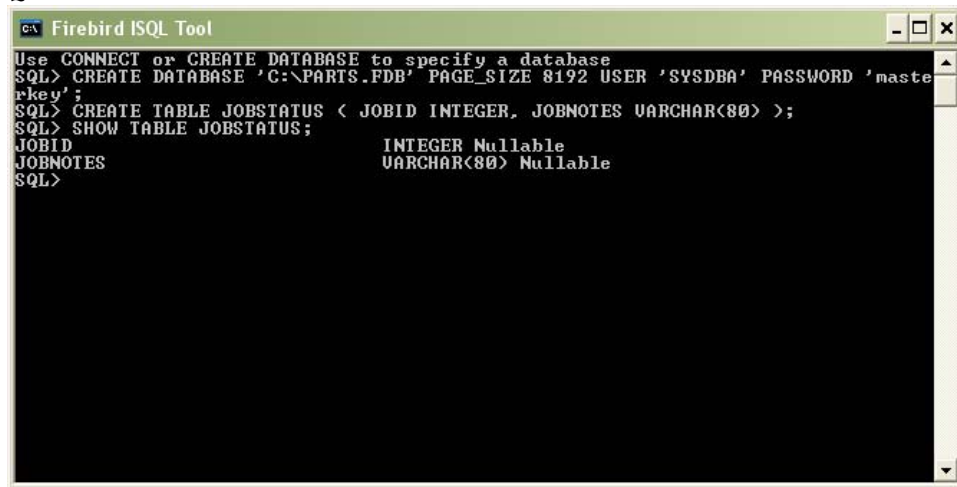
```
CREATE TABLE JOBSTATUS ( JOBID INTEGER, JOBNOTES  
VARCHAR(80) );
```

Verify that the table was created

```
SHOW TABLE JOBSTATUS;
```

Exit the ISQL session.

```
QUIT;
```

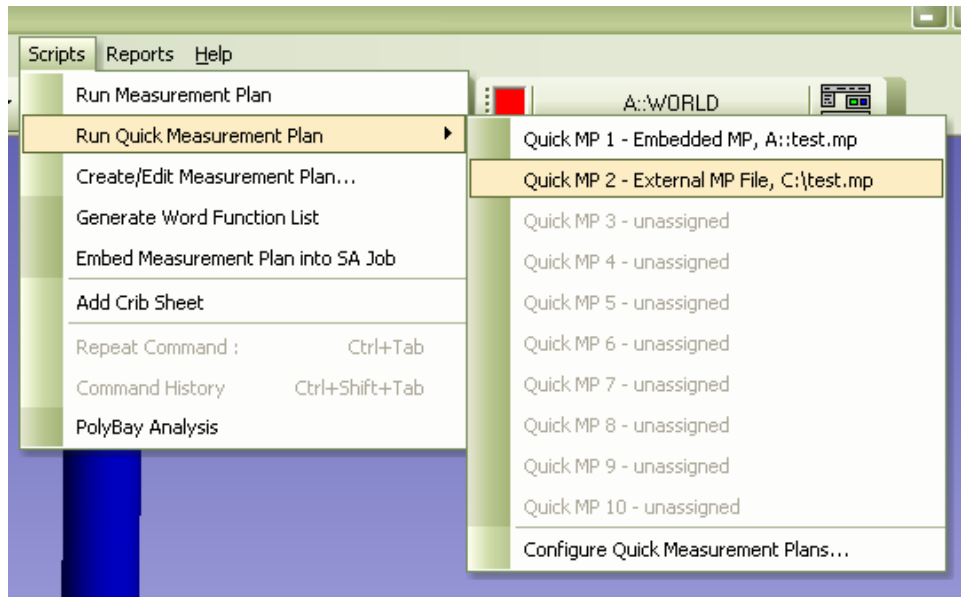


Quick Measurement Plans (QuickMPs)

This feature provides 10 Quick MP slots per job (.xit) file. Each Quick MP slot can be assigned to an embedded MP or to an external MP file. Once assigned, the associated MP can be run by clicking on its entry in the menu. Using the recently added User Interface Customization feature, the Quick MPs can also be assigned to toolbar buttons or toolbar drop-down menus for easier access.

Accessing Quick MPs

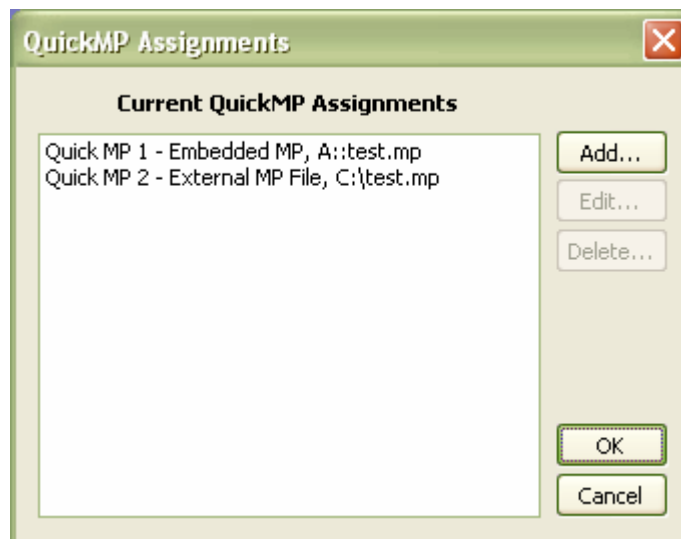
The Quick MP feature is accessed using the "Scripts>>Run Quick Measurement Plan" menu shown below.



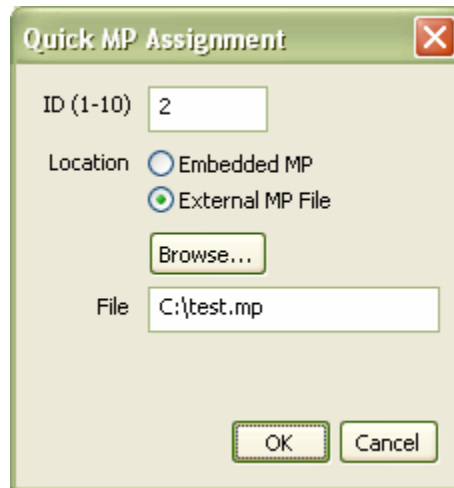
The above menu shows the 10 Quick MP slots for the currently open job file. Assigned slots indicate whether the MP is embedded or external and also provide the specific MP information. Slots which have not been assigned are indicated as such.

Configuring Quick MP Assignments

To configure Quick MP Assignments, select “Scripts>>Run Quick Measurement Plan>>Configure Quick Measurement Plans...”. This will open the “QuickMP Assignments” window shown below.



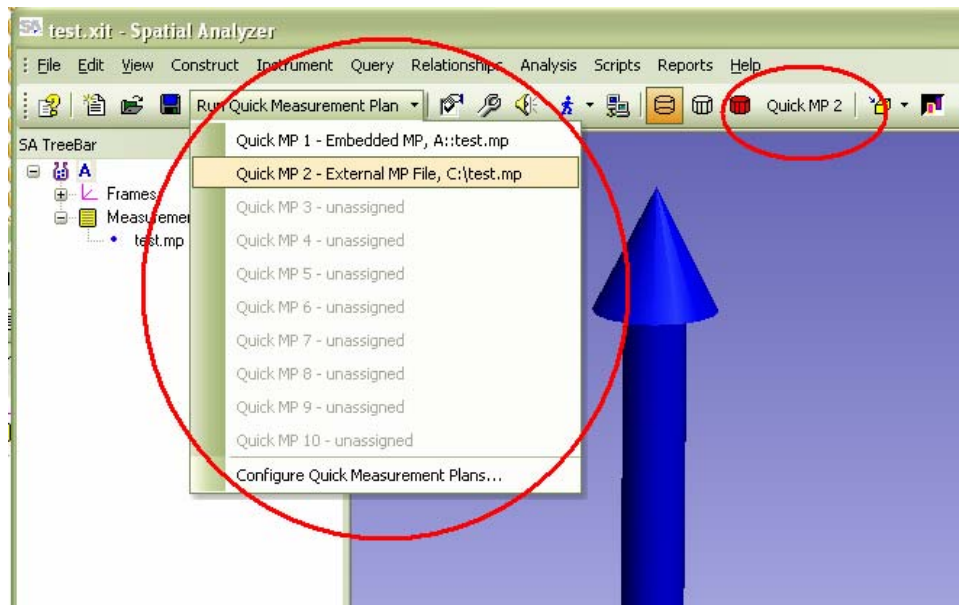
The above window allows adding new Quick MP assignments, editing existing Quick MP assignments, and removing Quick MP assignments which are no longer needed. When adding or editing Quick MP assignments, the “Quick MP Assignment” window is shown.



Using the above window, the Quick MP assignment can be configured. A numeric ID (1-10) can be set for the Quick MP, and the user can decide whether the Quick MP assignment refers to an embedded MP or to an external MP file. Finally, the user can provide the collection and name pair (for embedded MPs) or the filename (for external MP files).

Utilizing User Interface Customization

Using the recently added User Interface Customization feature (see View>>User Interface Customization, then click on the "?" button for more info), Quick MPs can be put directly on the toolbar for easier access. A toolbar drop-down menu can also be created to hold the Quick MPs, making it easy to pick from the list of MPs and execute them. Some examples of the possible customizations are shown below.



MP Function Listing

File Operations...

New SA File

Function	
Description	Opens a new SA file from the default SA template file.
Return	
Description	Success when a new SA file is opened.
Remarks	The default SA job is loaded from Analyzer Data\Templates\default.xit file.

This Function has no arguments.

Open SA File

Function	Opens a new SA file from the default SA template file. The default SA job is loaded from Analyzer Data\Templates\default.xit file.
Description	
Return	
Description	Success if file is found and loaded. Failure if the file can not be found.
Remarks	Path to the SA file to open. Supports both absolute path (e.g., c:\Analyzer Data\test1.xit) and relative path (e.g., .\test1.xit) names.

Arg	Type	Description
0	File Path	SA File Name

Save

Function	
Description	Saves the current file with its current file and path.
Return	
Description	Success is returned after saving the file.
Remarks	The File Save As dialog is opened if the file has not been previously named.

This Function has no arguments.

Save As...

Function	Saves the file with the name and path specified in argument 0. An optional serial number (provided in argument 2) will be added to the file name. If the file is named test.xit and an serial number (e.g., 7) is provided the file is named test7.xit.
Description	

Return Description	Returns success if the file and path are accessible. Check to ensure path exists and the user has adequate privileges if step returns failure.
Remarks	The file name can be an absolute or relative file name and path for the SA file. The ".xit" extension should be in this name. When the Add Serial Number is true the function adds the serial number provided in argument 2 to the file/path name provided in argument 0. A false for the Add Serial Number argument means the serial number argument is ignored. The Optional Number to add is an integer. It is added to the file/path name.

Arg	Type	Description
0	File Path	File Name
1	Boolean	Add Serial Number?
2	Integer	Optional Number

Backup Now

Function Description	Saves a backup copy of the file to the SA backup directory e.g., c:\Analyzer Data\Backup.
Return Description	Returns true when file is successfully backed up. If failure, please check that the user has adequate privileges to the backup directory.
Remarks	

This Function has no arguments.

Open Template File

Function Description	Opens a Read-Only SA template file from the specified path and file name.
Return Description	Returns success if the template file is found and successfully loaded. Failure when the file can not be found or loaded.
Remarks	The path/file name can be absolute (e.g, C:\Analyzer Data\Templates\Setup1.xit) or relative (.\\Templates\Setup1.xit).

Type	Description
0	File Path
	Template File Name

Save As Read-Only Template

Function	
Description	Saves the current SA job file as a read-only template file.
Return	Returns success if the template path/file exists and the file is successfully saved. Failure when the path/file can not be found or saved. If failure ensure check that path exists and that the user has adequate privileges.
Description	
Remarks	The path/file name can be absolute (e.g, C:\Analyzer Data\Templates\Setup1.xit) or relative (.\\Templates\Setup1.xit).

Arg	Type	Description
0	File Path	Template File Name

Exit Measurement Plan

Function	
Description	Exits the measurement plan.
Return	
Description	Returns success as it exits the measurement plan.
Remarks	Used frequently as a referenced step to exit the script when necessary. This function should not be called from Subroutines.

This Function has no arguments.

Shut Down SA

Function	
Description	Closes SA application.
Return	
Description	Success when a new SA file is opened.
Remarks	A file save dialog is opened if the file has unsaved changes. Note: if Utility Operations >> Set Interaction Mode >> SA Interaction argument is set to Silent or Automatic the file save dialog is not presented to the user.

This Function has no arguments.

Run Another Program

Function	Measurement Plan function that initiates an external application with optional command line arguments.
Description	
Return	
Description	Returns success.

This step will wait for the external application to close if argument 2 is true. When argument 2 is false the measurement plan starts the external application and immediately continues to the next step in the plan. The Program Path is a file name to the external application, e.g., "excel.exe" starts MS Excel without opening a workbook. Specifying a file that has an associated application will launch the associated application and open the file. For example if readme.txt exists in the working directory WordPad or Notepad is opens and loads the readme.txt file.

The Command Line Arguments are used by the external application. For example a Visual Basic macro can be specified as a command line option.

A true for the Wait For Program Completion flag sets SA into to wait state for the external application to close before continuing the measurement plan. When true SA waits until the external application shuts down. If false the measurement plan immediately moves to the next step in the process.

Remarks

Arg	Type	Description
0	File Path	Program Path
1	String	Command Line Arguments (optional)
2	Boolean	Wait For Program Completion

Copy General File

Function

Description Copies a source file to a destination file name.

Return

Description Returns success when the file is copied. Failure if the path/file name for either the source or destination can not be found or accessed. Check that the path and file exist and the user's privileges allow access and read/write control to the source and destination directories.

Remarks

File Name arguments in this step are either absolute or a relative file name and path. If the destination file already exists it is overwritten when the Overwrite flag is true.

Arg	Type	Description
0	File Path	Source File Name
1	File Path	Destination File Name
2	Boolean	Overwrite?

Rename General File

Function

Description Changes the name of a file.

Return Description Returns success when the file is renamed. Failure if the path/file name for either the source or destination can not be found or accessed. Check that the path and file exist and the user's privileges allow access and read/write control to the source and destination directories.

Remarks File Name arguments in this step are either absolute or a relative file name and path. If the destination file already exists it is overwritten when the Overwrite flag is true.

Arg	Type	Description
0	File Path	Source File Name
1	File Path	Destination File Name
2	Boolean	Overwrite?

Delete General File

Function

Description Delete a file.

Return Description Returns success when the file is deleted. Failure if the path/file name for either the source or destination can not be found or accessed. Check that the path and file exist and the user's privileges allow access and read/write control to the source and destination directories.

Remarks Absolute or relative file name and path for the file to be deleted.

Arg	Type	Description
0	File Path	File Name

Verify General File Exists

Function

Description Function checks to see if a file exists.

Return Description Returns success when the file is found and failure if it not found.

Remarks Absolute or relative path and file name.

Arg	Type	Description
0	File Path	File Name

File Import...

Import SA File

Function	Imports an SA file. The supports an optional flag to allow the operator to select specific collections from the source file.
Description	
Return	Returns success when the SA job file is imported. Returns failure if the file is not found or the file format is not recognized. Ensure the path and file exist and that the user's privileges allow access to the directory and file.
Description	
Remarks	Absolute or relative path and file name to the SA file to be imported. When true a dialog showing all of the collections in the file to be imported is presented to the operator. Only selected collections are imported. When false all collections are imported into the current SA job file. Imported collection names are changed to avoid duplicate collection names.

Arg	Type	Description
0	File Path	SA File Name
1	Boolean	Allow Operator Selections

Import ASCII: Predefined Formats

Function	Imports an ASCII file of points, clouds, vectors, or planes into the active collection. Select the format, units and define the resulting object names.
Description	
Return	Returns failure if the file can not be accessed or the format is not recognized. Success is returned when the data is successfully imported.
Description	
Remarks	The File Path is an absolute or relative path and file name to the ASCII file to be imported. File format from a suite of pre-defined formats, e.g., PtName, X, Y, Z or GpName, PtName, X, Y, Z Select the units of the data in the ASCII file. The ratio of the file's units verse SA current units setting is applied to the data as it is imported. Name of the gr oup. Not used if the selected format has a group designation in it. When true points are imported as cloud points. If false they are imported into point groups.

Arg	Type	Description
-----	------	-------------

Arg	Type	Description
0	File Path	ASCII File Path
1	File Format	File Format
2	Units	Units
3	Group Name	Group Name
4	Boolean	Import as Cloud

Import STEP File

Function

Description Imports a STEP file of objects into the active collection.

Return

Description Returns failure if the file is not accessible or the format is not recognized. Success is returned after the objects are imported.

Remarks

The File Path is absolute or relative path and file name to the STEP file to be imported. When the Dialog Entity Filter is true a dialog to select which object types is to be imported is shown to the user. If false all of the objects in the file are imported into the active collection.

Arg	Type	Description
0	File Path	STEP File Path
1	Boolean	Display Entity Filters

Import IGES File

Function

Description Imports a IGES file of objects into the active collection.

Return

Description Returns failure if the file is not accessible or the format is not recognized. Success is returned after the objects are imported.

Remarks

The File Path is absolute or relative path and file name to the IGES file to be imported.

Arg	Type	Description
0	File Path	IGES File Path

Import VDA/FS File

Function	
Description	Imports a VDA/FS file of objects into the active collection.
Return	Returns failure if the file is not accessible or the format is not recognized.
Description	Success is returned after the objects are imported.
Remarks	The File Path is absolute or relative path and file name to the VDA/FS file to be imported.

Arg	Type	Description
0	File Path	VDA/FS File Path

Import CATIA V4 File

Function	
Description	Imports a CATIA V4 file of objects into the active collection.
Return	Returns failure if the file is not accessible or the format is not recognized.
Description	Import of the file returns failure if loading fails due to licensing issues. Success is returned after the objects are imported.
Remarks	The File Path is absolute or relative path and file name to the CATIA V4 .model file to be imported.

Arg	Type	Description
0	File Path	CATIA V4 File Path

Import CATIA V5 File

Function	
Description	Imports a CATIA V5 file of objects into the active collection.
Return	Returns failure if the file is not accessible or the format is not recognized.
Description	Import of the file returns failure if loading fails due to licensing issues. Success is returned after the objects are imported.
Remarks	The File Path is absolute or relative path and file name to the CATIA V5 .CATPart or CATProduct file to be imported.

Arg	Type	Description
-----	------	-------------

Arg	Type	Description
0	File Path	CATIA V5 File Path

Import CATIA CGR File

Function	
Description	Imports a CATIA CGR file of objects into the active collection.
Return	Returns failure if the file is not accessible or the format is not recognized.
Description	Import of the file returns failure if loading fails due to licensing issues. Success is returned after the objects are imported.
Remarks	The File Path is absolute or relative path and file name to the CATIA .CGR file to be imported.

Arg	Type	Description
0	File Path	CATIA .CGR File Path

Import Unigraphics File

Function	
Description	Imports a Unigraphics file of objects into the active collection.
Return	Returns failure if the file is not accessible or the format is not recognized.
Description	Import of the file returns failure if loading fails due to licensing issues. Success is returned after the objects are imported.
Remarks	The File Path is absolute or relative path and file name to the Import Unigraphics .prt file to be imported.

Arg	Type	Description
0	File Path	Import Unigraphics File Path

Import Embedded File

Function	
Description	Imports a file to be embedded in the active collection.
Return	Returns failure if the file is not accessible or the format is not recognized.
Description	Success is returned after the file is imported.
Remarks	The File Path is absolute or relative path and file name to be embedded.

Arg	Type	Description
0	File Path	Embedded File Path

File Export...

Export ASCII Points

Function Description Writes an ASCII file of one or more groups of points. The delimiter and target name format are options.

Return Description Returns success when the ASCII file is written. The step returns failure if the file can not be written. Ensure the path and file exist and that the user's privileges allow access to the directory and file.

Remarks The File Path argument is an absolute or relative path and file name to the ASCII file for the points to be saved. Create a list of one or more point group names in the Group Name Ref List argument. A list of the available delimiters is provided in the Export Delimiter Space argument. Configure the target name format with the Target Name Format argument. Select the target name format e.g., Collection::Group::Target, Group::Target, Target, or None.

Arg	Type	Description
0	File Path	ASCII File Path
1	Group Name Ref List	Group Names to export
2	Export Delimiter Spec	Data Delimiter
3	Target Name Format	Target Name Format

Export Vector Container to Excel File

Function Description Creates an excel file of vectors from a Vector Group. The file contains vector group summary statistics and a row of data for each vector.

Return Description Returns success when the Excel file of vectors is written. This step returns failure if the file can not be written. Ensure the path and file exist and that the user's privileges allow access to the directory and file.

Remarks The File Path is absolute or relative path and file name to the Excel file for the vectors to be saved. The name of the vector group from within the active collection. When Overwrite existing file is true and a file with the same name already exist, it is deleted and the new file created. If the does not exist and the path and file name are valid the file is written. If false the new file is not written if a file with the same name exists.

Arg	Type	Description
0	File Path	Excel File Path
1	Vector Group Name	Vector group to export
2	Boolean	Overwrite existing file?

Export Vector Container to ASCII File

Function Description Creates an ASCII file of vectors from a Vector Group. The file contains vector group data for each vector.

Return Description Returns success when the ASCII file of vectors is written. This step returns failure if the file can not be written. Ensure the path and file exist and that the user's privileges allow access to the directory and file.

Remarks The File Path is absolute or relative path and file name to the ASCII file for the vectors to be saved. The name of the vector group from within the active collection. When Overwrite existing file is true and a file with the same name already exist, it is deleted and the new file created. If the does not exist and the path and file name are valid the file is written. If false the new file is not written if a file with the same name exists.

Arg	Type	Description
0	File Path	ASCII File Path
1	Vector Group Name	Vector group to export
2	Boolean	Overwrite existing file?

Export STEP File - Entire Model

Function Description Exports a STEP file of all of the objects in the SA job file, including those in inactive collections.

Return Description Return success if the exporting file name and path are valid.

Remarks The file path is an absolute or relative path and file name to the STEP file.

Arg	Type	Description
0	File Path	STEP File Path

Export STEP File - Partial Model

Function	
Description	Exports a STEP file of named objects.
Return	
Description	Return success if the exporting file name and path are valid. A partial success is return if all of the named objects can not be found to export. Failure is reported when the path and file name are not valid.
Remarks	The file path is an absolute or relative path and file name to the STEP file. List the objects to be exported in the Object Name Reference List.

Arg	Type	Description
0	File Path	STEP File Path
1	Object Name Ref List	Object Name List

Export IGES File - Entire Model

Function	
Description	Exports a IGES file of all of the objects in the SA job file, including those in inactive collections.
Return	
Description	Return success if the exporting file name and path are valid.
Remarks	The file path is an absolute or relative path and file name to the IGES file.

Arg	Type	Description
0	File Path	IGES File Path

Export IGES File - Partial Model

Function	
Description	Exports an IGES file of named objects.
Return	
Description	Return success if the exporting file name and path are valid. A partial success is return if all of the named objects can not be found to export. Failure is reported when the path and file name are not valid.
Remarks	The file path is an absolute or relative path and file name to the IGES file. List the objects to be exported in the Object Name Reference List.

Arg	Type	Description
-----	------	-------------

Arg	Type	Description
0	File Path	IGES File Path
1	Object Name Ref List	Object Name List

Export VDA/FS File - Entire Model

Function Description Exports a VDA/FS file of all of the objects in the SA job file, including those in inactive collections.

Return Description Return success if the exporting file name and path are valid.

Remarks The file path is an absolute or relative path and file name to the VDA/FS file.

Arg	Type	Description
0	File Path	VDA/FS File Path

Export VDA/FS File - Partial Model

Function Description Exports a VDA/FS file of named objects.

Return Description Return success if the exporting file name and path are valid. A partial success is return if all of the named objects can not be found to export. Failure is reported when the path and file name are not valid.

Remarks The file path is an absolute or relative path and file name to the VDA/FS file. List the objects to be exported in the Object Name Reference List.

Arg	Type	Description
0	File Path	VDA/FS File Path
1	Object Name Ref List	Object Name List

Datashare Operations...

Load Datashare File

Function Description Loads a DataShare file.

Return Description	<p>Returns success when the DataShare file is imported. Returns failure if the file is not found or the file format is not recognized. Ensure the path and file exist and that the user's privileges allow access to the directory and file.</p> <p>Browse to an existing DataShare file to load arguments to each entry (e.g., variable) in the file. The data from the file is available as references for subsequent steps. DataShare files are data type (e.g., Boolean, integer, string, double, etc) specific formatted files used to share data between measurement plan functions or external applications. The files are either binary or ASCII. Before importing a DataShare a copy of the file is created with the File Operations >> File Export >> Save DataShare File function. Load the argument list of the Load DataShare File function with the browse method in the File Path argument.</p> <p>Use the browse method to get a fresh argument list of the data in the file into this step's list of arguments. A DataShare File must exist before the argument list can be populated. Absolute or relative path and file name to the DataShare file to be imported.</p>
Remarks	

Arg	Type	Description
0	File Path	DataShare File Path

Save DataShare File

Function Description	<p>Saves a DataShare file from a list of arguments that is assembled by adding them to the step. DataShare files are used to share data between measurement plans and external applications.</p>
Return Description	<p>Returns success when the DataShare file is written. Returns failure if the file can not be written. Ensure the path and file exist and that the user's privileges allow access to the directory and file.</p> <p>Construct the data share file format by adding arguments (e.g., variables) with descriptions to the list. Each argument is declared from a list of data types (e.g., string, integer, double, Boolean, vector, transform). Once the argument is declared its value is assigned by value or reference. The data type, description and values are written to the file in the sequence defined by the argument list. Data share files are ASCII or binary format. Binary files are suggested if the file size is large or data should not be easily viewed or edited. ASCII file formats are viewable in a text editor.</p> <p>The File Path is absolute or relative path and file name to the DataShare file to be saved. When Binary Format is true the file will be saved in a binary format and if false the file is saved in an ASCII format. When Append to Existing File is false the file is saved with only the defined arguments. If true the file is opened and the defined arguments are appended to the bottom of the file. This is good option when using a DataShare file as a log file mechanism.</p>
Remarks	

	Type	Description
	0 File Path	DataShare File Path
	1 Boolean	Save in Binary Format?
	2 Boolean	Append to existing file?

Get Integer from Datashare file

Function Description	This function retrieves an integer datashare value by name. Specify the datashare source with the DataShare file path and then the name of the datashare variable. The function returns the Double value in the Double Results Only argument.
Return Description	Returns success when the DataShare file variable is found. Returns failure if the file or variable can not be found. Ensure the path and file exist and that the user's privileges allow access to the directory and file.
Remarks	Specify a datashare file and then retrieve a variable by naming it in the Integer name argument. This function works with embedded datashare files.

	Type	Description
	0 File Path	DataShare File Path
	1 String	Double Name
	2 Double	Double Value

Get Double from Datashare file

Function Description	This function retrieves a double datashare value by name. Specify the datashare source with the DataShare file path and then the name of the datashare variable. The function returns the integer value in the integer Results Only argument.
Return Description	Returns success when the DataShare file variable is found. Returns failure if the file or variable can not be found. Ensure the path and file exist and that the user's privileges allow access to the directory and file.
Remarks	Specify a datashare file and then retrieve a variable by naming it in the Double name argument. This function works with embedded datashare files.

	Type	Description
	0 File Path	DataShare File Path
	1 String	Integer Name
	2 Integer	Integer Value

Get String from Datashare file

Function Description	This function retrieves an String datashare value by name. Specify the datashare source with the Datashare file path and then the name of the datashare variable. The function returns the String value in the String Results Only argument.
Return Description	Returns success when the DataShare file variable is found. Returns failure if the file or variable can not be found. Ensure the path and file exist and that the user's privileges allow access to the directory and file.
Remarks	Specify a datashare file and then retrieve a variable by naming it in the String name argument. This function works with embedded datashare files.

	Type	Description
	0 File Path	DataShare File Path
	1 String	String Name
	2 String	String Value

Get Vector from Datashare file

Function Description	This function retrieves a Vector datashare value by name. Specify the datashare source with the Datashare file path and then the name of the datashare variable. The function returns the Vector value in the Vector Results Only argument.
Return Description	Returns success when the DataShare file variable is found. Returns failure if the file or variable can not be found. Ensure the path and file exist and that the user's privileges allow access to the directory and file.
Remarks	Specify a datashare file and then retrieve a variable by naming it in the Vector name argument. This function works with embedded datashare files.

	Type	Description
	0 File Path	DataShare File Path
	1 String	Vector Name
	2 Vector	Vector Value

Get Transform from Datashare file

Function Description	This function retrieves a Transform entity datashare value by name. Specify the datashare source with the DataShare file path and then the name of the datashare variable. The function returns the Transform value in the Transform Results Only argument.
Return Description	Returns success when the DataShare file variable is found. Returns failure if the file or variable can not be found. Ensure the path and file exist and that the user's privileges allow access to the directory and file.
Remarks	Specify a datashare file and then retrieve a variable by naming it in the Transform name argument. This function works with embedded datashare files.

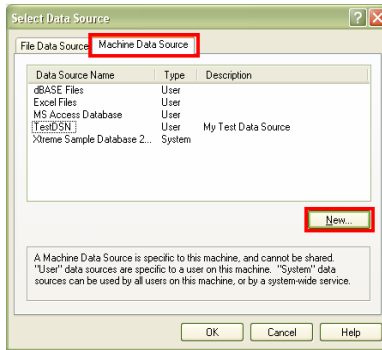
	Type	Description
	0 File Path	DataShare File Path
	1 String	Transform Name
	2 Transform	Transform Value

Database Operations...

MP Database (ODBC) functions allow putting data to and getting data from any ODBC-capable database. They allow the MP programmer to specify an ODBC connection string (which identifies the database driver, the specific database, and any login info), table name, and a set of columns with values to be put to the database or a set of columns to be retrieved from the database. An SQL WHERE clause can also be used which allows updating existing records in the database and which is used when retrieving data from the database.

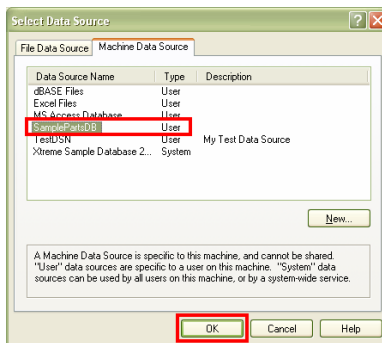
Put to ODBC Database

Function Description	For the connection string argument, change the Method to Browse and then click the hotspot button which should appear.
----------------------	--



New button.

The ODBC Setup window will now be displayed,



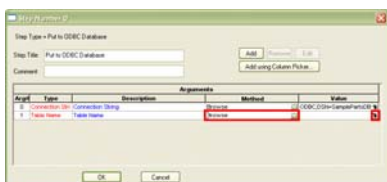
for future MP editing. Select the SamplePartsDB data source and click OK.

Verify that the connection string argument has been set.

The complete connection string is shown below. As is evident from the below, it is usually much easier to browse for the string than to manually enter the string.

DSN=SamplePartsDB;Driver=Firebird/InterBase(r)

driver;Dbname=C:\PARTS.FDB;CHARSET=NONE;PWD=masterkey;UID=SYSDBA



Select the sample table created earlier and then click OK. You can ignore the RDB\$ tables. These are system tables created by Firebird.



Verify that the Table Name argument has been set.



Return Description Returns success when the data is sent to the database. Returns failure when the database, table, or records are not found. Ensure the connection string, table name, and field names are correct.

At this point, we have enough information in this step to connect to the database and direct our attention to a particular table in that database. It is now necessary to add arguments for each column we wish to update in the database. When adding arguments, the argument Description should be the column name and the argument Value will be what gets written to the column in the database.



Click on the Add using Column Picker... button. This will open the column picker window which will allow you to interactively pick a column to add. When using this method, the data type is also automatically selected based on the column data type in the database. If the database is available at the time of MP creation/editing, this is the best approach to ensure there are no typos in the column name or the incorrect data type is chosen. If the database is not available, the column arguments can also be manually added using the Add button.



Note that when manually adding column arguments, if the argument type does not match the database column type, the data will be converted to the database column type if possible.

Select the JOBID column and click OK. Verify that an argument for this column has been added to the MP step argument list.



Now repeat the above to add the JOBNOTES field as another argument. When done, you should see both columns listed in the MP argument list. Enter some sample values for both the JOBID and JOBNOTES fields. Note that these could also be set to reference other arguments in the MP. Click OK.

Remarks

Get from ODBC Database

Open the MP in the editor and add a new step by double-clicking File Operations...>>Database Operations...>>Get from ODBC Database. Double-click on the new step to open the properties window. Use the same procedure as before to setup the connection string and table name.

When putting data to the database, the WHERE clause is optional and controls whether a new record is inserted or an existing record is updated. When retrieving data from the database, however, the WHERE clause is required. If the WHERE clause results in more than one record being returned from the database, only the first record is considered.

Function Description

Columns are added as arguments just as before and when run, the values will be set using the data returned from the database. These values can then be referenced in later MP steps. Note that if using the Column picker, the argument types will be automatically determined by the column types in the database. If you manually add column arguments, you can pick the desired type and the value from database will be converted to your chosen type if possible.

Enter a WHERE clause which will retrieve the record created and updated earlier. Use the Column picker to add both fields to the argument list. When done the MP step should look like the figure.



In order to verify that the values were loaded from the database, it is necessary to add some diagnostic output steps to the MP.

Return Description	Returns success when the data is retrieved from the database. Returns failure when the database, table, or records are not found. Ensure the connection string, table name, and field names are correct.
Remarks	Test and verify the results with any of the other MP mechanisms.

MP Subroutines...

Run Subroutine

Function Description	Runs an MP subroutine.
Return Description	Can return success, partial success or failure depending on how the subroutine defines its return condition.
Remarks	<p>A list of the subroutines arguments are added to this step when the measurement plan subroutine is browsed to or linked to the embedded subroutine. Please note the argument list is refreshed only when the subroutine has been browsed to or linked to an embedded subroutine. Subroutine arguments are either inputs or return values. Each one is type specific (e.g., string, integer, double, vector, Boolean, or transform). Input arguments are those added to the subroutines first line... Define Subroutine Input Values. Return arguments are added to the last line in the subroutine... Define Subroutine Return values.</p> <p>Input the path to the subroutine or link it to an embedded MP subroutine. This path may be an absolute or relative path or possible linked too an embedded subroutine. Embedded measurement plans are those that have been added to a specific SA's job database. Once embedded they are saved with the job file.</p>

Arg	Type	Description
	File Path or Embedded MP Subroutine 0 File	File Path

Define Subroutine Input Values

Function	
Description	The measurement plan step that must be at the beginning of MP subroutines.
Return	
Description	No return value
Remarks	A list of subroutine arguments are added to the step. They are type specific variable entries. The types include String, Integer, Double, Boolean, Vector, and Transform. Descriptions can accompany each argument added to the list. Specific default values in the value field. The calling MP will load each of the arguments at run time.

This Function has no arguments.

Return from Subroutine Now

Function	This function is used to return from within the subroutine (i.e., before the end of the subroutine). Its return argument list types must match the last line in e subroutine.
Description	
Return	
Description	Return value is defined by the setting in the argument.
Remarks	The last line of subroutines is always the Define Subroutine Return Values. The return condition (e.g., Success, Partial Success, or Failure) and the returned values can differ from the Define Subroutine Return Values step. While the values can differ, return arguments their sequence and type must match the Define Subroutine Return Values step. The return condition for the subroutine, e.g., Success, Partial Success, or Failure. The Run Subroutine's return condition is set by this argument.

Arg	Type	Description
	Measurement MP Subroutine Return Step 0 Plan Result	Result

Define Subroutine Return Values

Function	All measurement plan subroutines must use this step as its last line. The subroutine's return condition and return argument list are defined by this step.
Description	

Return	
Description	Return value is defined by the setting in the argument.
	Add return arguments to the list when needed. Each return argument's type (string, integer, Boolean, double, vector, or transform) must be defined. These arguments will be available in the calling MP's Run Subroutine step after the subroutine is saved and either browsed too or linked too as an embedded Measurement Plan.
Remarks	The return condition for the subroutine, e.g., Success, Partial Success, or Failure. The Run Subroutine's return condition is set by this argument.

Arg	Type	Description
0	Measurement Plan Result	MeasurementMP Subroutine Return Step Result

View Control...

Auto-Scale

Function	Auto-scales SA's graphics view to fit all of the visible objects into the graphics view port.
Description	
Return	
Description	Returns success.
Remarks	A simple function that sets up the graphics window to show all of the visible objects.

This Function has no arguments.

Show Labels

Function	Sets SA's show labels toggle. The argument enables the toggle to be set to true or false. The labels function shows labels to all visible points, instruments, and frame axes.
Description	
Return	
Description	Returns success.
Remarks	When the Labels On flag is set to true ... point, instrument and frame axes label are shown. If false the label are turned off.

Type	Description
------	-------------

Type	Description
0 Boolean	Labels On?

Set Render Mode

Function	
Description	Sets the SA shading mode to Solid or Wireframe.
Return	
Description	Returns success.
Remarks	When the Shading On flag is true SA renders objects with shading on. If false the rendering mode is set to wireframe.

Arg	Type	Description
0 Boolean		Shading On?

Set point of view

Function	Sets SA graphics viewport to named view (e.g., Side, Front, Top, Oblique). Any of the views in the SA file can be called by named. If the SA job file contains a V1 view. Call it by name. Note the view's Restore Zoom Settings is set in the MP call and the number of intermediate viewport states is controlled by the User Options >> Display Tab >> View Animation Steps parameter.
Description	
Return	Returns true when the named view is found. Failure is return when the view is found in the list of views in the SA job.
Description	
Remarks	Name of view in the current SA job, e.g., Side, Top, Front, Oblique. Configure other views in a template and call them by name.

Arg	Type	Description
0 View Name		View Name

Hide Objects

Function	
Description	Sets the show flag to 'off' for a list of objects.
Return	
Description	Returns partial success when all of the listed objects are not found. Success is returned when all the object names are found.

Remarks Create a list of object either by name or reference with the enter method. Reference other Object Name Reference Lists to use lists of objects that exist in other MP steps.

Arg	Type	Description
0	Object Name Ref List	Objects To Hide

Hide Object

Function
Description Set an object's show property to off.

Return
Description Return success when the object name is found. Failure is returned when the object name can not be found.

Remarks Name of an object in the active collection.

Arg	Type	Description
0	Object Name	Object To Hide

Show Objects

Function
Description Sets the show property to true for a list objects. Enter a list of object names or reference an existing object name reference list.

Return
Description Returns partial success when all of the object names are not found. Success is returned when all of the object names are found.

Remarks Create a list of objects either by name or reference with the enter method. Reference Object Name Reference Lists to use lists of objects that exist in other MP steps.

Arg	Type	Description
0	Object Name Ref List	Objects To Show

Show Object

Function
Description Set an object's show property to on.

Return Description Return success when the object name is found. Failure is returned when the object name can not be found.

Remarks Name of an object in the active collection.

Arg	Type	Description
0	Object Name	Object To Show

Show/Hide Instruments

Function Description Sets a list of instrument show properties to either true or false.

Return Description Returns success when all the instrument id's are found.

Remarks List of instrument id's to set their show property. When the Show Instrument flag is true the instrument's is shown. When false the instruments in the list are hidden.

Arg	Type	Description
0	Instrument ID List	Instrument IDs
1	Boolean	Show Instruments?

Set SA's Window State

Function Description Sets SA window state to Maximize, Minimize, Restore, Show or Hide.

Return Description Returns success.

Remarks Select from the available options e.g., Maximize, Minimize, Restore, Show or Hide.

Arg	Type	Description
0	Window State	Window State

Construction Operations...

Copy Object

Function

Description Copies a named collection::object to a new object name.

Return

Description Returns success if the source object is found and destination object is created. Failure is returned if the source object is not found or when the destination object exists and the overwrite flag is set to false.

An overwrite option enables it to avoid overwriting an existing object. The overwrite option can also be used to ensure that the copied object overwrites an existing one.

Remarks

Specify the names of the source object and new destination object name e.g., Point Group Name, Plane Name, Sphere Name with Object Name arguments. When the Overwrite flag is true an existing object with the same type and name as the destination name is overwritten. If false and the destination object exists it is not copied.

Arg	Type	Description
0	Collection Object Name	Source Object
1	Collection Object Name	New Object Name
2	Boolean	Overwrite if exists?

Copy Objects to a collection

Function

Description Copies a list of objects from the active collection to a target/destination collection.

Return

Description Returns success when the source objects are found in the active collection and copied to the target collection. Failure is returned if the source objects are not found.

A new collection is created if the target/destination collection name is not found in the SA job. When the target object(s) already exist the names of objects being copied are automatically sequenced to avoid duplicate names.

Remarks

A list of object names (e.g., Point Group Names, Plane names, Circle Names etc.) in the active collection is specified in the Collection Object Name Ref List argument. Name of the destination collection where the objects are copied too.

Arg	Type	Description
-----	------	-------------

Arg	Type	Description
0	Collection Object Name Ref List	Source Objects
1	Collection Name	Target Collection Name

Move Objects to a collection

Function Description	Moves a list of objects from the active collection to a target/destination collection.
Return Description	Returns success when the source objects are found in the active collection and moved to the target collection. Failure is returned if the source objects are not found.
Remarks	<p>A new collection is created if the target/destination collection name is not found in the SA job. When the target object(s) already exist the names of objects being moved are automatically sequenced to avoid duplicate names.</p> <p>A list of object names (e.g., Point Group Names, Plane names, Circle Names etc.) in the active collection is specified in the Collection Object Name Ref List argument. Name of the destination collection where the objects are moved too.</p>

Arg	Type	Description
0	Collection Object Name Ref List	Source Objects
1	Collection Name	Target Collection Name

Copy Objects - Point to Point Delta

Function Description	Copies and then translates a list of objects in the active collection. The copies are moved (e.g., translated) from their original position by the vector defined between the points.
Return Description	Returns success when the source objects are found in the active collection and moved along the vector defined by the two points. Failure is returned if the source objects or the points are not found.
Remarks	<p>The copied object names are automatically sequenced to avoid duplicate names. Note: the group(s) with the first and second points can also be copied and moved along the point to point delta in this process.</p> <p>A list of object names (e.g., Point Group Names, Plane names, Circle Names etc.) in the active collection is specified by the Object Name Ref List. The first and second points are (i.e., origin to end) point name for translation vector. The point name are constructs of Collection::Group::Target. The active collection name is assumed if the Collection name is omitted.</p>

Arg	Type	Description
0	Object Name Ref List	Objects to Copy
1	Point Name	First Delta Point
2	Point Name	Second Delta Point

Move Objects - Point to Point Delta

Function Description	Moves (i.e., translates) a list of objects in the active collection. The objects are moved (e.g., translated) from their original position by the vector defined between the points.
Return Description	Returns success when the objects are found in the active collection and moved along the vector defined by the two points. Failure is returned if the source objects or the points are not found.
	Note: the group(s) with the first and second points can also be moved in this process.
Remarks	A list of object names (e.g., Point Group Names, Plane names, Circle Names etc.) in the active collection is specified by the Object Name Ref List. The first and second points are (i.e., origin to end) point name for translation vector. The point name are constructs of Collection::Group::Target. The active collection name is assumed if the Collection name is omitted.

Arg	Type	Description
0	Object Name Ref List	Objects to Move
1	Point Name	First Delta Point
2	Point Name	Second Delta Point

Rename Point

Function Description	Renames a point.
Return Description	Returns success when the original point is found. Failure is returned if the original point name is not found or when the new point name exist and the overwrite flag is set to false.
	An overwrite option enables the original point to overwrite another point when a point of the same name exists. When the overwrite option is set to false a point with the matching name is not overwritten.
Remarks	Name of the original (e.g., source) point being renamed is specified and then

the destination point name. When the Overwrite flag is true the original point overwrites an existing point with the new name.

Arg	Type	Description
0	Point Name	Original Point Name
1	Point Name	New Point Name
2	Boolean	Overwrite if exists?

Rename Collection

Function

Description Renames a collection.

Return

Description Returns success when the original collection is found. Failure is returned if the original name is not found or when the new collection name exists.

Remarks

Name of the original (e.g., source) collection being renamed is specified and then the destination object name.

Arg	Type	Description
0	Collection Name	Original Collection Name
1	Collection Name	New Collection Name

Rename Object

Function

Description Renames an object.

Return

Description Returns success when the original object is found. Failure is returned if the original name is not found or when the new object name exist and the overwrite flag is set to false.

An overwrite option enables the original object to overwrite another object when an object of the same type and new name exists. When the overwrite option is set to false an object with the matching name is not overwritten.

Remarks

Name of the original (e.g., source) object being renamed is specified and then the destination object name. When the Overwrite flag is true the original object overwrites an existing object with the new name.

Arg	Type	Description
0	Object Name	Original Object Name
1	Object Name	New Object Name
2	Boolean	Overwrite if exists?

Delete Object

Function Description	Deletes an object by name. Specify the object name with a collection object name.
Return Description	Returns success when the object name is found. Failure is returned when the object name is not found.
Remarks	Specify the collection object name of the object to delete.

Arg	Type	Description
0	Collection Object Name	Object Name

Delete Points

Function Description	Deletes a list of points.
Return Description	Return success when all of the point names are found and deleted. Failure is returned when one or more point names are not found.
Remarks	The individual points are defined by their Collection::Group::Target names. If the collection name is omitted the active collection name is assumed.

Arg	Type	Description
0	Point Name Ref List	Point Names

Collections...

Set (or construct) default collection

Function Description	Makes the collection the active collection in the SA job.
Return	Returns success when the collection is made active. Failure is returned if the new collection name can not be created. The collection name must follow SA

Description	naming conventions.
Remarks	If the collection name is not found in the SA job it is created and then made the active collection. Name of the collection to make active. The collection name must follow SA naming conventions.

Arg	Type	Description
0	Collection Name	Collection Name

Points and Groups...

Construct a Point in Working Coordinates

Function	
Description	Constructs a point relative to the current working frame.
Return	Returns success when the new point is created. Failure is returned if the point name does not follow SA's naming conventions or the point vector is not properly defined.
Description	The point name and the point's X, Y, and Z components in vector form are inputs. If a point with same name already exists the new point name is automatically sequenced to avoid duplicate point names within a point group.
Remarks	Name of the point being created is provided in the Point Name argument. The name is defined by Collection::Group::Target. The active collection name is assumed if the Collection name is omitted. The point name must be consistent with SA's naming conventions. The vector specifies the points X, Y, Z components relative to the origin of SA's current working frame.

Arg	Type	Description	Definition
0	Point Name	Point Name	
1	Vector	Working Coordinates	

Construct a Point at line MidPoint

Function	
Description	Constructs a point at the mid-point of a line.
Return	Return success when the line object is found and the mid-point is created.
Description	Failure is returned if the line object is not found in the active collection or the mid-point name is not a valid SA name.

Remarks The line object must be in the active collection. When the name for mid-point already exists within the group, the mid-points name is automatically sequenced to avoid duplicate names within a point group.

Arg	Type	Description
0	Object Name	Line Name
1	Point Name	Point Name

Construct Point Groups from Vector Groups

Function Description Constructs points from vectors. The beginning and end points can be made from each vector. Flags control whether one or both begin and end points are built for each vector. Specify the vector groups with a Collection Object Name Ref List. More than one vector group can be specified. A string argument to use as a point group name prefix is optional. A Results Only argument is returned with a list of the point group names.

Return Description Success is returned after creating the points of the begin and or end points from the vectors.

Remarks All of the vectors in each of the identified vector groups are used to create points. Control over the beginning or end points on the vector are set with flags in the step. When set to TRUE their respective points are created. Both options can be set too TRUE.

Arg	Type	Description
0	Collection Object Name Ref List	Vector Groups
1	String	Optional Group Name Prefix
2	Boolean	Make Vector Begin Point
3	Boolean	Make Vector End Point
4	Collection Object Name Ref List	Point Groups

Construct a Point at Circle Center

Function Description Constructs a point at the center a circle.

Return Description Return success when the circle object is found and the center-point is created. Failure is returned if the circle object is not found in the active collection or the

center-point name is not a valid SA name.

The circle object must be in the active collection. When the name for center-point already exists within the group, the center-points name is automatically sequenced to avoid duplicate names within a point group. The point name is defined by Collection::Group::Target. The active collection name is assumed if the Collection name is omitted. The point name must be consistent with SA's naming conventions.

Remarks

Arg	Type	Description
0	Object Name	Circle Name
1	Point Name	Point Name

Construct Point at Intersection of Planes

Function Description Constructs a point at the intersection of three planes. The three plane objects must intersect (e.g., not be parallel planes) and be in the active collection.

Return Description Returns success when the three plane objects are found and the intersection-point is created. Failure is returned if the three plane objects do not intersect (e.g., two or more are parallel.) When any of the planes are not found in the active collection or the intersection-point name is not a valid SA name failure is the returned state for this step.

Remarks When the name for intersection-point already exists within the group, the intersection-point's name is automatically sequenced to avoid duplicate names within a point group. Specify the plane name in the first three arguments. The point name is defined by Collection::Group::Target. The active collection name is assumed if the Collection name is omitted. The point name must be consistent with SA's naming conventions.

Arg	Type	Description
0	Object Name	Plane 1 Name
1	Object Name	Plane 2 Name
2	Object Name	Plane 3 Name
3	Point Name	Point Name

Construct Point at Intersection of Two Lines

Function Description Constructs a point at the intersection of two lines. The two line objects must

Description not be parallel.

Return Description Returns success when the two line objects are found and the intersection-point is created. Failure is returned if the two line objects are parallel or the intersection-point name is not a valid SA name.

Remarks When the two lines are not in the same plane the intersection point is created at mid-point of the closest intersection between the lines. When the name for intersection-point already exists within the group, the intersection-point's name is automatically sequenced to avoid duplicate names within a point group.

Specify the line names with their collection names. The active collection name is assumed when the collection name is omitted. The point name is defined by Collection::Group::Target. The point name must be consistent with SA's naming conventions.

Arg	Type	Description
0	Collection Object Name	First Line Name
1	Collection Object Name	Second Line Name
2	Point Name	Resulting Point Name

Construct Point at Intersection of Plane and Line

Function Description Constructs a point at the intersection of plane and line.

Return Description Returns success when the plane and line objects are found and the intersection-point is created. Failure is returned if the plane and line objects are parallel or the intersection-point name is not a valid SA name.

Remarks The plane and line objects must not be parallel. The plane and line objects must be in the active collection. When the name for intersection-point already exists within the group, the intersection-point's name is automatically sequenced to avoid duplicate names within a point group.

Name the plane and line object and then the resulting point name. The point name is defined by Collection::Group::Target. The point name must be consistent with SA's naming conventions.

Arg	Type	Description
0	Object Name	Plane Name
1	Object Name	Line Name

Arg	Type	Description
2	Point Name	Resulting Point Name

Construct Points at Intersection of Circle and Line

Function	
Description	Constructs two points at the intersections of a circle and line.
Return	Returns success when the circle and line objects are found and the intersection-points are created. Failure is returned if the line (after being projected on to plane that the circle is in) does not intersect the circle. Failure is returned when the intersection-point name is not a valid SA name.
Description	The line is projected onto the plane that the circle is in and then two points are created at the intersection of the projected line and circle object. The line on the plane of the circle must intersect the circle. The circle and line objects must be in the active collection. Points 1 and 2 are constructed using a base point name e.g., Pt → Pt1 and Pt2. When the names for intersection-points already exist within the group, the intersection-point names are automatically sequenced to avoid duplicate names within a point group.
Remarks	Specify a base point name of the intersection-points being created. Points 1 and 2 are constructed in this step. The name uses the base point name e.g., Pt → Pt1 and Pt2. The base name is defined by Collection::Group::Target. The active collection name is assumed if the Collection name is omitted. The point name must be consistent with SA's naming conventions.

Arg	Type	Description
0	Object Name	Circle Name
1	Object Name	Line Name
2	Point Name	Base Point Name for results

Construct Points from Cylinder

Function	Three points are constructed at the center (mid), base and top points on the center-line of the cylinder.
Description	
Return	Return success when the cylinder object is found in the active collection and the three points (Mid, End A, and End B) are created in the Point Group object. Failure is returned when the cylinder object can not be found or the point group can not be created. The group name must follow SA's naming conventions.
Description	
Remarks	The points are automatically named Mid, End A, and End B. An input for a point group to contain the three points is provided as an input. If the Point Group and point names already exist in the active collection they are overwritten with the cylinders Mid, End A and End B points. The cylinder must

be in the active collection.

Specify the cylinder name in the active collection. Identify the point group name where the Mid, End A and End B points are constructed. If the point group is not found in the active collection it is created.

Arg	Type	Description
0	Object Name	Cylinder Name
1	Group Name	Group Name

Construct a Point at Projection of Point onto An Object

Function

Description

Constructs a point at the projection point on an object.

Return

Description

Returns success when the point to project and object is found in the job plus the projected point is created. Failure is returned when the point to project or object can not be found. When the projected point name can not be constructed the step returns failure.

The projection point is the point that is on the object and the closest to the point being projected (i.e., the nearest point.) The object must be in the active collection. When the name for projected-point already exists, the name is automatically sequenced to avoid duplicate names within a point group.

Remarks

Name the point to project. Point names are defined by Collection::Group::Target. The active collection name is assumed if the Collection name is omitted. Name the object to project the point too. Input the projected point name.

Arg	Type	Description
0	Point Name	Point to Project
1	Object Name	Object Name
2	Point Name	Resulting Point Name

Construct Point at Object Origin

Function

Description

Constructs a point at the origin of an object e.g., circle center, beginning of a line, centroid of a plane, center of a sphere etc.

Return

Success is returned when the object name is found and the origin point is constructed. Returns failure if the object name can not be found or the origin

Description	point can not be constructed.
	The constructed point's vector and individual X, Y, and Z components in Results Only fields.
Remarks	The name of the object is specified in the Object Name argument. A field for the resulting point name is provided.

Arg	Type	Description
0	Object Name	Object Name
1	Point Name	Resultant Point Name
2	Vector	Vector Representation
3	Double	X Value
4	Double	Y Value
5	Double	Z Value

Construct Points Shifted in Working Frame

Function	Constructs a list of points into a new collection::group. The points are shifted along a vector relative to the working frame.
Description	
Return	Returns success when the points in the list are found and the new group is created in the collection. Failure is returned if the points are not found or the new group can not be created. Partial success is returned when some of the points are found and others are not found.
Description	
	Create the list of points in the Point Name Ref List. An existing point group can be used to create the Point Name Ref List with the Construction Operations >> Other MP Types >> Make a Point Name Ref List From a Group function.
	The new group name is specified as collection and group name. When either the collection or group within the collection does not exist they get created by the step.
	Enter the vector or reference it from a prior step. It is common to reference a vector returned as a Result Only argument from a vector group or from an MP step that get object properties.
	The point names are retained unless the conflict with existing point names within the group. If the names conflict the constructed point names are automatically incremented.
Remarks	

Arg	Type	Description
0	Point Name Ref List	Original Points
1	Collection Object Name Group for New Points	
2	Vector	Shift Vector

Construct Points WildCard Selection

This function is used to create a new point group from existing point groups. Points are selected from existing point groups using a wildcard alpha-numeric search of point names. A point is created for each matched named point.

Function Description	A collection object name ref list argument is used to specify the point groups to search. Create a Collection::Group::Point Name object with the wild card selection criteria. The wild card search criteria is used to match the Collection::Group::Point name. A collection object name argument is used to set the new point group name. The source point group name can be prefixed to each point name by setting a flag in the step. When FALSE the source point group name is not added to the new point name. When TRUE the point group name is added to each point.
Return Description	Returns success after searching each specified point group for point names that match the search criteria are used to create new points. Failure is returned if the source point groups are not found.
Remarks	A Point Name object is used to hold the Collection, Group and Target wild card search criteria strings. Configure the wild card selection with '*' or '?' characters in the strings with the other specific search characters. Use the function to create a "Point Name from Strings." Reference the resulting point name object created in the "Point Name from Strings" in the argument. For example to search e collection use a '*' character for the collection name.

Create Hidden Point

Function Description	Computes a hidden point from two points and a reference index to a hidden point bar in the SA template file. An option to overwrite an existing point is when the name in the step conflicts.
Return Description	Returns success when the two inputs points and the index to the hidden point bar in the template file are found. Failure is returned when either point is not found or the reference to the hidden point bar is not found.
Remarks	Inputs for points 'End A' and 'End B' can be entered directly or referenced from prior steps. The index to the hidden point bar is an integer and refers to indexes in the Hidden Point Bar database. Specific bars are indexed on the User Options >>

Hidden Point Bar database.

Use the Overwrite flag to control whether the step overwrites an existing point. When false the step checks to see if the Point to Create exists and auto-indexes the target name to a new name.

Average a set of Groups

Function	Computes average points from a series of groups. Points with matching names
Description	are averaged. The averaged points have the same name in the new group. Returns success when the input point groups are found and the averaged points all meet the RMS, maximum absolute, and maximum average tolerances. Failure is returned when the input point groups are not found or when all the tolerance checks in this step fail. Partial Success is returned when at least one tolerance checks passes and one or more tolerance fails.
Return	The computed RMS, maximum absolute and maximum average statistics for
Description	the Points with matching names are returned as Read Only arguments in the step. An average point for each point name from the groups input in the step is constructed in the new point group. Tolerance inputs are available. When the tolerances are not met the step returns either failed or partial success. Use the return state to branch to methods of handling out of tolerance conditions.
Remarks	The statistics of RMS, maximum absolute and maximum average are returned as Read Only results fields for reporting or subsequent computations.

Lines...

Construct Line 2 Points

Function	A line is constructed from two points. The step has three inputs for the name of
Description	the line and each point. Success is returned when the line is constructed from the two points. If either of the points is not found or the new line name does not follow SA naming conventions the step returns failure.
Return	Input the name of the line being constructed. It is added into the active
Description	collection. The line direction is defined from the first point to the second point. If there is a name conflict the name of the new line is automatically incremented. If the two points have identical coordinates a line object is
Remarks	created with zero length. Analysis with a line of length 0 is not defined.

Construct Line - Project Line to Object Reference Plane

Function	Constructs a line on a plane. Inputs for the name of the line to construct, the
Description	line to project and the plane to project the line too are available.

Return Description	Success is returned when the line to project and plane being projected too are found. A return of failure is the result when either the plane or the line are not found.
Remarks	The new line on the plane is added to the active collection. The Object Reference Plane must be a plane. Projecting a line that is normal to the plane creates a line with a length of 0. The line is constructed at the projection point. Analysis with a line of length 0 is not defined.

Construct Line - Normal to Object through Point

Function Description	Constructs a line that is parallel to an object's normal direction and passes through a point.
Return Description	Returns success when the object and point are found. Failure is returned when the object or point are not found.
Remarks	The new line is parallel to the object's normal direction and passes through the point. The line is added to the active collection. If the point is on the object's normal vector the constructed line has a length of 0. Analysis to a line of length 0 is not defined.

Planes...

Construct Plane

Function Description	Constructs a plane based on vector inputs for a plane center and normal direction. A dimension for the plane boundaries allows the graphical size of the plane to be set.
Return Description	Returns success when the center and normal vectors are properly defined (e.g., non-zero length vectors). Failure is returned if the plane is not created.
	Set the center of the plane and its positive normal direction with the input vectors. The plane's graphical dimension (i.e., representation in SA's graphics view) has to be a positive double number. Plane definitions have infinite boundaries for analysis purposes.
Remarks	The new plane is added to the active collection. If a name conflict occurs in the active collection the new plane's name is automatically incremented.

Construct Plane, Normal to Object, Through Point

Function Description	Construct a plane where its normal direction is parallel to another object's normal direction. The plane passes through the point in the step's list of arguments. A dimension for the plane boundaries allows the graphical size of the plane to be set.
Return Description	Returns success when the object and point are found. Failure is returned when the plane can not be constructed.
Remarks	The constructed plane is added to the active collection. If a name conflict occurs in the active collection the new plane's name is automatically

incremented.

The plane's graphical dimension (i.e., representation in SA's graphics view) has to be a positive double number. Plane definitions have infinite boundaries for analysis purposes.

Construct Planes, Bounding Point Group

Function Description	Finds the high and low points from the group based on the plane's normal direction. Bounding planes are constructed such that they passing through the high and low points and are parallel to the input planes. An override for target offsets is available.
Return Description	Returns success when the plane and point group are found. Failure is returned when the plane can not be constructed.
Remarks	<p>The two bounding planes are added to active collection. Arguments for the name of the high and low planes allow them to be named explicitly. If name conflicts occur the names are automatically incremented based on SA's naming conventions.</p> <p>The point group must have at least one point in it. When only one point is found both high and low planes are the same.</p> <p>When the override target offset flag is false the targets offset are used to set the high and low bounding plane offsets along the input planes normal direction. The optional target offset is used if the override target offset flag is true.</p>

Circles...

Construct Circle

Function Description	Construct a circle by defining its name, center, normal vector direction and radius.
Return Description	Returns success when the circle is constructed. Failure is returned if the circle is not added to the database.
Remarks	Use vectors to define the circle center and normal direction. Construct the vectors directly from doubles or reference vectors in other steps within the measurement plan. The radius of the circle is defined by a positive non-zero floating point number i.e., double.

Cylinders...

Construct Cylinder

Function Description	Construct a cylinder by defining its name, two center end points, diameter and length.
----------------------	--

Return Description	Returns success when the cylinder is constructed. Failure is returned if the cylinder added to the database.
Remarks	Use vectors to define the center end points of the cylinder. The end points are the centers of the circles at the top and bottom of the cylinder. Construct the two vectors directly from doubles or reference vectors in other steps within the measurement plan. The diameter and length of the cylinder are defined by positive non-zero floating point numbers i.e., doubles.

Spheres...

Construct Sphere

Function Description	Construct a sphere by defining its name, center, and radius.
Return Description	Returns success when the sphere is constructed. Failure is returned if the sphere is not added to the database.
Remarks	Use a vector to define the sphere's center. Construct the vector directly from doubles or a reference vector from another step within the measurement plan. The radius of the sphere is defined by a positive non-zero floating point number i.e., double.
Remarks	The named sphere is added to the active collection.

B Splines...

Construct BSpline From Points

Function Description	Constructs a B-Spline from a list of points. The B-Spline will pass exactly through each point in the list.
Return Description	Returns success when the spline is constructed. Failure is returned if the spline is not added to the database.
Remarks	The sequence of points in the list defines their order as inputs for the spline. The B-Spline will pass exactly through each point in the list. At least four different points are required to construct the B-Spline. Build a point list from a group with the Construct >> Other MP Types >> Make a Point Name Ref List From a Group function. The B-Spline is added to the active collection.

Construct B-Spline From Several B-Splines

Function Description	Constructs a B-Spline from a list of existing B-Spline. The order of the B-Splines is important. An object name argument for the constructed B-Spline controls the B-Splines name. Control whether the B-spline is closed is provided with a flag. When set to true the function closes the B-Spline. The end point of the last B-Ppline is joined to the begin point of the first B-Spline.
----------------------	--

Return Description	Returns success when the spline is constructed. Failure is returned if the list of splines is not found.
	The sequence of B-Spline in the list defines their order as inputs for the constructed spline. The B-Spline will pass exactly through each spline in the list.
Remarks	Build a Collection Object Name Ref List to specify which splines are used. The B-Spline is added to the active collection.

Construct B-Spline From Lines

Function Description	Constructs a series of B-Splines from a list of existing lines. An object name argument controls the B-Spline name prefix.
Return Description	Returns success when the splines are constructed. Failure is returned if the list of lines is not found.
	The sequence of lines in the list defines their order as inputs for the constructed splines. The B-Splines will match exactly each line in the list. The important difference is B-Spline have a specific length and can be used to construct surfaces.
Remarks	Build a Collection Object Name Ref List to specify which lines are used. The B-Splines are added to the active collection. Their names are returned in a Results Only Collection Object Name Ref List.

Surfaces...

Construct Surface From BSplines

Function Description	Constructs and names a surface from a list of B-Splines. At least 4 B-Splines are required to construct the surface.
Return Description	Returns success when the surface is constructed. Failure is returned if the surface is not added to the database.
	Add B-Splines to the Object Name Reference List in the order they are used within the surface. A reference to an existing Object Name Reference List is an alternate method to input the B-Splines. This method/function is expecting the sequence of the B-Splines in a ribbon type configuration.
Remarks	The named surface is added to the active collection.

Frames...

Construct Frame

Function Description	Makes a frame from a transform. Name the frame with a collection object name.
----------------------	---

Return Description	Success is returned after adding the frame to the job. If the frame name already exists the function returns failed.
Remarks	Specify the frame name with a collection object name argument. Either enter or reference a Transform from a prior step.

Construct Frame on Instrument Base

Function Description	Adds a frame on an instruments base frame. The axes of the frame are aligned with the instrument's base coordinate system. A name for the frame is an option.
Return Description	Success is returned after adding the frame in the database. When the instrument index is not found (e.g., in the active collection) the step returns failure.
Remarks	The instrument's index within the active collection is used to define which instrument base the frame is constructed from. An optional name for the frame is available. Naming the frame is useful as a reference in subsequent steps to get an instrument's position and orientation relative to the working frame.

Construct Frame on Object

Function Description	Constructs a frame on an object. Name the frame with an optional argument. The object's inherent orientation is used to define the directions of the frame axes.
Return Description	Returns success after the frame is constructed. Failure is returned if the object is not found or the frame's name is not properly formed.
	As an example the z-axis of a frame constructed on a plane is aligned with the plane's normal direction. The origin of the frame on a plane is at the plane's centroid.
	To make a constructed frame the working frame use the Utility Operations >> Set Working Frame function. Get the frame's properties with Construction Operations >> Other MP Types >> Get Working Transform of Object (Fixed XYZ).
Remarks	The frame is constructed in the active collection.

Construct Frame, 3 Points

Function Description	Construct a frame using 3 points. The first point defines the origin, the second sets the primary axis and the third point is used to clock the secondary axis. An optional argument names the frame.
Return Description	Returns success after the frame is constructed. Failure is returned if any of the points are not found or the frame's name is not properly formed.
Remarks	Define the axis sequence with the first argument (e.g., Origin-X-XY, Origin-Y-YZ, etc.) Configure the sequence of the axes to fit the required frame with the

first argument.

The 3 points must have different coordinates. A frame constructed with duplicate points will have an unpredictable orientation.

To make a constructed frame the working frame use the Utility Operations >> Set Working Frame function. Get the frame's properties with Construction Operations >> Other MP Types >> Get Working Transform of Object (Fixed XYZ).

The frame is constructed in the active collection.

Construct Frame, at Point, with working Z, and clocked axis

Function Description	Constructs a frame with the origin at a point. The new frames Z-axis is parallel with the current working frame. Select the positive/negative X or Y axis as the secondary axis. Provide a point which lies on the plane defined by the Z and selected secondary axes. An optional argument names the frame.
Return Description	Returns success after the frame is constructed. Failure is returned if any of the points are not found or the frame's name is not properly formed.
	The clocking point sets the direction for the secondary axis. The clocking point may not lie exactly along the secondary axis but it will lie in the plane defined by the Z and the selected secondary axis (e.g., +X, -X, +Y, -Y).
	The 2 points (origin and clocking) must have different coordinates. A frame constructed with duplicate points will have an unpredictable orientation. Do not set +Z or -Z as the clocking axis.
	To make a constructed frame the working frame use the Utility Operations >> Set Working Frame function. Get the frame's properties with Construction Operations >> Other MP Types >> Get Working Transform of Object (Fixed XYZ).
Remarks	The frame is constructed in the active collection.

Construct Frame, Known Origin, Object Direction, Object Direction

Function Description	Construct a frame by defining a point and its coordinates in the constructed frame. Then set the primary and secondary axes to be parallel with object (e.g., plane, line, circle, cylinder, cone etc.) directions. The objects used to set the primary and secondary axis directions must not be the same objects. Select the primary and secondary axes from any of the following +X, -X, +Y, -Y, +Z or -Z axis directions. An optional argument names the frame.
Return Description	Returns success after the frame is constructed. Failure is returned if the point or objects are not found.
Remarks	The frame is constructed with its origin offset from a point by a coordinate vector. Input the point name its coordinate vector by entering them or referencing them from prior steps.

Primary and secondary axis directions are controlled by object directions. For example the +Z-axis is set to be parallel to a circle's normal direction by selecting the +Z direction and entering or referencing the circles name.

Configure the frame's axial directions consistently. A frame constructed with the same axis for the primary and secondary direction will have an unpredictable orientation. Take care that the object directions used to set the primary and secondary axes are not parallel.

To make a constructed frame the working frame use the Utility Operations >> Set Working Frame function. Get the frame's properties with Construction Operations >> Other MP Types >> Get Working Transform of Object (Fixed XYZ).

The frame is constructed in the active collection. Objects used to set axis directions need to be in the active collection.

Perimeters...

Construct Perimeter From Points

Function Description	Builds either an open or closed perimeter from a list of points. Set the Open Perimeter flag to true to construct an open perimeter or false for a closed perimeter. An argument is provided to name the perimeter.
Return Description	Returns success after the perimeter is constructed and added to the database. Failure is return when all of the points in the list are not found.
	The sequence of points in the list determine to perimeter boundary. A Point Group can be converted into Point Name Reference List with the Construction Operations >> Other MP Types >> Make a Point Name Ref List From a Group function.
	Name the perimeter and the reference its name in subsequent steps to drive the measurement system.
Remarks	The perimeter is added into the active collection.

Vector Groups...

Construct a Vector Group - Group to Group Compare

	Computes and saves the differences between points with the same name from two point groups. The analysis results are saved in a Vector Group. An argument is provided to name the Vector Group.
	Statistics from the vector group are returned in Results Only fields. Returned statistics include the number of vectors (i.e., number of matched point names), plus the computed RMS Deviation, Maximum Deviation, and Average Deviation errors. Statistics are based on the samples total point to point deltas.
Function Description	Inputs for RMS, Maximum, and Average deviation error tolerances are provided. The steps return condition depends on whether the tolerances are

	met or failed.
Return Description	The step returns success when both point groups are found and the RMS, Maximum, and Average deviation tolerance conditions are met. Failure is returned when one or both the point groups are not found. Partial Success is returned after the vector group is constructed and one or more of the tolerances are not met.
	Point groups A and B have to be from the active collection. The named vector group is added to the database in the active collection.
Remarks	Tolerance deviations are computed on the samples total point to point deltas. The step will ignore deviation tolerance tests (RMS, Maximum, or Average) when the value is 0.0.

Construct a Vector Group - Area Profile Check

	Creates an Area Profile Vector Group from a set of reference vectors compared against a set of sample vectors. The area is specified with a radius parameter. A delta is computed for e sample vector within the radius of a reference vector.
Function Description	The reference vectors are used to base differences between themselves and any vector in the sample vectors within an area around it. For each vector within the radius (e.g., 5 inches) from a reference vector the difference between the magnitudes is used to create a profile vector. The profile vector direction is set to the direction of the sample vector. Its length is the difference between the reference vector and its sample vector.
Return Description	Returns success when Area Profile Vector Group is created. Failure is returned if the references or sample vectors to check is not found or when the area profile vector group has deltas greater than the area tolerance parameter.
	A Vector Name Ref List argument is used to identify the reference vectors. A Collection Vector Group Name Ref List is used to specify the sample vectors to check. Specify the area radius (double) to search around each reference vector for a sample vector to compare. A tolerance parameter is used to check if the profile check passes.
Remarks	As an example: Compute the signed difference in lengths between the sample vectors and the reference vector (e.g., $-0.010 - (-0.008) = -0.002$). The profile vector is constructed in the direction of the sample vector with a -0.002 delta length.

Construct a Vector in Working Coordinates(Begin/Delta)

Function Description	Constructs a named vector from a beginning point and the vectors delta components. The delta components are specified in working coordinates. A flag to control the sign of the resulting vector sets the positive or negative sense of the vector. The vector direction is not changed with the sign flag only its sign when reported.
Return	Returns success after constructing the vector. Failure is returned when the

Description	vector group or vector name are not valid SA names.
	The positive or negative sense of vector delta is controlled with the flag. When positive the vector will report as positive. When the flag is negative the vector's magnitude is reported as negative. The begin point and delta coordinate components are relative to the working frame.
Remarks	Specify the Vector Group name with a Collection Vector Group Name Object and the vectors name with a string. Duplicate vector names are allowed within a vector group.

Construct a Vector in Working Coordinates(Begin/Direction/Mag.)

Function Description	Constructs a named vector from a beginning point, relative direction vector and a signed magnitude. The direction components are specified in working coordinates. A double argument of the vectors signed magnitude is multiplied against the direction components. The direction components should be specified as a positive direction unit vector.
Return Description	Returns success after constructing the vector. Failure is returned when the vector group or vector name are not valid SA names.
	The positive or negative sense of vector delta is controlled with the signed magnitude argument. The begin point coordinate and direction components are relative to the working frame.
	The direction components should be specified as a positive direction unit vector. The double argument for the vector's signed magnitude is multiplied against the direction components. As an example consider a beginning point of [0 0 0] with a direction vector of [1 1 1]. With a signed magnitude of -5.0 the resulting vector has a length of -5.0 and a components of [-2.9 -2.9 -2.9] (one digit of precision).
Remarks	Specify the Vector Group name with a Collection Vector Group Name Object and the vectors name with a string. Duplicate vector names are allowed within a vector group.

Construct Vectors WildCard Selection

	This function is used to create a new vector group from existing vector groups. Vectors are selected from existing vector groups using a wildcard alpha-numeric search of vector names. A vector is created for each matched name vector.
	A collection object name ref list argument is used to specify the vector groups to search. Create a Collection::Group::Point Name object with the wild card selection criteria. The wild card search criteria is used to match the Collection::Group::Vector name. A collection object name argument is used to set the new vector group name. The source vector group name can be prefixed to each vector name by setting a flag in the step. When FALSE the source vector group name is not added to the new vector. When TRUE the vector group name is added to each vector.
Function Description	

Return Description	Returns success after searching each specified vector group for vector names that match the search criteria are used to create new vectors. Failure is returned if the source vector groups are not found.
Remarks	A Point Name object is used to hold the Collection, Group and Target wild card search criteria strings. Configure the wild card selection with '*' or '?' characters in the strings with the other specific search characters. Use the function to create a "Point Name from Strings." Reference the resulting point name object created in the "Point Name from Strings" in the argument. For example to search e collection use a '*' character for the collection name.

Construct a Vector Group From a Relationship

	This function constructs a vector group from a relationship. Relationship stats including sampling, outlier rejection, and fit constraints are reported in the Vector Group Properties notes field. Relationship properties are considered when making the vector group. Projection properties, Target Offsets, Tolerances, Sub-Sampling, Outlier Rejection, and Fit Constraint properties are each used when the individual vectors are created.
Function Description	The relationship is identified with a Collection Object Name argument. The vector group is named with a Collection Vector Group Name argument.
Return Description	Success is returned after the vector group is created from the relationship. Failure is returned when the relationship is not found or the vector group can not be created.
Remarks	Configure the relationship's properties before creating the vector group from it. In doing so the vector group automatically inherits its properties and stats on specific sampling, fit constraints, and rejection outliers are retained. These parameters are also included in reports from the vector group.

Other MP Types...

Make a Boolean

Function Description	Makes a Boolean data variable. Set the variable to true or false. The results field returns the variables current state.
Return Description	Returns success
Remarks	When creating the variable either enter the value or reference it from a prior step. Reference the Result Only argument to get the current state of the variable.

Make a Integer

Function Description	Makes an integer data variable. Set the variable to a positive, zero or negative whole number. The results field returns the variables current state.
Return	Returns success

Description

Remarks When creating the variable either enter the value or reference it from a prior step. Reference the Result Only argument to get the current state of the variable.

Make an Integer from a String

Function Description Makes an integer data variable from a string. Set the variable to a positive, zero or negative whole number. The results field returns the variables current state.

Return Description Returns success

Remarks When creating the variable either enter the value or reference it from a prior step. Reference the Result Only argument to get the current state of the variable.

Make a Double

Function Description Makes a double precision floating point data variable. Set the variable to a rational floating point number. The results field returns the variables current state.

Return Description Returns success

Remarks When creating the variable either enter the value or reference it from a prior step. Reference the Result Only argument to get the current state of the variable.

Make a Double from an Integer

Function Description Converts an integer variable to a double precision floating point number. The resulting double value is loaded into the Results Only argument.

Return Description Returns success

Remarks Needed to convert integer data to a floating point number for math operations.

Make a Boolean from an Integer

Function Description Converts a Boolean variable to an integer number. The resulting integer value is loaded into the Results Only argument. A False value is converted to a 0 integer. A True value is converted to a 1.

Return Description Returns success

Remarks	Needed to convert a Boolean data to an integer number for math operations. A False value is converted to a 0 integer. A True value is converted to a 1.
---------	---

Make a String

Function Description	Concatenates two source strings into one. Arguments for the first and second parts of the resulting string are provided. The concatenated string is returned in a Results Only field.
----------------------	---

Return Description	Returns success.
--------------------	------------------

Enter or reference prior strings to merge them together to get the concatenated string. Reference the Results Only field in subsequent steps.

Remarks	An example: "PT" + "Num" = "PTNum"
---------	------------------------------------

Make String from Integer

Function Description	Converts an integer number into a string (e.g., -1, becomes "-1"). The string representation is accessible as a "Result Only" field argument. Make String from Integer to use an option digits format. It can pad the string with leading 0's.
----------------------	--

Return Description	Returns success when the integer is converted into a string.
--------------------	--

Remarks	Enter or reference integers or step numbers from other MP steps as the input for the function. Specify the minimum number of digit in the string. The string add zeros to the beginning of the string.
---------	--

Make String from Double

Function Description	Converts a floating point number (i.e., double) into a string (e.g., -1.0062, becomes "-1.0062"). Control the amount of decimal precision with an integer argument in the step. The resultant string is accessible as a Result Only field.
----------------------	--

Return Description	Returns success when the double is converted into a string. Returns failure if the number is not converted.
--------------------	---

The double is rounded to the nearest digit based on the number of significant digits. For example 2 significant digits past the decimal point results in a string of "-1.01" from a double value of -1.0062.

Remarks	Enter or reference double values from other MP steps as the input for the function.
---------	---

Make String from Decimal Degrees Angular Value

Function Description	Create a string from a double value. A double is used for decimal degree angles. Select the output units from a list. Control the amount of decimal precision of the remainder with an integer argument in the step. The resultant
----------------------	--

string is accessible as a Result Only field.

Return Description	Returns success when the decimal degree angle is converted into a string. Returns failure if the angle is not converted.
	The output units include Decimal Degrees, Degrees::Minutes::Seconds, Degrees::Minutes, Radians, and Gons.
	The remainder of the decimal degree angle is a double precision value. The remainder is rounded to the nearest digit based on the number of significant digits.
Remarks	Enter or reference double values from other MP steps as the input for the function.

Make an Incremented String

Function Description	A string incremented from a base string is returned in a Result Only argument. SA's auto-incrementing utility is called to setup a base string as the next valid name.
Return Description	Returns success when the name is incremented. Failure is returned if the name is not incremented.
Remarks	Enter or reference an existing object, group or target name from other MP steps as the input for the function. Use the results to configure inputs for other MP steps or to guide a user through a process.

Make a System String

Function Description	Creates a string from a list of system information choices. The possible options include SA Version, SA File Name, MP File Name, System Date + Time, Date, and Time. The string is accessible as a Result Only argument for use in other steps.
Return Description	Returns success
Remarks	This function returns a Result Only argument of the system string. It provides an easy to use set of strings with specific system information. Include this data in report tables and other customizable reports.

Concatenate Strings

Function Description	Creates a new string from a list a strings. The list of strings can be entered or referenced from other steps. The string is accessible as a Result Only argument.
Return Description	Returns success
Remarks	This function returns a Results Only string argument of the concatenated

strings.

Make a Strings from a Point Name

Function Description	Creates three strings from a point name. Strings with the collection, point group and target names are returned in Results Only arguments. An input for a Collection::Group::TargetName is split into the component strings.
Return Description	Returns success when the source Point Name is split. Failure is returned if the name can not be split.
Remarks	The active collection is returned if it was omitted. Enter or reference the source Point Name from other MP steps.

Make a Strings from Collection Object Name

Function Description	Creates two strings from a collection object name. Strings with the collection and object names are returned in Results Only arguments. An input for a Collection::Object Name is split into the component strings.
Return Description	Returns success when the source object name is split. Failure is returned if the name can not be split.
Remarks	The active collection is returned if it was omitted in the source name. Enter or reference the source Collection Object Name from other MP steps.

Make a Point Name from Strings

Function Description	Creates a composite point name from three input arguments. Inputs for a Collection::Group::TargetName are merged into a Point Name object.
Return Description	Returns success when the Point Name is created. Failure is returned if the name can not be created. Confirm that the string inputs only contain valid name characters.
Remarks	<p>Point Name objects are used to completely specify a point or target. They are used as inputs for measurements and analysis functions.</p> <p>When the collection string is omitted the active collection name is assumed. Enter or reference each component string (e.g., collection, group, and target name) as inputs for the composite Point Name. Reference the Resultant Point Name in subsequent analysis or measurement MP steps.</p>

Make a Point Name - Runtime Select

Function Description	Function allows the user to select a point from SA with runtime selection methods. The point name is returned in the Resultant Point Name argument.
Return Description	Returns success when the point is selected. Failure is returned if the user escapes out of the selection.

Remarks	Specify a prompt for the user to select a point. SA displays the prompt and asks the user to select a point. Only one point is allowed to be selected. As soon as the point is selected the function returns. If the user hits the escape key the function returns failure.
---------	---

Make a Point Name Ref List

Function Description	Builds a list of point names one point name at a time. Each point name in the list is built or referenced as a Collection::Group::TargetName. Use the enter method to build the list. Reference the Point Name Ref List from subsequent MP steps that require a list of point names.
Return Description	Return success when the Point Name Ref List is constructed. Failure is returned if one or more of the point names in the list are not found.
	Use this function to construct a list of point name references that cross group and collection boundaries. Reference the
Remarks	Each point name in the list is either entered directly with the Point Name List Editor dialog or is referenced from other Point Name objects in the MP.

Make a Point Name Ref List From a Group

Function Description	Builds a Point Name List directly from a Point Group. The resulting list contains a complete of each point in the group. Each point name in the list references the point as Collection::Group::TargetName. A Result Only argument of the constructed Point Name List is provided for other functions to reference.
Return Description	Return success when the Point Name List is constructed. Failure is returned if the point group is not found.
	Use this function to setup a Point Name List for a group of points. A list of point names are commonly used in analysis and measurement functions. The point list is built in the same order that the points are saved in the group. Reference the Resultant Point Name Reference List in subsequent functions that require a list of points as an input.
Remarks	When the collection name is omitted the group is assumed to be in the active collection.

Make a Point Name Ref List - Runtime Select

Function Description	Function allows the user to select points from SA with runtime selection methods. The point names are returned in the Resultant Point Name Ref List argument.
Return Description	Returns success when the points are selected. Failure is returned if the user escapes out of the selection.
Remarks	Specify a prompt for the user to select the points needed for the process. SA displays the prompt and asks the user to select the points. Only points are allowed to be selected. F2, F3, selection off the tree, and graphically selection all available to the user. The user clicks the Enter key to end the selection of

points. When the Enter key is selected the function returns with a Point Name Ref List of all the selected points. The list is ordered in the selection sequence. If the user hits the escape key the function returns failure.

Make a Vector Name Ref List From a Vector Group

Function Description	Builds a Vector Name Ref List from an existing Vector Group. The resulting list contains a complete of each vector in the group. Each vector name in the list references the vector as <code>Collection::Group::VectorName</code> . A Result Only argument of the constructed Vector Name Ref List is provided for other functions to reference.
Return Description	Return success when the Vector Name Ref List is constructed. Failure is returned if the vector group is not found.
Remarks	<p>Use this function to setup a Vector Name Ref List for a group of vectors. A list of vector names are commonly used in analysis and measurement functions. The vector ref list is built in the same order that the vectors are saved in the group. Reference the Vector Name Ref List in subsequent functions that require a list of vectors as an input.</p> <p>When the collection name is omitted the group is assumed to be in the active collection.</p>

Make a Collection Object Name from Strings

Function Description	Builds a Collection Object Name from two strings. A string for the collection name and one for the object name are combined together to specify an object name as <code>Collection::ObjectName</code> . The combined Collection Object Name is returned as a Result Only argument.
Return Description	Returns success when the collection object name is constructed.
Remarks	<p>Enter or reference the two input strings arguments, collection and object name respectively.</p> <p>A Collection Object Name is special data type used in Measurement Plans. They combine a collection name and an object name into a single variable. Collection Object Names are used as inputs for measurement plan functions. Instead of having to always specify the collection and object name in each function the Collection Object Name reduces the number of arguments needed in each function.</p> <p>Reference the resultant Collection Object Name argument from subsequent MP functions. The Collection Object Name is returned as a Result Only argument by the step.</p>

Make a Collection Object Name - Runtime Select

Function Description	Function allows the user to select an object from SA with runtime selection methods. The <code>Collection::Object</code> name is returned in the Resultant Collection
----------------------	---

	Object Name argument.
Return Description	Returns success when the object is selected. Failure is returned if the user escapes out of the selection.
Remarks	Specify a prompt for the user to select an object. SA displays the prompt and asks the user to select an object. Only one object is allowed to be selected. F2, F3, selection off the tree, and graphically selection all available to the user. As soon as the object is selected the function returns. If the user hits the escape key the function returns failure.

Make a Collection Object Name Reference List- Runtime Select

Function Description	Function allows the user to select objects from SA with runtime selection methods. The object names are returned in the Resultant Collection Object Name Ref List argument.
Return Description	Returns success when the objects are selected. Failure is returned if the user escapes out of the selection.
Remarks	Specify a prompt for the user to select the objects needed for the process. SA displays the prompt and asks the user to select the objects. Only objects are allowed to be selected. F2, F3, selection off the tree, and graphically selection all available to the user. The user clicks the Enter key to end the selection of points. When the Enter key is selected the function returns with a Collection Object Name Ref List of all the selected points. The list is ordered in the selection sequence. If the user hits the escape key the function returns failure.

Append two Collection Object Name Ref Lists

Function Description	Add two Collection Object Name Ref Lists together to form a new list. The new list is returned in a Results Only argument.
Return Description	Returns success after adding the two collection object name to the lists together as setting up the Results Only arguments. Failure is returned when either of the source collection object ref list names are not found.
Remarks	Enter the reference lists directly or reference existing collection object name ref lists. The lists can be created in the Collection Object Name List Editor directly. To use the resulting ref list reference the Results Only argument from subsequent functions. The sequence of names in the list is how objects are input into functions that require a list of object names.

Make a Collection Object Name Ref List from all Groups in a Collection

Function Description	Makes a Collection Object Name Ref List from all the Groups in a Collection. The object is available as a Results Only argument.
Return Description	Returns success after building the Collection Object Name Ref List and set it up in a Results Only argument. Failure is returned when collection object is not found.

Enter the collection name or reference it from other step arguments.

Remarks To use the resulting ref list reference the Results Only argument from subsequent functions. The sequence of names in the list is how objects are indexed in the database.

Add a Collection Object Name to a Ref List

Function Description Add a Collection Object Name to an existing Collection Object Name List. The Collection Object Name is appended to a ref list. The list can be an existing list that is referenced or created the Collection Object Name List argument. The new Collection Object is added to the bottom of the list.

Return Description Returns success after adding the collection object name to the list. Failure is returned when the collection object name or the existing collection object name ref list is not found.

Remarks Enter the reference list directly or reference an exist collection object name ref list. The list can be created in the Collection Object Name List Editor directly.

The new collection object name is added to the list. To use this reference list reference the list argument from subsequent functions. The sequence of names in the list is how objects are input into functions that require a list of object names.

Make a Collection Object Name Ref List

Function Description Builds a list of Collection Object Names. Enter or reference the names in the Collection Object Name List Editor.

Return Description Returns success after building the list of Collection Object Names.

Enter directly the collection and object names of a series of objects in the Collection Object Name List Editor directly or reference previously defined Collection Object Names to build the list.

Remarks Reference the list as constructed in the Collection Object Name Editor from subsequent functions that require a list of object names as an input. The sequence of names in the list is how objects are input into functions that require a list of object names.

Make a Transform from Doubles (Fixed XYZ)

Function Description Constructs a Transform Object from a set of 6 double values. Specify the X, Y, Z, Rx (Roll), Ry (Pitch), and Rz (Yaw) as inputs for the transform object. A transform object is returned as a Result Only argument.

Return Description Returns success when the transform object is constructed. Failure is returned if the transform object is not built.

Remarks Enter directly or reference each of the 6 input components for the transform object. Use the resultant transform object to move objects or instruments

relative to the working frame. Apply the Result Only Transform argument by referencing it from other MP steps.

Translation components are used in current job units (e.g., inches, mm, m, etc.)
Rotation components are input as decimal degrees.

Transform Object's are passed into and out of MP subroutines as arguments.
Data Share files can import and export Transform objects directly.

Get Working Transform of Object (Fixed XYZ)

Function Description	Returns a transform of an object relative to the currently working frame. Specify the object name as a Collection Object Name and its inherent frame is returned as a transform in a Result Only argument.
Return Description	Returns success when the transform is constructed and available in the Result Only argument. Failure is returned when the object either is not found or it does not have an inherent transform.
	Input the object name as a Collection Object Name by entering it or referencing a prior Collection Object Name.
	A frame with its origin at a plane's centroid with the frame's Z axis aligned to the planes normal direction is an example of an object's inherent transform. Getting a frame's position and orientation relative to the working frame is another application for this MP function. Specify the frames name and the collection that it is in and this function returns its transform relative to the working frame.
Remarks	Use the transform object to report, analyze or construct other geometry based on the object's current position and orientation. Get access to the transform components with the Decompose Transform into Double or Vector MP functions.

Decompose Transform into Doubles (Fixed XYZ)

Function Description	Decomposes transform objects into their 6 component doubles [e.g., X, Y, Z, Rx (Roll), Ry (Pitch), and Rz (Yaw)]. The position and orientations are presented relative to the working frame. Each component double is accessible as Result Only arguments.
Return Description	Returns success when the component doubles are each accessible as Result Only arguments.
	Reference an existing transform object or enter the 6 components directly. Each component double is accessible as a Result Only argument after the function is executed.
Remarks	Resultant position and orientation doubles are provided relative to the working frame and in the current job units. The orientation doubles (Rx, Ry, and Rz)

are output in rotation angles about each component axis in decimal degrees.

Decompose Transform into Vectors (Fixed XYZ)

Function Description	Decomposes transform object into two vectors (one for position and another for orientation). The position and orientations are presented relative to the working frame. Both component vectors are accessible as Result Only argument. The vectors are provided relative to the working frame.
Return Description	Returns success when the component vectors are each accessible as Result Only arguments.
	Reference an existing transform object or enter the 6 components directly. Position and orientation vectors are accessible as a Result Only arguments after the function is executed.
Remarks	Resultant position and orientation vectors are provided relative to the working frame and in the current job units. The orientation vector is output in rotation angles (e.g., Rx, Ry, Rz) about each component axis in decimal degrees.

Decompose World Transform Operator into Doubles (Fixed XYZ in World)

Function Description	Decomposes a World Transform Operator into its 7 component doubles [e.g., X, Y, Z, Rx (Roll), Ry (Pitch), Rz (Yaw), and Scale]. Each component double is accessible as Result Only arguments. The position, orientation, and scale are presented relative to the world frame.
Return Description	Returns success when the component doubles are each accessible as Result Only arguments.
	Reference an existing World Transform Operator or enter the 6 position and orientation components directly. Each component double is accessible as a Result Only field after the function is executed.
	World Transform Operators are generally used as output arguments from best fit functions. They are applied to objects and instruments to move them from their current positions to the ones solved for in best fit operations.
Remarks	Resultant position and orientation doubles are provided relative to the world frame and in the current job units. The orientation doubles (Rx, Ry, and Rz) are output in rotation angles about each component axis in decimal degrees. The Result Only scale component is the ratio of the World Transform Operator scales relative to the default World frame.

Decompose World Transform Operator into Vectors (Fixed XYZ in World)

Function Description	Decomposes a World Transform Operator into its 2 component vectors (e.g., position and orientation.) The relative Scale of the World Transform Operator to the World frame is also provided. Each component is accessible as Result Only arguments. The position, orientation, and scale are presented relative to the world frame.
----------------------	---

Return Description	<p>Returns success when the vectors and scale are each accessible as Result Only arguments.</p> <p>Reference an existing World Transform Operator object or enter the 6 components directly. Each component vector is accessible as a Result Only field after the function is executed.</p> <p>World Transform Operators are generally used as output arguments from best fit functions. They are applied to objects and instruments to move them from their current positions to the ones solved for in best fit operations.</p> <p>Resultant position and orientation vectors are provided relative to the working frame and in the current job units. The orientation vector is output in rotation angles (e.g., Rx, Ry, Rz) about each component axis in decimal degrees. The Result Only scale component is the ratio of the World Transform Operator scales relative to the default World frame.</p>
Remarks	

Make a Vector from Doubles

Function Description	Constructs a vector from three double values. The vector is accessible as a Result Only argument.
Return Description	Returns success after the vector is constructed.
Remarks	Enter three doubles directly or reference them from other MP step arguments. Reference the resulting vector argument from subsequent MP functions.

Decompose a Vector into Doubles

Function Description	Decomposes a vector into three (X, Y, and Z) double values. The component values are accessible as a Result Only arguments.
Return Description	Returns success after the vector is decomposed and the values are accessible.
Remarks	Enter the vector directly or reference it from other MP step argument. Reference the resulting X, Y, and Z arguments from subsequent MP functions.

Split String into Two Strings

Function Description	Divides an input string and copies the parts into two separate strings. The string is split after a number of characters. Characters are copied into the first string up to and including the character at the index count. Characters after the count index are copied into the second string. The Dividing Character Index argument is an integer data type. Result Only arguments to the two new strings are provided.
Return Description	Returns success after the string is divided. Failure is returned if the character index is more than the number of characters in the input string. Failure is also returned when the index is less than 0.

Enter or reference a string from another MP step for the input string. The index is an integer. It is used as a count based index. To divide a ten character string into two strings of five characters use five as the index. The first five characters are put into the first string. The second set of five characters is copied into the second string. Reference the two new strings from Result Only arguments.

Remarks Use the Does String Contain Sub-String function to search a string for a specific character or sequence of characters. The function returns an index within the input string to the beginning character.

Make a Normalized Vector

Function Description Constructs a normalized vector from an input vector. The resultant normalized vector is accessible as a Result Only argument.

Return Description Returns success when the normalized vector is available as a Result Only argument. Failure is returned if the vector can not be normalized, e.g., the input vector has length 0.

Remarks Enter three double values or reference an input vector from a prior MP step. A normal vector has a length of 1. The square root of the sum of the component squares is also 1.

Analysis Operations...

Rename points based on proximity to reference points

Function Description Renames points based on their proximity to points in a reference group. Specify the reference group and the group of points to be renamed with Collection Object arguments. The radial proximity between points is set with a double value. Set a Boolean flag to control whether a verification dialog is presented to the user before the points are re-named.

Return Description Returns success after each of the point within the proximity tolerance is renamed. Failure is returned if the reference group or the group to be renamed is not found. When all of the points fail the proximity tolerance the step returns Failure.

Remarks Enter or reference the reference group and group to be renamed as Collection Object Names. The proximity tolerance is a radius about each point to be renamed. When more than one point from the reference group falls within the tolerance, the closest reference point is used to rename the point. Points that do not have a reference point the proximity tolerance are renamed with a prefix "Orig. " + point name (e.g., pt9 becomes Orig. pt9).

Rename points based on inter-point distance to reference points

Function Description Renames points based on inter-point distance comparison to a set of reference points. Specify the reference group and the group of points to be renamed with Collection Object arguments. The inter-point distance comparison tolerance between points is set with a double value. Set a Boolean flag to control whether a verification dialog is presented to the user before the points are re-

named.

Return Description	Returns success after each of the point within the comparison tolerance is renamed. Failure is returned if the reference group or the group to be renamed is not found. When all of the points fail the comparison tolerance the step returns Failure.
Remarks	Enter or reference the reference group and group to be renamed as Collection Object Names. The comparison tolerance is a difference between point to point distances for each point to be renamed. When more than one point from the reference group falls within the comparison tolerance, the smallest delta is used to rename the point. Points that do not have a reference point the proximity tolerance are renamed with a prefix "Orig. " + point name (e.g., pt9 becomes Orig. pt9).

Get Number of Points in Group

Function Description	Gets the number of points in a point group. The total number of points in the group is returned in an integer Result Only argument.
Return Description	Returns success after the number of points in the group is determined and set into the Result Only Total Count argument.
Remarks	This function assumes the point group is in the active collection. Use a reference to the returned Total Count integer to control looping operations on individual points within a group.

Get Number of Vectors in Vector Group

Function Description	Gets the number of Vectors within a Vector Group. The total number of vectors is returned in an integer Result Only argument. Set the Vector Group name with a Collection Object Name.
Return Description	Returns success after the number of vectors in a vector group is determined and set into the Result Only Total Count argument. Failure is returned if the vector group is not found.
Remarks	Configure the Vector Group Name argument by entering the collection and vector group directly or by referencing its Collection Object Name from another MP step. When the collection name is not specified, the active collection name is assumed. Use a reference to the returned Total Count integer to control looping operations on individual vectors within a vector group.

Get Number of characters in a string

Function Description	Gets the number of characters in a string. The total number of characters is returned in an integer Result Only argument. Reference or enter strings from other step arguments.
Return	Returns success after the number of characters in the string is determined and

Description	set into the Character Count argument.
Remarks	Use a reference to the returned Character Count integer to control looping operations on individual characters within the string.

Get Vector Group Properties

Function Description	Gets statistics properties of a Vector Group. Set the Vector Group name with a Collection Object Name argument. The vector group properties are returned as integer and double Result Only arguments. The returned properties include Total Vectors, Vectors in Tolerance and Vectors Out of Tolerance as integer arguments. Vector group properties returned as double arguments include % Vectors in Tolerance, % Vectors Out of Tolerance, Absolute Max Magnitude, Absolute Min Magnitude, Max Magnitude, Min Magnitude, Standard Deviation, Standard Deviation Mean Zero, Avg Magnitude, and Avg of Abs Magnitude.
Return Description	Success is returned when the vector group is found and the property Result Only arguments are each loaded with the values. Failure is returned if the vector group is not found.
	Configure the Vector Group Name argument by entering the collection and vector group directly or by referencing its Collection Object Name from another MP step. When the collection name is not specified, the active collection name is assumed.
Remarks	Use a references to the vector group properties to control process decisions concerning acceptable or out tolerance conditions.

Auto-Range and Set Vector Group Colorization (Selected)

Function Description	This function helps configure a series of vector groups to use common colorization options and auto-ranges the group's high and low color key parameters consistently. A flag controls whether the auto-ranging is set to range across all of the selected vector groups or set to auto-range each vector group individually. Configure the vector group colorization options for the selected vector groups. All of the vector groups will use these colorization options.
Return Description	Success is returned when the vector groups are auto-ranged and their colorization is configured. Failure is returned if the vector groups are not found or the colorization options are not possible. Partial Success is return when some not and all vector groups identified in the Collection Vector Group Name Ref List argument are not found.
	Setup the list of selected Vector Groups with a Collection Vector Group Name Ref List. This can be configured with editor provided by the argument or referenced from another step's argument list.
Remarks	When the "Treat Individually" flag is FALSE this function configures all the vector group ranges consistently. The max and min values are evaluated across all of the vector groups. Once the max and min are determined, the selected vector groups are configured to use them as the color key max and min values. When the "Treat Individually" flag is TRUE the max and min

properties are determined and the color key is set from each group.

Configure the colorization options to match the application requirements. Set the tolerances, Color Ranging Methods, Base Color, Whisker Appearance parameters, Draw Arrow Heads, Indicate Values, Vector Magnitude, Vector Width, Draw Color Blotches, Size, Show Out of Tolerance Only, Color Bar and Appearance parameters.

Auto-Range and Set Vector Group Colorization (All)

Function Description	This function helps configure all vector groups to use a common colorization option and auto-ranges their high and low color key tolerance parameters consistently. A flag controls whether the auto-ranging is set to range across all vector groups or auto-ranges the high and low color key for each vector group individually. Configure vector group colorization options argument. All vector groups currently in the job use these colorization options. New vector groups use the setting from User Options >> Colorized Object Options dialog.
Return Description	Success is returned when vector groups are auto-ranged and their colorization configured. Failure is returned if vector groups are not found or the colorization options are not possible for the vector groups.
Remarks	<p>When the “Treat Individually” flag is FALSE this function configures all the vector group ranges consistently. The max and min values are evaluated across all of the job’s vector groups. Once the max and min are determined, all vector groups color key max and min values are configured to use them. When the “Treat Individually” flag is TRUE the max and min properties are determined from each group.</p> <p>Configure the colorization options to match the application requirements. Set the Tolerances, Color Ranging Methods, Base Color, Whisker Appearance parameters, Draw Arrow Heads, Indicate Values, Vector Magnitude, Vector Width, Draw Color Blotches, Size, Show Out of Tolerance Only, Color Bar and Appearance parameters.</p>

Set Vector Group Colorization Options (Selected)

Function Description	This function helps configure a series of vector groups to use common colorization options. Configure the vector group colorization options for the selected vector groups. All of the vector groups will use these colorization options.
Return Description	Success is returned when the selected vector group colorization parameters are configured. Failure is returned if the vector groups are not found or the colorization options are not possible. Partial Success is return when some not and all vector groups identified in the Collection Vector Group Name Ref List argument are not found.
Remarks	<p>Setup the list of selected Vector Groups with a Collection Vector Group Name Ref List. This can be configured with editor provided by the argument or referenced from another step’s argument list.</p> <p>Configure the colorization options to match the application requirements. Set</p>

the tolerances, Color Ranging Methods, Base Color, Whisker Appearance parameters, Draw Arrow Heads, Indicate Values, Vector Magnitude, Vector Width, Draw Color Blotches, Size, Show Out of Tolerance Only, Color Bar and Appearance parameters.

Set Vector Group Colorization Options (All)

Function Description	This function helps configure all vector groups to use a common colorization options. Configure vector group colorization options argument. All vector groups currently in the job use these colorization options. New vector groups use the setting from User Options >> Colorized Object Options dialog.
Return Description	Success is returned when colorization properties for all the vector groups in the job are configured. Failure is returned if vector groups are not found or the colorization options are not possible for the vector groups.
Remarks	Configure the colorization options to match the application requirements. Set the Tolerances, Color Ranging Methods, Base Color, Whisker Appearance parameters, Draw Arrow Heads, Indicate Values, Vector Magnitude, Vector Width, Draw Color Blotches, Size, Show Out of Tolerance Only, Color Bar and Appearance parameters.

Get Point Coordinate

Function Description	Gets a point's coordinate as a vector and as individual component values. The coordinates are relative to the working frame. Specify the point name with Collection::Group::TargetName. Point coordinate vector representation, X, Y, and Z values are returned in Result Only arguments.
Return Description	Returns success after the point is found and the coordinates and vector are loaded into the Result Only arguments. Failure is returned if the point is not found.
Remarks	Load the point name by entering or referencing a Point Name argument from another MP step. The values for the coordinates are provided as a vector and as individual component values.

Get Point Coordinate (Cylindrical)

Function Description	Gets a point's coordinate and represents them in a cylindrical coordinate system. The coordinate system is relative to the working frame. Cylindrical coordinates are returned as Radius, Theta, and Z values Result Only arguments. Specify the point name with Collection::Group::TargetName.
Return Description	Returns success after the point is found and the coordinates are loaded into the Result Only arguments. Failure is returned if the point is not found.
Remarks	Load the point name by entering or referencing a Point Name argument from another MP step. The values for the coordinates are provided as cylindrical component values. Theta angle values in a cylindrical coordinate system are the counter-clockwise angle from the X-axis and the point's normal projection on the XY plane. Angular units for the Theta values are expressed in decimal

degrees.

The returned Theta angle for a point on the Z axis is 0.0.

Get Point Coordinate (Polar)

Function Description	Gets a point's coordinate and represents them in a Polar coordinate system. The coordinate system is relative to the working frame. Polar coordinates are returned as Radius, Theta, and Phi values Result Only arguments. Specify the point name with Collection::Group::TargetName.
Return Description	Returns success after the point is found and the coordinates are loaded into the Result Only arguments. Failure is returned if the point is not found.
	Load the point name by entering or referencing a Point Name argument from another MP step.
	The values for the coordinates are provided as polar component values. Theta angle values in a polar coordinate system are the counter-clockwise angle from the X-axis and the point's normal projection on the XY plane. Phi angle values in a polar coordinate system are the angle from the Z-axis down to a ray from the working frames origin passing through the point. A point on the XY plane has a Phi angle of 90 degrees.
	Angular units for Theta and Phi values are expressed in decimal degrees.
Remarks	The returned Theta angle for a point on the Z axis is 0.0. Points on the positive Z-axis have Phi values of 0.0 while points on the negative Z-axis have values of 180 degrees.

Get Point To Point Distance

Function Description	Computes the distance between two points. Distance and vector values are provided in Result Only arguments. The function takes inputs for two point names and returns the difference vector between the two points along with the X, Y, and Z components relative to the working frame. The distance is returned in a Magnitude argument. Specify the first and second point names with Collection::Group::TargetName arguments.
Return Description	Returns success after computing the point to point vector and settings the results in the argument fields. Failure is returned when one or both point names are not found.
	Load the first and second point names by entering or referencing a Point Name argument from other MP steps.
Remarks	Reference the computed Result Only values in subsequent steps. The returned vector and individual component values are relative to the working frame. The vector starts at the first point and goes to the second point. Magnitude values are always positive.

Get Point To Line Distance

Function Description	Computes the distance between a point and line object. Distance and vector values are provided in Result Only arguments. The function takes inputs for the point and line object names. It returns a normal projection difference vector between the point and line. The X, Y, and Z components of the difference vector are also provided in Result Only arguments. Vector and individual component values are relative to the working frame. The distance is returned in a Result Only Magnitude argument. Specify the point name with a Collection::Group::TargetName argument.
Return Description	Returns success after computing the point to line vector and settings the results in the argument fields. Failure is returned when the point or line is not found.
Remarks	<p>Load the point name by entering or referencing a Point Name argument from another MP step. Specify the line object name by entering or referencing it from another MP step.</p> <p>Reference the computed Result Only values in subsequent steps. The returned vector and individual component values are relative to the working frame. The vector starts at the point and goes to the projected point on the line. Magnitude values are always positive.</p>

Get i-th Point From Group

Function Description	<p>Get properties for a point within a group based on its index within the group. An index within a group is a 0 based integer. The last point in the group is accessed with 1 minus the total number of points.</p> <p>Specify the name of the group with a Collection Object Name (e.g., Collection::GroupName) argument. Provide the index for a point within the group with an integer argument.</p> <p>Access to the point properties is provided in a series of Results Only arguments. The arguments include Complete Point Name, Point Name Only and a Vector of the point coordinates relative to the working frame.</p>
Return Description	<p>Reference the Results Only arguments from subsequent steps to use the values.</p> <p>Returns success after indexing within the group to the point and loading its properties into the Results Only arguments. Failure is returned when the Collection Object Name is not found or the point index is not valid for the specified group.</p> <p>The Complete Point Name argument combines the Collection::Group::Target details into a Point Name variable. The Point Name argument only has the Target name field. X, Y and Z coordinates loaded into the Vector argument are relative to the current working frame.</p>
Remarks	This function is used frequently in loops where each point within a group is accessed by iterating from 0 to the number for total points in the group. When iterating through the group ensure that the last point index is the total number of

points minus 1 (e.g., counter is less than the number of points within a group).

Get i-th Vector From Vector Group

Get properties for a vector within a vector group based on its index within the vector group. An index within a vector group is a 0 based integer. The last vector in the vector group is accessed with 1 minus the total number of vectors.

Specify the name of the vector group with a Collection Object Name (e.g., Collection::VectorGroupName) argument. Provide the index for the vector within the vector group with an integer argument.

Access to the vector properties is provided in a series of Results Only arguments. Arguments for vector properties include Beginning and Ending points, Total Delta vector, i j k Unit vector in the working frame and the magnitude of the delta vector.

Function Description	Reference the Results Only arguments from subsequent steps to use the values.
----------------------	---

Return Description	Returns success after indexing within the vector group to the vector and loading its properties into the Results Only arguments. Failure is returned when the Collection Object Name is not found or the vector index is not valid for the specified vector group.
--------------------	--

The Vector Name argument has the name field for the vector. X, Y and Z coordinates loaded in the vector arguments are each relative to the current working frame.

The Begin point is as the name suggests is the origin point for the vector. The End point is the end that the vector passes through. A vector from the Total Delta contains the point to point dX, dY, and dZ components. An ijk unit vector of the Total Delta has a length one and it is parallel to the Total Delta vector. A double argument for the magnitude (e.g., length) of the Total Delta vector is also provided.

Remarks	This function is used frequently in loops where each vector within a vector group is accessed by iterating from 0 to the number for total vectors in the vector group. When iterating through the vector group ensure that the last vector index is the total number of vectors minus 1 (e.g., counter is less than the number of vectors within a vector group).
---------	---

Get Line Properties

Gets properties for a line. Specify the name of the line.

Access to the line properties is provided in a series of Results Only arguments. Arguments for line properties include Beginning and Ending points, Total Delta vector, i j k Unit vector in the working frame and the magnitude of the delta vector. A series of angles that the line forms against different axes when projected on to a based axis plane are provided in Results Only double arguments.

Function Description

	Reference the Results Only arguments from subsequent steps to use the values.
Return Description	<p>Returns success after finding the line and loading its properties into the Results Only arguments. Failure is returned when the Object Name is not found.</p> <p>X, Y and Z coordinates loaded in the line vector arguments are each relative to the current working frame.</p> <p>The Begin point is as the name suggests is the origin point for the line. The End point is the end that the line passes through. A Delta Component vector from the Begin and End points has the dX, dY, and dZ components. An ijk unit vector of the Delta Component has a length one and it is parallel to the Delta Component vector. A double argument for the magnitude (e.g., length) of the Delta Component vector is also provided.</p> <p>A series of angles that the line forms against component axis are in double Results Only arguments. The angles are expressed in decimal degrees.</p>
Remarks	<ul style="list-style-type: none"> • Angle from the X about the Z in the XY plane • Angle from the Y about the X in the YZ plane • Angle from the Z about the Y in the XZ plane

Get Sphere Properties

Gets properties for a sphere. Specify the name of the sphere.

Access to the sphere properties is provided in a series of Results Only arguments. An argument for sphere properties includes center coordinates as a vector. The coordinates in the vector are relative to the current working frame. The radius and diameter are provided in Results Only double arguments.

Function Description	Reference the Results Only arguments from subsequent steps to use the values.
Return Description	Returns success after finding the sphere and loading its properties into the Results Only arguments. Failure is returned when the Object Name is not found.
Remarks	

Get Circle Properties

Gets properties for a circle. Specify the name of the circle.

Access to the circle properties is provided in a series of Results Only arguments. An argument for circle properties includes center point coordinates and normal vector components as vectors. The components in the vectors are relative to the current working frame. The radius and diameter are provided in Results Only double arguments.

Function Description	Reference the Results Only arguments from subsequent steps to use the
----------------------	---

	values.
Return Description	Returns success after finding the circle and loading its properties into the Results Only arguments. Failure is returned when the Object Name is not found.
Remarks	

Get Cylinder Properties

	Gets properties for a cylinder. Specify the name of the cylinder.
	Access to the cylinder properties is provided in a series of Results Only arguments. Arguments for cylinder properties include Beginning and Ending points, Axis Direction vector in the working frame and the length of the axis direction vector. The radius and diameter are provided in Results Only double arguments.
Function Description	Reference the Results Only arguments from subsequent steps to use the values.
Return Description	Returns success after finding the cylinder and loading its properties into the Results Only arguments. Failure is returned when the Object Name is not found.
	X, Y and Z coordinates loaded in the cylinder's vector arguments are each relative to the current working frame.
Remarks	The Begin point is as the name suggests is the origin point for the cylinder. The End point is the end that the centerline passes through. An Axis Direction vector from the Begin and End points has the dX, dY, and dZ components. A double argument for the length (e.g., magnitude) of the Axis Direction vector is also provided.

Get Plane Properties

	Gets properties for a plane. Specify the name of the plane.
	Access to the plane properties is provided in a series of Results Only arguments. Arguments for plane properties include Normal Direction vector in the working frame and a point on the plane. The D Parameter is provided in a Results Only double argument.
Function Description	Reference the Results Only arguments from subsequent steps to use the values.
Return Description	Returns success after finding the plane and loading its properties into the Results Only arguments. Failure is returned when the Object Name is not found.
Remarks	X, Y and Z coordinates loaded in the plane's vector arguments are each relative to the current working frame.

The plane's normal direction is as the name suggests is the direction of the plane's normal vector. A point on the plane along with a the distance the plane is from the working frame origin are provided.

Set Default Colorization Options

Function Description	Sets the default colorization object for vector groups. Configure the tolerance and saturation limit values, vector magnification, whether the magnitudes are shown and other parameters.
Return Description	Returns success
Remarks	Once set the colorization parameters are used by vector groups created in the Measurement Plan. The colorization parameters set in the MP do not change the default SA colorizations.

Query Groups to Objects

Queries all the points in point groups against a list of objects. A vector group and/or projected point group are created by the query.

List the point groups in a Group Name Ref List argument. Identify the object(s) in an Object Name Ref List. Provide the name for the vector group with the Resulting Object Name argument. Set the projection and type of output options for the vector group analysis and reporting options in a Projection Options argument.

Specify tolerance values (as double variables) for RMS and Maximum deviations. When the tolerance values are 0.0 they are not used to set the return status of the step.

Access to statistic results from the query are provided in a series of Results Only arguments. Arguments for a series of statistics include RMS Deviation, Max Absolute Deviation, Average Deviation, and Standard Deviation.

Function Description	Reference the Results Only arguments from subsequent steps to use the values.
Return Description	Returns success after creating the vector group and/or projected point group based on the projection options settings. Failure is returned when the input argument objects are not found. When the tolerance condition tests are failed Partial Success is returned as the step status.
Remarks	Set up the Projection Option to create vector groups and point groups at the same time. Point and vector names are set by input points used in the query.

Transform Objects - Frame To Frame

Function Description	Moves objects by transforming them by the relative differences between two frames. Identify the objects to move with an Object Name Ref List. Name the source and destination frames. The frames must be in the active collection.
----------------------	--

Specify the number of steps to animate the move with. As an example a value of 2 shows the instrument at mid-point and then at the end point of the move.

Return Description Returns success after moving the referenced objects through the transform. Failure is returned when the source or destination frames are not found.

Remarks

Transform Objects by Delta (World Transform Operator)

Function Description Moves objects by transforming them by a World Transform Operator. Identify the objects to move with an Object Name Ref List. Enter or reference the World Transform Operator. Transform Objects by World Transform Operator includes a scale component application from the source transform.

Return Description Returns success after moving the referenced objects through the transform.

Reference an existing World Transform Operator or enter the 6 position and orientation components directly.

Remarks World Transform Operators are generally used as output arguments from best fit functions. They are applied to objects and instruments to move them from their current positions to the ones solved for in best fit operations.

Transform Objects by Delta (About Working Frame)

Moves objects by transforming them by dx, dy, dz, Roll, Pitch and Yaw components relative to the working frame.

Function Description Identify the objects to move with an Object Name Ref List. Enter or reference the position and rotation transform components.

Return Description Returns success after moving the referenced objects through the transform.

Remarks The transform is set by configuring the dx, dy, dz, dRx, dRy, and dRz components. The values are applied relative to the working frame.

Translate Objects by Delta

Moves objects by translating them by a vector relative to the working frame.

Function Description Identify the objects to move with an Object Name Ref List. Enter or reference the translation vector.

Return Description Returns success after translating the referenced objects by the vector.

Remarks The translation vector is set by configuring the dx, dy and dz components. The values are applied relative to the working frame.

Fit Geometry to Point Group

	<p>Constructs geometry by best fitting points from a point group. Pick the geometry type to construct from a list e.g., line, plane, circle, cylinder, cone, sphere. Identify the point group to fit the geometry from. The point group has to be in the active collection. Specify the name for the object being constructed.</p> <p>When a specific fit profile is needed enter the name of the profile in the Fit Profile Name argument. Set the Report Deviation flag to control whether the Fit Results dialog is shown to the user. When false the dialog is not shown.</p>
Function Description	Enter or reference a fit interface tolerance. The interface automatically uses this value to filter points into in and out of tolerance categories.
Return Description	<p>Returns success after constructing the geometry. Failure is returned when the point group is not found. Failure is also returned when there are not enough points in the group to construct the geometry.</p> <p>Fit profiles are configured and named on the Analysis tab within the User Options dialog. Use template files to setup a consistent set of geometry construction fit profiles.</p>
Remarks	When the point group contains a lot of points the Fit Results dialog can take a considerable amount of time to configure and present.

Best Fit Transformation - Group to Group

	<p>Solves for the transformation that minimizes the errors between two groups of points. A best fit solution is used to find the optimum transform. The transform is saved as a world transform operator in a Results Only argument. Reference the argument to apply it to objects and instrument stations.</p> <p>The points are matched by name. There must be at least three points with common names.</p> <p>A Show Interface flag controls whether the Points to Points Best Fit dialog is configured and presented to the user. Once up the user confirms the residual errors, changes individual component weights and selectively used or not use elements of the point sets.</p> <p>Specify tolerance values (as double variables) for RMS and Maximum deviations. When the tolerance values are 0.0 they are not used to set the return status of the step.</p> <p>Access to statistic results from the fit are provided in a series of Results Only arguments. Arguments for a series of statistics include RMS Deviation and Max Absolute Deviation.</p>
Function Description	Reference the Results Only arguments from subsequent steps to use the values.
Return	Success is returned after solving the transform and passing the tolerance tests. Failure is returned when the reference or corresponding group is not found. A

Description	<p>partial success state is returned when one but not both tolerance test are not passed.</p> <p>The transform is solved for by the function but not applied to objects or instruments. Use Transform Objects by Delta (World Transform Operator) and Transform Instrument by Delta to apply the results.</p>
Remarks	Points with saved weights from prior fits are recalled and used during the fit.

Best Fit Transformation - Group to Group (Scale Free)

Solves for the transformation that minimizes the errors between two groups of points. A best fit solution including the scale parameter is used to find the optimum transform. The transform is saved as a world transform operator in a Results Only argument. Reference the argument to apply it to objects and instrument stations.

The points are matched by name. There must be at least three points with common names.

A Show Interface flag controls whether the Points to Points Best Fit dialog is configured and presented to the user. Once up the user confirms the residual errors, changes individual component weights and selectively used or not use elements of the point sets. This function carries the scale in the output World Transform Operator.

Specify tolerance values (as double variables) for RMS and Maximum deviations. When the tolerance values are 0.0 they are not used to set the return status of the step.

Access to statistic results from the fit are provided in a series of Results Only arguments. Arguments for a series of statistics include RMS Deviation, Max Absolute Deviation, and the scale factor solved for by the optimization.

Function Description	Reference the Results Only arguments from subsequent steps to use the values.
Return Description	<p>Success is returned after solving the transform and passing the tolerance tests. Failure is returned when the reference or corresponding group is not found. A partial success state is returned when one but not both tolerance test are not passed.</p> <p>The transform is solved for by the function but not applied to objects or instruments. Use Transform Objects by Delta (World Transform Operator) and Transform Instrument by Delta to apply the results.</p>
Remarks	Points with saved weights from prior fits are recalled and used during the fit.

Compute Group to Group Orientation (Rx,Ry,Rz)

Function	Computes the rotational differences between two groups of points. Specify the names of Reference and Corresponding Groups. The groups need to be in the
----------	---

Description	active collection.
	The points used to compute the orientation between the groups are match by name.
	Access to the Rx, Ry, and Rz results from the fit are provided in a series of double Results Only arguments. Reference the Results Only arguments from subsequent steps to use the values.
Return Description	Returns success after solving for the rotational differences between the groups. Failure is returned when the point groups are not found.
Remarks	The rotational angles between the point sets are saved in the Results Only arguments as decimal degrees.

Temperature Compensate a group

Function Description	Creates a temperature scaled copy of an original point group. The group is scaled relative to a frame. The scale is computed by specifying the Material CTE, with initial and final temperatures.
Return Description	Success is returned after scaling the original point group. Failure is returned when the referenced objects can not found.
	The measurement plan step indicates Fahrenheit as the required units. Celsius values can also be used. The important aspect of the units is this function is that they are consistent. When the CTE value is specified in Fahrenheit then the initial and final temperature must be expressed in Fahrenheit.
Remarks	Example CTE values for Aluminum is 0.0000135 in/in/F [0.0000245 mm/mm/C]

Query Clouds to Surface

	Queries all the points in a set of clouds against a surface. A vector group and/or projected point group are created by the query.
	List the point clouds in an Object Name Ref List argument. Identify the surface with an Object Name argument. Provide the name for the vector group with the Resulting Object Name argument. Set the projection and type of output options for the vector group analysis and reporting options in a Projection Options argument.
	Set a Proximity value to limit the distance that cloud points are projected to the surface. Cloud points are not used when they are further than the proximity distance from the surface. A skip factor controls the number of cloud points that are skipped between points that are queried against the surface.
	Specify tolerance values (as double variables) for RMS and Maximum deviations. When the tolerance values are 0.0 they are not used to set the return status of the step.
Function Description	Access to statistic results from the query are provided in a series of Results Only arguments. Arguments for a series of statistics include RMS Deviation

and Max Absolute Deviation.

Reference the Results Only arguments from subsequent steps to use the values.

Return Description	Returns success after creating the vector group and/or projected point group based on the projection options settings. Failure is returned when the input argument objects are not found. When the tolerance condition tests are failed Partial Success is returned as the step status.
Remarks	Set up the Projection Option to create vector groups and point groups at the same time. Point and vector names are set by input points used in the query.

Relationship Operations...

Show/Hide Relationship Report

Function Description	Opens or Closes a collection's relationship report. Specify a collection with relationships. Set a Boolean argument as either TRUE or FALSE to control whether the relationship report is shown.
Return Description	Return success when the collection has a relationship to show or hide the report. Failure is returned when the collection does not exist or there are no relationships in it.
Remarks	The collection does not need to be the active collection to control the report's Show and Hide state.

Show/Hide Relationship Watch

Function Description	Opens or Closes a relationship watch window. Specify the relationship name with a collection object name. Set a Boolean argument as either TRUE or FALSE to control whether the relationship's watch window is shown.
Return Description	Return success when the collection has a relationship to show or hide its watch window. Failure is returned when the collection does not exist or there are no relationships in it.
Remarks	The collection does not need to be the active collection to control the relationship's Watch Window state.

Make Point to Point Relationship

Function Description	Make and name a point to point relationship. Identify the two points by configuring two Point Name objects.
Return Description	Returns success after making and configuring the relationship. Failure is returned when either point is not found.
Remarks	The relationship is named with collection object argument.

Make Frame to Frame Relationship

Function Description	Make and name a frame to frame relationship. Identify the two frames by configuring two Collection Object Name objects.
Return Description	Returns success after making and configuring the relationship. Failure is returned when either frame is not found.
Remarks	The relationship is named with collection object argument.

Make Points to Objects Relationship

Function Description	Make and name a relationship between points and objects. Identify the points by configuring a Point Name Ref List. The objects included in the relationship are selected with a Collection Object Ref List. Configure the projection properties between the points and objects with the Projection Options argument (e.g., target offset, extra material thickness, and projection directions.)
Return Description	Returns success after making and configuring the relationship. Failure is returned when points or objects are not found.
Remarks	The relationship is named with collection object argument.

Make Groups to Objects Relationship

Function Description	Make and name a relationship between point groups and objects. Identify the point groups and objects by configuring a Collection Object Ref List. Configure the projection properties between the points and objects with the Projection Options argument (e.g., target offset, extra material thickness, and projection directions.)
Return Description	Returns success after making and configuring the relationship. Failure is returned when groups or objects are not found.
Remarks	The relationship is named with collection object argument.

Get Relationship Weighting

	Gets a relationship's weight parameter value. Name the relationship with the Object Name argument.
Function Description	The relationship's weight is set into a Results Only argument. Access the weight values from subsequent steps by referencing it with a double argument.
Return Description	Returns success when the relationship's weight parameter is loaded into the double Results Only argument. Failure is returned when the relationship is not found.
Remarks	Relationship weights are used in relationship fits. The weight's value is respect to the other relationships included fits.

Delete Relationship

Function Description	Deletes a relationship. Configure the collection object name argument with the relationship name.
Return Description	Returns success after deleting the relationship. Failure is returned when the relationship name can not be found.
Remarks	

Do Relationship Fit

Solves for a relationship minimization transformation and moves a list of objects and instruments. The optimization is based on minimizing the distance components modeled in relationships in a collection.

Identify the name of the collection with the relationships with an Object Name argument. The list of moving objects is setup with an Object Name Ref List argument. A list of the instrument indexes to move during the fit are specified in the Instruments ID List. Note ... moving an instrument also moves its measured points and objects. As an example, point groups measured by an instrument do not need to be in the list of objects to move.

A Direct Search flag allows control over the type of optimization used to move the objects and instruments. Use the direct search optimization when there is significant difference between the initial and final positions of the points/instruments. Use the general run optimization method when the movements are relatively small. The primary difference is speed.

Configure the Fit Degree of Freedom Option argument. Set the options to limit axial movements relative to the working frame. The default option is to include all degrees of freedom in the solution.

Function Description	An option to open the fit dialog for the user is set with a Boolean flag argument. Set true to show the fit dialog or leave it false to prevent the user from interacting with the fit constraints.
Return Description	Returns success after completing the fit and moving the objects and instruments. Failure is returned when objects or instruments in the lists are not found.
Remarks	

Get General Relationship Statistics

Gets a relationship's statistics values. Name the relationship with the Object Name argument. The relationship must be in the active collection.

Function Description	The relationship's statistics Maximum and RMS deviations are set into a Results Only argument. Access the values from subsequent steps by referencing it with double arguments.
----------------------	---

Return Description	Returns success when the relationship's statistics parameters are loaded into the double Results Only argument. Failure is returned when the relationship is not found.
Remarks	Get a relationship's MAX and RMS deviation statistics for use in reporting or fitting operations.

Get Points to Objects Relationship Statistics

	Gets points to objects relationship's statistics values. Name the relationship with the Collection Object Name argument.
Function Description	The relationship's statistics Maximum and RMS deviations are set into a Results Only argument. Other returned arguments include the number of Candidate Points, Points Sampled, Points Rejected, Points Used, and Points Out of Tolerance. Access the values from subsequent steps by referencing it with Results Only double arguments.
Return Description	Returns success when the relationship's statistics parameters are loaded into the double Results Only argument. Failure is returned when the relationship is not found.
Remarks	This function gets the current statistics for the named relationship. The stats are returned in Results Only fields. The number of Candidate points, Points Sampled, Points Rejected, Points Used, and Points Out of Tolerance stats are returned in integer arguments. Double arguments for the Max and RMS statistics are returned for the relationships.

Relationship Attributes...

Set Relationship Weighting

Function Description	Function sets a relationships weight value with a double argument. Name the relationship with an Object Name argument. The relationship needs to be in the active collection.
Return Description	Returns success when the relationships weight property is set to the double value. Failure is returned if the relationship is not found.
Remarks	Name or reference the relationship name to set its current weight property with the double argument value.

Get Relationship Weighting

Function Description	Function returns a relationships current weighting value. The value is returned in a Results Only double argument. Name the relationship with an Object Name argument. The relationship needs to be in the active collection.
Return Description	Returns success when the relationships weight property is set into the Results Only argument. Failure is returned if the relationship is not found.
Remarks	Name or reference the relationship name to get its current weight property.

Set Relationship Sub Sampling Options

	Set a relationship sub sampling properties with this function. Name the relationship with an Object Name argument. The relationship has to be in the active collection.
Function Description	The sub sampling options can be configured at a specific number of points or by setting it to use e n-th point. Arguments for each option are available along with flags for whether they are active. Select only one of the sampling flags to be TRUE to control this property.
Return Description	Returns success when the sub sampling property is set for the relationship. Failure is returned if the relationship name is not found.
Remarks	<p>Relationships that contain a lot of points can take a long time to process fitting and reporting functions. In some cases all of the points are not needed for these operations. Sub sampling properties control the number of points used during the evaluation.</p> <p>There is more than one method to control sub-sampling. One method sets the maximum number of points evaluated in the relationship to a specific number, e.g., 200. When set to sample with a maximum number of points the relationship uses an evenly spaced set of points. The spacing increment is set to evenly gather the points over the entire set of points. When there are fewer points than the sampling number all of the points are used.</p> <p>Another sampling method is configured by using e n-th point, e.g., 20. For example when the relationship has 1000 points and the sub-sampling property is set to e 20th point the relationship is evaluated with 50 points.</p>

Relationship Attributes (Scalar Types)...

Set Relationship Outlier Rejection (Scalar Type)

Function Description	This function applies an Outlier Rejection object to a specific relationship. Define the Outlier Rejection object with the options in the available argument. An Outlier Rejection object can be referenced from other steps. They can be defined with other MP functions. Specifically the Make Outlier Rejection Options and Make Symmetric Outlier Rejection Options.
Return Description	Returns success when the relationship gets setup with the outlier properties. Failure is returned when the named relationship is not found.
Remarks	Outliers are samples that are beyond a specific threshold. The threshold depends on the application. Configure the outlier rejection threshold to the application requirements. By setting the outlier rejection threshold the relationship is able to ignore those components. Setup the outlier rejection criteria in this function's argument or reference another steps outlier rejection object.

Set Relationship Fit Constraints (Scalar Type)

Function	Set a relationships fit constraint property with this function. Name the relationship with a Collection Object Name and configure the constraint or
----------	---

Description	reference a Fit Constraint Object from another step.
Return Description	<p>Returns success when the constraint is configured for the relationship. Failure is returned if the named relationship is not found.</p> <p>Default fit constraints are set to +/- 0.0. In the default state fitting and analysis functions compare observations against the nominal. The default fit constraints mechanism can be changed so fitting and analysis operations compare to a high and low enveloped zone relative to the nominal.</p> <p>The zone does not have to be symmetric nor does the nominal have to be between the high and low thresholds. Deltas between the high and low fit constraints are not used to compute corrections when fitting. Vectors with deltas with magnitudes between the fit constraints are not created.</p> <p>As an example, both high and low thresholds can be positive. In this case fitting operations fit to a positive material condition, i.e., measurements are moved to end up on the plus side of the nominal. This effect is useful when fitting measurements of a casting before machining operations. It produces a configuration where a consistent amount of material is machined off of the casting to yield a nominal part.</p>
Remarks	The Fit Constraint Options (Scalar Type) object is set with two high and low double values. They do not need to be equal or opposite in sign.

Set Relationship Tolerance (Scalar Type)

Function Description	Set a relationships tolerance property to specific values. Name the relationship with a Collection Object Name argument. The tolerance values are used to when making vector groups from the relationship.
Return Description	<p>Returns success when the tolerance is configured for the relationship. Failure is returned if the named relationship is not found.</p> <p>Relationship tolerances are used when getting statistics on the relationships current state. The function "Get General Relationship Statistics" and when creating a vector group from the relationship use the tolerance property when evaluating the statistics. Configure this property either with the argument in this function or reference a tolerance options argument.</p> <p>Other MP functions are available to configure a relationship tolerance object. They include "Make Scalar Tolerance" and "Make Symmetric Scalar Tolerance" functions. Flags in the Scalar Tolerance Option object allow one or both of the tolerances to be turned off or not evaluated. For example, if the low tolerance's Use flag is false, it is not used to evaluate the statistics or reported in a vector group created from the relationship.</p>
Remarks	Tolerances do not have to be symmetric or on opposite sides of the nominal. Tolerances where both high and low are positive or negative are allowed.

Make Scalar Tolerance

Function	Makes a scalar tolerance object. It is configured and then returned in a Results
----------	--

Description	Only argument. This argument is typically referenced by an argument in “Set Relationship Tolerance (Scalar Type)” functions.
Return Description	Returns success when the Tolerance Option (Scalar Type) is created.
Remarks	The scalar tolerance object is able to reference doubles for both high and low tolerance values. Flags are provided that enable one or both of the tolerances to be turn off and not evaluated. Tolerances do not have to be symmetric or on opposite sides of the nominal. Tolerances where both high and low values are positive or negative are allowed.

Make Symmetric Scalar Tolerance

Function Description	Makes a symmetric scalar tolerance object. Symmetric means the high and low tolerances are the same value but opposite in sign. The object is configured and then returned in a Results Only argument. This argument is typically referenced by an argument in “Set Relationship Tolerance (Scalar Type)” functions.
Return Description	Returns success when the Tolerance Option (Scalar Type) is created.
Remarks	The scalar tolerance object uses one double value for both high and low tolerance values. The low tolerance is configured with the negative of the double argument.

Make Outlier Rejection Options

Function Description	This function is used to configure an Outlier Rejection object. Outlier Rejection object are applied to relationships by another step. Use the “Set Relationship Outlier Rejection (Scalar Type)” step to apply this Outlier Rejection object to a relationship. An Outlier Rejection object is configured with high and low double arguments.
Return Description	Returns success when the Outlier Rejection Object is created and made available in the Results Only argument.
Remarks	<p>Outliers are samples that are beyond a specific threshold. The threshold depends on the application. Configure the outlier rejection object thresholds (high and low) to application requirements.</p> <p>By setting the outlier rejection threshold a relationship is able to ignore those components outside of the rejection criteria. Setup the outlier rejection criteria in this function’s argument by entering or referencing the high and low double arguments.</p> <p>Flags are provided that enable one or both of the rejection criteria to be turn off and not evaluated. The high and low rejection criteria do not have to be symmetric or on opposite sides of the nominal. Rejection criteria where both high and low values are positive or negative are allowed.</p>

Make Symmetric Outlier Rejection Options

Function Description	This function is used to configure a Symmetric Outlier Rejection object. Symmetric means the high and low rejection criteria are the same value but opposite in sign. Outlier Rejection objects are applied to relationships by other steps. Use the “Set Relationship Outlier Rejection (Scalar Type)” step to apply this Outlier Rejection object to a relationship.
Return Description	Returns success when the Outlier Rejection Object is created and made available in the Results Only argument.
Remarks	<p>Outliers are samples that are beyond a specific threshold. The threshold depends on the application. Configure the outlier rejection object thresholds to application requirements. The outlier rejection object uses one double value for both high and low rejection criteria values. The low criteria is configured as the negative of the double argument.</p> <p>By setting the outlier rejection threshold a relationship is able to ignore those components outside of the rejection criteria. Setup the outlier rejection criteria in this function’s argument by entering or referencing the double argument.</p>

Cloud Filters...

Filter Clouds to Plane

Function Description	<p>Filters a point cloud(s) against a plane making points out of those inside the proximity tolerance. Those cloud points outside the proximity are not converted into points in the point group.</p> <p>Name the clouds in the Object Name Reference List argument. Name the plane with an Object Name argument. The output point group name is specified. A proximity tolerance is used to only make points that are within the proximity tolerance.</p>
Return Description	Returns success after the points are created in the point group. Failure is returned when the objects are not found.
Remarks	

Filter Clouds to Group

Function Description	<p>Filters a point cloud(s) against a reference point group. It makes points out of cloud points that are inside the proximity tolerance to any of the reference points. Those cloud points outside the proximity are not converted into points in the point group.</p> <p>Name the clouds in the Object Name Reference List argument. Name the reference Point Group with an Object Name argument. The output point group name is specified. A proximity tolerance is used to only make points that are within the proximity tolerance.</p> <p>Set the maximum number of points per reference point with the integer argument. This variable limits the number of points in the point group. Leave the value at 0 when an all of the cloud points inside the proximity are converted</p>
----------------------	---

into points.

Return Description Returns success after the points are created in the point group. Failure is returned when the objects are not found.

Remarks

Filter Clouds to Surface

Filters a point cloud(s) against a surface object. It makes points out of cloud points that are inside the proximity tolerance to the surface. Those cloud points outside the proximity are not converted into points in the point group.

Name the clouds in the Object Name Reference List argument. Name the surface with an Object Name argument. The output point group name is specified. A Low and High proximity tolerance are used to only make points that are within the proximity tolerance envelope.

Function Description A skip factor controls the number of cloud points that are skipped between points that are queried against the surface.

Return Description Returns success after the points are created in the point group. Failure is returned when the objects are not found.

Remarks

Filter Clouds to BSplines

Filters a point cloud(s) against a list of B-Spline objects. It makes points out of cloud points that are inside the proximity tolerance to the B-Spline. Those cloud points outside the proximity are not converted into points in the point group.

Function Description Name the clouds in the Object Name Ref List argument. Name the B-Splines with an Object Name Ref List argument. The output point group name is specified. Minimum and Maximum proximity tolerances are used to only make points that are within the proximity tolerance envelope.

Return Description Returns success after the points are created in the point group. Failure is returned when the objects are not found.

Remarks

Filter Clouds to Line Segment

Filters a point cloud(s) against a line segment defined by two points. It makes points out of cloud points that are inside the proximity tolerance to the line segment. Those cloud points outside the proximity are not converted into points in the point group.

Function Description Name the clouds in the Object Name Ref List argument. Name the two points with Point Name arguments. The output point group name is specified.

	Minimum and Maximum proximity tolerances are used to only make points that are within the proximity tolerance envelope.
Return Description	Returns success after the points are created in the point group. Failure is returned when the objects are not found.
Remarks	

Raster Scan Edge Inspection

	<p>Function computes the differences between cloud points and a surface edge. A list of BSplines are used to project the points too. They provide a reference comparison for the cloud points on the edge.</p> <p>Specify a prefix for a series of output resultant group objects. A tolerance is used to test the condition of each segment of the edge points.</p> <p>A Minimum Number of Points argument defines the number of points required on the edge. When the number of points that project to the edge is less than the minimum the segment is failed.</p> <p>When the Maximum Percentage of Bad Points is exceeded the edge test is failed.</p>
Function Description	A string Results Only argument provides an effective method for reporting the results to documents and users.
Return Description	Returns success when the tests are passed. Failure is returned when the objects are not found or the conditional tests all fail. Partial success is returned when the test is completed and one of the test conditions fails.
Remarks	

New Raster Scan Edge Inspection

	<p>Function computes the differences between cloud points and a surface edge. The edge is broken up into a series of segments. The segments are defined by the distance along the sequence of BSplines on the edge. A list of BSplines are used to project the points too. They provide a reference comparison for the cloud points on the edge.</p> <p>Specify a prefix for a series of output resultant group objects. A tolerance is used to test the condition of each segment of the edge points.</p> <p>A Minimum Number of Points argument defines the number of points required per segment. When the number of points that project to the segment is less than the minimum the segment is failed.</p> <p>When the Maximum Percentage of Bad Points is exceeded the edge test is failed.</p>
Function Description	A string Results Only argument provides an effective method for reporting the

results to documents and users.

Return
Description Returns success when the tests are passed. Failure is returned when the objects are not found or the conditional tests all fail. Partial success is returned when the test is completed and one of the test conditions fails.

Remarks

Mushroom Target Hole Inspection

Function analyzes a point cloud for a button target (sphere) nested in a hole. The process results in a center point of the hole on the part surface.

Configure the Group Name argument to identify the group with the sphere points. A double for the known sphere radius is used to constrain the sphere fit computation.

Setup the Object Name argument to select the plane on the part surface. The center of the sphere is automatically projected to the plane of the part surface. The projected point becomes the measured center of the hole on the part surface.

Specify the Point Name argument to name the computed Center of the Hole Point.

Function
Description Results Only arguments for the sphere fit RMS and Max errors enable subsequent step to reference them and check the results to flag potential errors in the sphere fit process.

Return
Description Returns success when the tests are passed. Failure is returned when the objects are not found or the conditional tests all fail. Partial success is returned when the test is completed and one of the test conditions fails.

Remarks

Sphere Axis Check

Function computes a sphere with a fixed radius from a point group. A center point is created and named with the Point Name argument.

Results Only arguments for the sphere fit RMS and Max errors enable subsequent steps to reference them and check the results to flag potential errors in the sphere fit process.

Specify a line object to define an axis direction.

Function
Description The minimum difference between the sphere center and the line is computed and posted in vector, component and magnitude Results Only arguments. Reference these results from subsequent steps to use the objects and values.

Return
Description Returns success when the sphere center point and difference vector is created. Failure is returned when the objects are not found.

Remarks	The objects in the function need to be in the active collection.
---------	--

Patch Normal Shift - Point

Function copies and shifts a point along a plane's normal direction by a specified distance. The distance to shift the point is referred to as an Additional Material Thickness.

Identify the point group with the points to make the plane from using the Group Name argument. A plane is computed from the points in the group. The normal direction is used as the direction to shift the point. The group with the plane points needs to be in the active collection.

Select the point to shift with the Point to Shift argument. The point is copied and name with the point name in the Resulting Point Name argument.

Function Description	A double argument called Additional Material Thickness to set the distance to shift the point along is provided.
----------------------	--

Return Description	Returns success after copying the point and shifting it by the Additional Material Thickness parameter along the planes normal direction. Failure is returned when the objects are not found or there are not enough points in the group to compute a plane.
--------------------	--

Remarks	The positive direction for the plane's normal is set to point towards the working frame.
---------	--

Patch Normal Shift - Hole / Pin

Function copies and shifts a group of points along a plane's normal direction by a specified distance. The distance to shift the point is referred to as an Additional Material Thickness.

Identify the point group with the points to make the plane from using the Group Name argument. A plane is computed from the points in the group. The normal direction is used as the direction to shift the point. The group with the plane points needs to be in the active collection.

Select the point group to shift with the Perimeter Points Group Name argument. The points are copied and named with the point name base in the Resulting Point Name argument.

Function Description	A double argument called Additional Material Thickness to set the distance to shift the point along is provided.
----------------------	--

Return Description	Returns success after copying the points and shifting them by the Additional Material Thickness parameter along the planes normal direction. Failure is returned when the objects are not found or there are not enough points in the plane group to compute a plane.
--------------------	---

Remarks	The positive direction for the plane's normal is set to point towards the working frame.
---------	--

Angle Between Two Planes

Computes the angle between two planes and compares it against a nominal angle. A tolerance is used to evaluate the condition of the computed angle.

Name the planes with the Plane A and B Object Name arguments. The planes must be in the active collection.

Specify a tolerance value (a double variable) for the angular tolerance. When the tolerance value is 0.0 it is not used to set the return status of the step.

Function
Description

The angle between the planes is set into a double Results Only argument.

Return
Description

Success is returned when the angle is computed and it passes the tolerance test. A return of failure is the result when the plane objects are not found or the tolerance test does not pass.

Remarks

The angle is reported in decimal degrees. Access the angle between the planes from subsequent steps by referencing it with a double argument.

Angle Between Line and Plane

Computes the angle between a line and a plane and compares it against a nominal angle. A tolerance is used to evaluate the condition of the computed angle.

Name the line and plane objects with the Selected Line and Selected Plane Object Name arguments. The objects must be in the active collection.

Specify a tolerance value (a double variable) for the angular tolerance. When the tolerance value is 0.0 it is not used to set the return status of the step.

Function
Description

The angle between the line and plane is set into a double Results Only argument.

Return
Description

Success is returned when the angle is computed and it passes the tolerance test. A return of failure is the result when the line or plane objects are not found or the tolerance test does not pass.

Remarks

The angle is reported in decimal degrees. Access the angle between the line and plane from subsequent steps by referencing it with a double argument.

Angle Between Two Lines

Computes the angle between two lines and compares it against a nominal angle. A tolerance is used to evaluate the condition of the computed angle.

Name the lines with the Line 1 and Line 2 Object Name arguments. The lines must be in the active collection.

Function
Description

Specify a tolerance value (a double variable) for the angular tolerance. When the tolerance value is 0.0 it is not used to set the return status of the step.

	The angle between the lines is set into a double Results Only argument.
Return Description	Success is returned when the angle is computed and it passes the tolerance test. A return of failure is the result when the line objects are not found or the tolerance test does not pass.
Remarks	The angle is reported in decimal degrees. Access the angle between the lines from subsequent steps by referencing it with a double argument.

Group To Surface Fit

	<p>Solves for the transformation that minimizes the errors between a group of points and a surface. A best fit solution is used to find the optimum transform. The transform is saved as a World Transform Operator in a Results Only argument. Reference the World Transform Operator argument to apply it to objects and instrument stations.</p> <p>Select the group and surface objects with the Group and Object Name arguments. The objects must be in the active collection. The fit automatically uses current target offsets when solving the best fit optimization.</p> <p>Set the Do Conventional Fit flag to control the type of fit used in the optimization. When false the fit does an exhaustive search for the optimum transform. This process can take longer to search the solution space before returning. It offers advantages when there possibly are local minimum on the path to the optimum solution.</p> <p>Specify tolerance values (as doubles) for RMS and Maximum deviations. When the tolerance values are 0.0 they are not used to set the return status of the step.</p> <p>Access to statistic results from the fit are provided in a series of Results Only arguments. Arguments for a series of statistics include RMS Deviation and Max Absolute Deviation.</p>
Function Description	Reference the Results Only arguments from subsequent steps to use the values.
Return Description	Success is returned after solving the transform and passing the tolerance tests. Failure is returned when the point group or surface objects are not found. A partial success state is returned when one but not both tolerance tests fail.
Remarks	The World Transform Operator is solved for by the function but not applied to objects or instruments. Use Transform Objects by Delta (World Transform Operator) and Transform Instrument by Delta to apply the results.

Scale Bar Check

Function Description	Function checks two points against a scale bar length. Compensation parameters for material CTE and temperatures enable the check to consider the expansion of the bar material in the check. A Go/No Go tolerance test is used to provide feedback on a scale bar acceptance as measured by an
----------------------	---

instrument.

Select ScaleBar points A and B with the two point name arguments.

Set the current temperature argument with the ambient temperature by either entering or referencing the value. The temperature argument must be provided in degrees on the Fahrenheit scale. Specify the calibrated bar length with the Length argument. Specify the length in the same units that the tolerance argument is set with to keep the compensation and evaluation consistent.

CTE is specified in PPM/F. As an example the CTE for Aluminum is 13.5 ppm/F.

The calibrated bar length is compensated for the difference between ambient temperature and 68 degrees F. The compensated length is provided in a double Results Only argument. Access the argument from subsequent steps by referencing it with a double variable.

The compensated length is compared against the actual distance between point A and point B. The delta is tested against the tolerance value. The result of the test is used to set the function's return status.

Return Description	Returns success when the difference between compensated scale bar length and the distance between the points is less than the tolerance value. Failure is returned when the point A or B is not found or the tolerance test is failed.
--------------------	--

Remarks

Reporting Operations...

Quick Report

Function Description	Produces a quick report of an item in the active collection and puts it to the specified collection. Define the item type as either a Relationship or Object.
----------------------	---

Return Description	Success after the quick report is created. Failure is returned when the object or relationship are not found.
--------------------	---

Remarks	The objects (e.g., Coordinates, Instruments, Frames, Lines, BSplines, Planes, Circles, Ellipses, Cylinders, Spheres, Cones, Paraboloids, Surfaces, Vector Groups, Point Clouds, and Fit/Query Report) values are reported relative to the current working frame. A relationship stats and results are posted to the quick report.
---------	---

Define Report Template

Function Description	Creates a report template for generating SA Reports. Specify the name of the report template along with the report's Title, Summary options, Graphical View Options, SA Object to report on, Report Output Options, and whether to generate the report when the template is created.
----------------------	--

Return	Success after the SA Report template is created. Failure is returned if the template can not be created. This may happen if the template name does not
--------	--

Description	use valid SA object names.
	There are a number of optional items within the template. The Summary information and graphical view objects within the template can be left in there default state.
Remarks	If the report name already exists in the SA Job or on the disk it is overwritten when the template is used.

Generate/Update Templated Options

Function	
Description	Specify an SA report template with function argument and it is activated.
Return	Success after the SA Report is created. Failure is returned if the report can not be created. This may happen if the report name does not use a valid SA object name.
Description	
Remarks	An existing SA Report is updated with current data if it already exists.

Make Report Graphical View Options

Function	Makes a Graphical View Options object. This object is used in the Define Report Template step. Controls the specific type of view (e.g., No View, Current View, or Callout View) to be shown within the report. This function sets up a Results Only argument with the configured Graphical View Options object.
Description	
Return	Success after the Graphical View Option object is setup. Failure is returned if the Graphical View object can not be created, e.g., a named callout is not found or if the view name does not use a valid SA object name.
Description	
Remarks	Used in the Define Report Template step. Reference the configured Graphical View Options object from the Define Report Template step.

Make Report Output Options

Function	Makes a Report Output Options object. This object is used in the Define Report Template step. Controls the specific type of report output (e.g., Embedded SA Report, Embedded SA Doc, Output to External PDF, Output to External RTF) created with the Report Template. This function sets up a Results Only argument for the configured Report Output Options object.
Description	
Return	Success after the Report Output Option object is setup. Failure is returned if the Report Output object can not be created, e.g., a named callout is not found or if the view name does not use a valid SA object name.
Description	
Remarks	The output file name is specified in this object. The output file name is only needed if the file is posted externally to SA. Used in the Define Report Template step by referencing the Results Only Report Output Options argument.

Generate Custom HTML Report

Function Description Opens an HTML template file and replaces keywords in the file with references to step argument data from within the MP script. After replacing the keywords in the file is saved it with an alternative name. A table relating the keywords to specific step arguments is used to make the association between the keywords in the HTML file and the data in the MP script.

Return Description Returns success after opening the HTML file, replacing the keywords with the referenced data, and saves the modified HTML file. Failure is returned when the path/filename to the HTML template file are not valid.

The keyword reference table associates a keyword to step argument data. The keyword can be any sequence of characters, e.g., "{X}", "{dx}", "http://xxx.xxx". This function searches the HTML template for any matches of the keyword and replaces them with the referenced step argument data. Data types include strings, doubles, integer, vectors, or Booleans. This function is also used to direct an HTML page to display a newly created image by replacing a path/filename within href statements.

When the keyword \$\$SYSTEMINFO is found in HTML files the function automatically adds a table of pertinent details to the file. An example is shown below:

SA Version	SA 2005.12.08
SA Filename	C:\Analyzer Data\Reports\MakeCustomHTMLofVectorGrouo.xit
Date & Time	Wednesday, December 14, 2005 20:18
Units	Inches
Reference Frame	A::WORLD

The another keyword example line is: \$\$OBJECT(A::TestVG)

The object TestVG in collection "A" called "TestVG" is automatically added to the file when the Generate Custom HTML Report function executes on the HTML Template file.

The HTML substitution process recognizes \n and \t, for new line and tab characters.

Remarks Use absolute or relative path and file name for the output HTML file.

Generate Standard HTML Report

Function Description A standard HTML report of the entities in the active collection is created and saved. Define the HTML path and filename and the number of significant digits for the report.

Return Description	Success after the file is created. Failure is returned when the file and path are not valid.
	The objects (e.g., Coordinates, Instruments, Frames, Lines, BSplines, Planes, Circles, Ellipses, Cylinders, Spheres, Cones, Paraboloids, Surfaces, Vector Groups, Point Clouds, Relationships, Point Group Comparison, and Fit/Query Report) values are reported relative to the current working frame.
Remarks	Use an absolute or relative path with a file name for the output HTML report file.

Capture Screen to File (BMP/JPG/PNG/GIF/TIFF)

Function Description	Captures and saves a screen shot to a file. The file format of the file is based on the file extension name. If the file extension is a BMP the file is saved as a BitMap.
Return Description	Success after the file is created. Failure is returned when the filename or path are not valid.
Remarks	Use an absolute or relative path with a file name for the file. The file extension of specifies the file format, e.g., .jpg means the file is saved as a jpeg file.

Save Current View (BMP/JPG/PNG/GIF/TIFF)

Function Description	Saves SA graphics view to a file.
Return Description	Success after the file is created. Failure is returned when the filename or path are not valid.
Remarks	Use an absolute or relative path with a file name for the file. The file extension of specifies the file format, e.g., .jpg means the file is saved as a jpeg file.

Create Chart from Vector Group

Function Description	Creates a chart from Vector Group data. Configure the chart name, type (e.g., Run Chart, Bulls-eye, and Individual X – Moving Range), and axes with step arguments.
Return Description	Returns success after the chart is created. Failure is returned when the Vector group is not found.
Remarks	Configure the chart with an existing chart as a template. The optional chart settings are used to set the newly created chart. An optional argument “Show Interface,” enables control of whether the chart is shown to the user.

Save Chart to JPeg file

Function Description	Creates and saves a jpg image of a chart to a file. Specify the chart name and the file name with its path.
----------------------	---

Return Description	Success after the jpg image file is created from the chart. Failure is returned when the chart or path/filename are not valid.
Remarks	The chart must be in the active collection. Use an absolute or relative path with a file name for the jpg file.

Notify User Integer

Function Description	Pops a dialog with a leading text message and an integer value for the user. This method stops the script and waits for the user to click the OK button on the dialog.
Return Description	Returns success
Remarks	Used frequently to debug scripts by referencing integer values from within the MP.

Notify User Double

Function Description	Pops a dialog with a leading text message and a double value for the user. This method stops the script and waits for the user to click the OK button on the dialog.
Return Description	Returns success
Remarks	Used frequently to debug scripts by referencing double values from within the MP.

Notify User String

Function Description	Pops a dialog with a leading text message and a string value for the user. This method stops the script and waits for the user to click the OK button on the dialog.
Return Description	Returns success
Remarks	Used frequently to debug scripts by referencing string values from within the MP.

Notify User Text Array

Function Description	Pops a dialog with a leading text message and a text area value for the user. This method stops script and waits for the user to click the OK button on the dialog.
Return Description	Returns success

Remarks	Used frequently to relate multi-line instructions to users.
---------	---

Notify User HTML

Function Description	Opens a browser dialog and loads an HTML file. This method stops the script and waits for the user to either click OK or Cancel buttons on the dialog.
Return Description	Return success when the OK button is clicked. Failure is returned when the Cancel button is clicked.
Remarks	Used to ask the user to accept or reject results with an HTML interface.

MS Office Reporting Operations...

Quick Report

Function Description	Generates a Quick Report from an Object or a Relationship. Identify the collection name where the object/relationship exists with the Collection Name argument. Completed reports are added to the database in the active collection. The report is opened automatically.
Return Description	Returns success after the report is created and added to the database. Failure is returned when the object or Relationship is not found.
Remarks	Quick Reports name are automatically set with the object or relationship name. A quick report overwrites existing quick reports with the same name.

Generate Template Report

Function Description	Builds and saves a report with an SA report template. Call a report template from the database by specifying its collection and name. The template is executed and the report is saved with the output parameters specified in the template.
Return Description	Return success after creating the template. Failure is returned when the template is not found in the database.
Remarks	Completely specify a report with an SA Report Template properties dialog. Once saved in the database call it with this MP step to create the report.

Initialize Office Report

Function Description	Initialize an OLE session between SA and MS Office Word document. Specify the file name and path number for the document. A report title is added to the document and its style setting is configured to use the document's Title Style. If a report template with headers/footers, background, margins, etc. is available specify it as the Report Template in the list of step arguments.
Return Description	Returns success when the office document is created and ready to receive input. Failure is returned when the path or filename are not legal path/file names.

MS Word and Excel version 2000 or greater are required to use this reporting method. The document must be initialized before posting data to it. After sending the data to the document, it must be closed with the Close Office Report function. Only one office report should be opened at a time.

Template baseline reports with company logos, backgrounds, margins etc. and call them with the template document option. Templates are opened and used as the basis for the report. By calling an existing report as a template additional data is appended to the document.

Remarks

Insert Section Break

Function Description Insert a Section Break into an open MS Word report. The break is added at the current cursor position in the document (typically at the end of the document).

Return Description Returns success after inserting the section break. Failure is returned if an OLE session to an MS Word document is not open.

Set section properties with other document properties functions e.g., Set Page Orientation.

Remarks An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before inserting a section break.

Insert Page Break

Function Description Insert a Page Break into an open MS Word report. The break is added at the current cursor position in the document (typically at the end of the document).

Return Description Returns success after inserting the page break. Failure is returned if an OLE session to an MS Word document is not open.

The page break function sets the break at the end of the document.

Remarks An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before inserting a page break.

Set Page Orientation

Function Description Set the page orientation for a MS Word report document. Set the orientation to either Landscape or Portrait with a Boolean switch. True configures the page orientation to Portrait and false sets it to Landscape. Page orientation is set for a document section.

Return Description Returns success after setting the page orientation. Failure is returned if an OLE session to an MS Word document is not open.

Remarks An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before setting the page orientation.

Add Section Heading to Report

Function Description	Inserts section heading into a MS Word report document. Enter the heading and set its respective heading type (e.g., Heading 1, Heading 2 ... Heading 4) in the document.
Return Description	Returns success after adding the heading and style. Failure is returned if an OLE session to an MS Word document is not open.
Remarks	Configure the document heading styles with a document template. An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before adding the section heading to the report.

Add Objects to Report

Function Description	Inserts reports of objects (e.g., Point Group, Frame, Vector Group, Relationship etc.) report into a MS Word report document.
Return Description	Returns success after adding the object reports. Failure is returned if an OLE session to an MS Word document is not open.
Remarks	Configure a list of objects to report into the (open) MS Word document. An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before adding the objects to the report.

Insert Graphics from file

Function Description	Add a graphics file (e.g., jpg, gif, bmp) to a MS Word report.
Return Description	Returns success after adding the file to the report MS Word document. The function returns failure if the office document is not open.
Remarks	<p>Set the caption for the image in the office document with a string argument. The caption is added below the image in the report. A path to the graphics file can be added by entering an absolute or relative path to the file. Methods for browsing or referencing a file name are also available. Set the relative percentage of the page width for the image in the report. Enter 100 to get the image to take the whole page width. Enter 50 to set the image width at half the page width.</p> <p>An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before adding the graphics file to the report.</p>

Add Graphics View to Report

Function Description	Grabs the SA graphics view and adds it to an open MS Word report document.
----------------------	--

Return Description	Returns success after adding the graphics view to the report MS Word document. The function returns failure if the office document is not open.
	Set the caption for the image in the office document with a string argument. The caption is added below the image in the report. Set the relative percentage of the page width for the image in the report. Enter 100 to get the image to take the whole page width. Enter 50 to set the image width at half the page width.
Remarks	An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before adding the graphics view to the report.

Add User Input Notes to Report

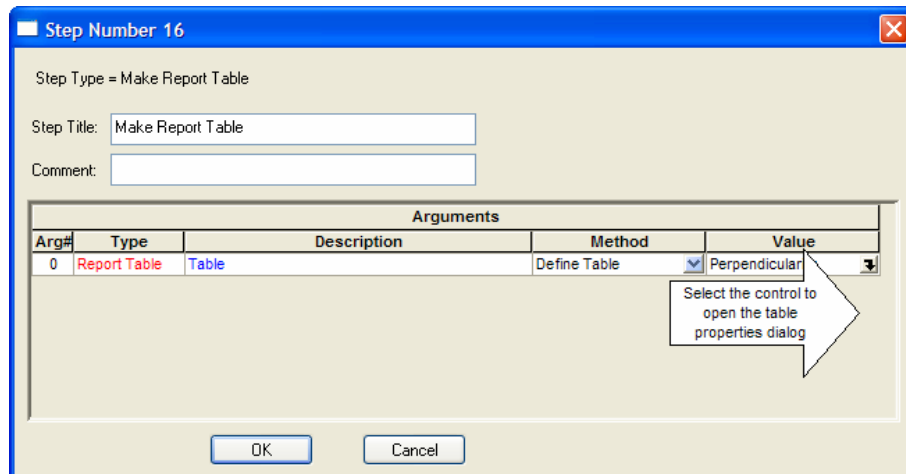
Function Description	A dialog is opened for the user to add in comments/notes for the MS Office report.
Return Description	Returns success after the user comments are added to the report. The function returns failure if the office document is not open.
Remarks	A simple open text dialog is opened where the operator adds text. The text is then posted to the open MS Word document. An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before adding the notes to the report.

Add Preset Notes to Report

Function Description	Adds text to a MS Word report document.
Return Description	Returns success after the user comments are added to the report. The function returns failure if the office document is not open.
Remarks	A simple open text dialog is opened where the text is posted to an open MS Word document. An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before adding the notes to the report.

Make Report Table

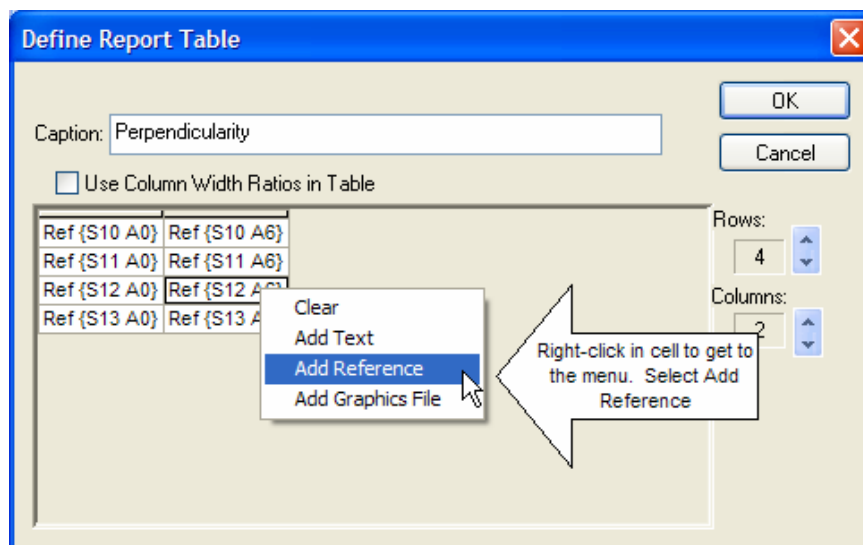
Function Description	Adds a table of data to an open MS Word document report. Configure a table with the interface in the function. The table can contain the primary MP data types by referencing and graphics files.
Return Description	Returns success after the table is added to the report. The function returns failure if the office document is not open.
	Programming a Report Table
Remarks	Select the control to open the table properties dialog (shown below)



Define the Report Table Caption

Set the number of rows and columns (shown below)

Right-click in each set to set it with one of the menu options (shown below)



An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) before adding the report table.

Close Office Report

Function Closes an open MS Office report session. This step is required before the MS Office document report can be accessed or printed.

Return
Description Returns success.

Remarks	Closes an open MS Office document session. After closing the document session it can be opened again where new report entries are appended to the document.
---------	---

Save Office Report as RTF

Function Description	Saves an MS Word document file in a Rich Text Format. Supply the path and filename for the rtf document.
Return Description	Returns success after the file is saved. Failure is returned if the file is not found.
Remarks	An open OLE session is required for this function. Initialize the document session (with the Initialize Office Report function) and add all of the objects to the report before saving the file as an RTF document.

Add SADoc From File (RTF)

Function Description	Adds an RTF document to the SA database as an SADoc. The file is added to the active collections.
Return Description	Returns success after the file is added to the SA database. Failure is returned if the file is not found.
Remarks	Use absolute or relative path with a file name for the RTF file to add to the SA database as a SADoc.

Process Flow Operations...

Jump To Step

Function Description	A goto statement that causes the script/program to unconditionally transfer control to the statement associated with the step specified in this function. Set the step to go to by value or by referencing a step.
Return Description	Returns success after jumping the script to the step identified in the argument. Failure is returned if the step does not exist.
Remarks	This function jumps to step 0 by default. Referencing a step ensures the jump goes to the selected step if the step numbers change.

Jump To Other Measurement Plan

Function Description	A goto statement that causes the script/program to unconditionally transfer control to a specified measurement plan and step number within the script. Set the measurement plan to open and start by entering a path, browsing to the file or by referencing an embedded measurement plan. Set the step to go to by value or by referencing a step.
Return Description	Returns success after jumping the script to the step identified in the argument. Failure is returned if the step doesn't exist.

Remarks	After transferring control to the other measurement plan the calling MP is closed. Use a subroutine to call another MP and control a return type and parameters.
---------	--

Step Status Test

Function Description	Checks the status of a measurement plan step. Specify steps to go too based on the steps return value. Jump to a different step if the step in question returns Success, Partial Success or Failure.
Return Description	Returns success
Remarks	Configure this step to control program flow. If the step in question is successful continue by jumping to the next sequence in the process. When the step does not return success the script can branch into error handling script sections.

Check Status Over Range of Steps

Function Description	Checks the status of a series measurement plan steps (e.g., steps 5 to 9). This function return success only if all the steps through the range of steps returned success. If any of the steps within the range returned partial success or failure this function returns failure.
Return Description	This function returns success only if all the steps through the range of steps returned success. If any of the steps within the range returned partial success or failure this function returns failure.
Remarks	Set the steps by entering the step number or by referencing the step. This function makes it simple to check a series of steps quickly. Check the return of this step with the Step Status Test function. Branch into error handling sections based on the return value.

Jump Based on Ranged Status Test

Function Description	Checks the status of a series measurement plan steps (e.g., steps 2 to 11). Configure the function arguments to jump to different sections of the measurement plan based on the return types. When all steps return success, partial success, or failure the script jumps to another section of the script. Specify the steps to jump too in the function's arguments.
Return Description	Returns success
Remarks	Configure this step to control program flow. If the steps in question all return success continue by jumping to the next sequence in the process. When the steps do not return success the script can branch into error handling script sections.

Go/No Go - Range Check Results

Function	This function sets a Boolean flag argument based on the return types of a range of measurement plan steps. When all the steps return success the
----------	--

Description	function branches to the argument set to define the step for continuing (e.g., Go). If any of the steps return failure this function branches to the step specified in the argument for NoGo. Configure text messages for Go or NoGo results.
Return Description	Return success
Remarks	The “Minor Error is Ok” flag argument allows control for how partial success returns are handled. When it is true a partial success branches to Go. A false Minor Error is Ok flag branch to NoGo when partial success return results are found. Check the results of the Boolean flag after this step executes to output the results.

Object Existence Test

Function Description	Checks to see if an object exists by name within the current collection. Configure the function to jump too a step if the object exists and another step when the object is not found.
Return Description	Returns success
Remarks	Used for error trapping and ensuring that objects exist before operating on them. Reference or enter the object name to check for and the steps to jump too.

Create Counter

Function Description	Creates an integer counter. Set the initial counter value with the argument. The default value is 0.
Return Description	Returns success
Remarks	Used in looping functions to index to within arrays objects (e.g., points within a Point Group or vectors within a Vector Group).

Increment Counter

Function Description	Adds one to a counter. Reference the counter value argument to increment.
Return Description	Returns success
Remarks	Used to move a counter (e.g., Index) up one.

Decrement Counter

Function	Subtracts one from a counter. Reference the counter value argument to
----------	---

Description	decrement.
Return Description	Returns success
Remarks	Used to move a counter (e.g., Index) down one.

Reset Counter

Function Description	Sets a counter to a specific integer value.
Return Description	Returns success. Failure is returned when a counter is not found to reset.
Remarks	Reference the counter to reset and either enter or reference the value too set the counter too.

Ask for String

Function Description	Asks for a string variable from the user. Specify the Question to Ask of the operator with the argument. A Results Only argument contains the string entered by the operator. A dialog with the question and a data entry field is presented to the user. The data entry field accepts string variable types.
Return Description	Returns success when the Ok button is clicked and failure is returned if the Cancel button is used.
	Function pauses the script and waits for the operator/user to enter a string and then use the enter key or click the Ok/Cancel buttons.
Remarks	Reference the Results Only argument to get access to the data entered by the user.

Ask for String (Pull-Down Version)

	Asks for a string variable from the user. A list of strings is setup for them to choose from. The choice is put in a Results Only string argument.
Function Description	The Question to Ask of the operator with the argument. A dialog with the question and a data entry field is presented to the user.
Return Description	Returns success when the Ok button is clicked and failure is returned if the Cancel button is used.
	Function pauses the script and waits for the operator/user to select a string from the pull-down list and then use the enter key or click the Ok/Cancel buttons.
Remarks	Reference the Results Only argument to get access to the selected string.

Ask for Point Name

	Asks for a Point Name variable (e.g., Collection::Group Name::Point Name) from the user. Specify the Question to Ask of the operator with the argument. Input fields for the Collection name, Group Name, and Point name are provided. A Results Only Point Name argument contains the point name as entered by the operator.
Function Description	A dialog with the question and a data entry fields for Collection, Group, and Point Name are presented to the user. The data entry field accepts string variable types.
Return Description	Returns success when the Ok button is clicked and failure is returned if the Cancel button is used.
	Function pauses the script and waits for the operator/user to enter a Collection name, Group name, and Point name variable and then use the enter key or click the Ok/Cancel buttons.
Remarks	Reference the Results Only argument to get access to the data entered by the user.

Ask for Integer

Function Description	Asks for an integer variable from the user. Specify the Question to Ask of the operator with the argument. A Results Only argument contains the integer entered by the operator. A dialog with the question and a data entry field is presented to the user. The data entry field only accepts integer variable types.
Return Description	Returns success when the Ok button is clicked and failure is returned if the Cancel button is used.
	Function pauses the script and waits for the operator/user to enter an integer and then use the enter key or click the Ok/Cancel buttons.
Remarks	Reference the Results Only argument to get access to the data entered by the user.

Ask for Double

Function Description	Asks for a double variable from the user. Specify the Question to Ask of the operator with the argument. A Results Only argument contains the double entered by the operator. A dialog with the question and a data entry field is presented to the user. The data entry field only accepts double variable types.
Return Description	Returns success when the Ok button is clicked and failure is returned if the Cancel button is used.
	Function pauses the script and waits for the operator/user to enter a double and then use the enter key or click the Ok/Cancel buttons.
Remarks	Reference the Results Only argument to get access to the data entered by the user.

Ask for User Decision

Function Description	Asks for a decision from the user by presenting a dialog with up to three configurable buttons. Specify the Question to Ask and then the labels on a series of buttons. Set up the step to jump too for each button by configuring STEP ID arguments. Based on which button is selected the script/program jumps to a different section.
Return Description	Returns success.
Remarks	Configure this step to let the user control program flow with a series of configurable buttons.

Ask for User Decision (HTML)

Function Description	Opens and displays an HTML file in a browser with up to three configurable buttons. Specify the labels on each button along with the step to jump too by configuring their respective STEP ID arguments. Based on which button is selected the script/program jumps to a different section.
Return Description	Return success
Remarks	Configure this step to let the user control program flow with a series of configurable buttons. Customize the HTML document with the Report Operations >> Generate a Custom HTML Report function and then open and use it with this function.

Ask for User Decision (Pull-Down Version)

Function Description	Opens and displays a dialog with a question. A list of responses in a pull-down menu lets the user select one. Each option in the list has a corresponding step id to go to when selected. After the user selects the option the steps goes to the step id referenced by the pull-down option.
Return Description	Return success when selected. If the user escapes out of the selection the step returns failure.
Remarks	Configure this step to let the user/operator select one of the choices from the pull-down list. The choices are configured in a list. The selected list item label is returned in a Results Only string argument. Reference the Results Only argument in subsequent steps to use as the operator's choice.

Ask for User Decision from Strings

Function Description	Opens and displays a dialog with a question. Up to three buttons with configurable labels are presented. The function sets a Results Only string argument with selected button label.
Return Description	Return success

Remarks	Configure this step to let the user/operator select one of three choices. The choices are configured as different button labels. The selected button label is returned in a Results Only string argument. Reference the Results Only argument in subsequent steps to use the operator's choice.
---------	---

Scalar Math Operations...

Integer Math Operation

Function Description	Function to add, subtract, multiply or divide two integers. Specify two integer inputs and an operation (+, -, *, or /) argument. An argument of the results of the operation is provided as an integer Result Only. Reference the resultant argument from subsequent steps to use it.
Return Description	Return success when the resultant value is available.
Remarks	Enter or reference the input integer variable and select the operation argument.

Double Math Operation

Function Description	Function to add, subtract, multiply or divide two doubles. Specify two double variable inputs and an operation (+, -, *, or /) argument. An argument of the results of the operation is provided as a double Result Only. Reference the resultant argument from subsequent steps to use it.
Return Description	Return success when the resultant value is available.
Remarks	Enter or reference the input double variables and select the operation argument.

Double Comparison

	Compare two double values and jump to different steps depending whether the comparison test is true or false. Comparison tests include:
	= Equal
	< Less Than
	> Greater Than
	<= Less Than or Equal
	>= Greater Than or Equal
	!= Not Equal
Function Description	Specify a step to jump to when the test is true and another step when the test is false.

Return Description	Returns success after the test is completed.
Remarks	Enter or reference the input double variables and select the comparison test operation argument. Referencing the steps to jump too helps by tracking the step index when new step are added or deleted from the MP.

Integer Comparison

Compare two integer values and jump to different steps depending whether the comparison test is true or false. Comparison tests include:

- = Equal
- < Less Than
- > Greater Than
- <= Less Than or Equal
- >= Greater Than or Equal
- != Not Equal

Function Description	Specify a step to jump to when the test is true and another step when the test is false.
Return Description	Returns success after the test is completed.
Remarks	Enter or reference the input integer variables and select the comparison test operation argument. Referencing the steps to jump too helps by tracking the step index when new step are added or deleted from the MP.

String Comparison

Compare two strings and jump to different steps depending whether the comparison test is true or false. Comparison tests include:

- = Equal
- < Less Than
- > Greater Than
- <= Less Than or Equal
- >= Greater Than or Equal

Function Description	!= Not Equal
	Specify a step to jump to when the test is true and another step when the test is

false.

Return
Description

Returns success after the test is completed.

Enter or reference the input string variables and select the comparison test operation argument. Referencing the steps to jump too helps by tracking the step index when new step are added or deleted from the MP.

Remarks

A string starting with "A" is Less Than one starting with a "B". It is also considered Not Equal. The string comparison is case sensitive.

Does String Contain Sub-String

Function
Description

Searches a "String to check" string for another a "Sub-string" string. When the sub-string found the function sets an integer Results Only field with the index of the first character in the source string.

Return
Description

Returns success after the test is completed.

Remarks

Boolean Comparison

Function
Description

Compare two Boolean variables and jump to different steps depending whether they are the same or different. Specify a step to jump to when they are the same and another step when the Boolean variables are different.

Return
Description

Returns success after the test is completed.

Remarks

Enter or reference the input Boolean variables and steps to jump too when the variable are the same or different.

Integer Absolute Value

Function
Description

Set a Results Only integer argument to the absolute value of an input integer variable. Reference or enter an input integer argument and then reference the Results Only argument from subsequent steps to use it.

Return
Description

Returns success after the Results Only integer argument is set as the absolute value of the input integer.

The absolute value of a number is always a positive number (or zero).

Examples:

$$|6| = 6$$

Remarks

$$|-12| = 12$$

$$|0| = 0$$

$$|1234| = 1234$$

$$|-1234| = 1234$$

Double Absolute Value

Function Description Set a Results Only double argument to the absolute value of an input double variable. Reference or enter an input double argument and then reference the Results Only argument from subsequent steps to use it.

Return Description Returns success after the Results Only double argument is set as the absolute value of the input double.

The absolute value of a number is always a positive number (or zero).

Examples:

$$|6.0112| = 6.0112$$

$$|-12.99| = 12.99$$

$$|0| = 0$$

$$|1.234| = 1.234$$

Remarks $|-123.4| = 123.4$

Double Square Root

Function Description Computes the square root from an input double variable. The resultant square root is put into a double Results Only argument. Reference or enter an input double argument and then reference the Results Only argument from subsequent steps to use it.

Return Description Returns success after the Results Only double argument is set as the square root of the input double.

Remarks The square root is the number which when multiplied by itself produces the input number. For example, 5 is the square root of 25, because $5 \times 5 = 25$.

Vector Math Operations...

Vector Addition

Function Description Function adds two vectors to produce a resultant vector. Specify the two input vectors by entering the vector components or by referencing vectors from other MP steps. An argument of the resultant vector is provided as a Results Only vector. Reference this resultant argument from subsequent steps to use it.

Return Description	Returns success after the Results Only vector argument is set as the sum of the two input vectors.
--------------------	--

Remarks	Vector addition is the process of combining two vectors. The algebraic addition of vectors involves adding up the like components (e.g., $X + X'$, $Y + Y'$ and $Z + Z'$) of the vectors.
---------	---

Vector Subtraction

Function Description	Function subtracts two vectors to produce a resultant vector. Specify the two input vectors by entering the vector components or by referencing vectors from other MP steps. An argument of the resultant vector is output as a Results Only argument. Reference this resultant argument from subsequent steps to use it.
----------------------	---

Return Description	Returns success after the Results Only vector argument is set as the subtraction of the two input vectors.
--------------------	--

Remarks	Vector subtraction is defined as the difference of two vectors, $A - B$, is a vector C that is, $C = A - B$ or $C = A + (-B)$. The algebraic subtraction of vectors is computed by subtracting like components (e.g., $X - X'$, $Y - Y'$ and $Z - Z'$) of the vectors.
---------	--

Vector Dot Product

Function Description	Computes the dot product from two input vectors. Specify the two input vectors by entering the vector components or by referencing vectors from other MP steps. An argument of the resultant vector is output as a Results Only double. Reference this resultant argument from subsequent steps to use it.
----------------------	--

Return Description	Returns success after the Results Only double argument is set as the dot product of the two input vectors.
--------------------	--

Remarks	The dot product of a and b , written $a \cdot b$, is defined as $a \cdot b = a b \cos \Theta$ where a and b are the magnitudes of a and b and Θ is the angle between the two vectors. When vectors a and b are unit vectors the dot product is the arc-cosine of the angle between the two vectors.
---------	---

Vector Cross Product

Function Description	Computes the cross product from two input vectors. Specify the two vector inputs by entering the vectors or by referencing them from other MP steps. An argument of the resultant cross product vector is output as a Results Only vector. Reference this resultant argument from subsequent steps to use it.
----------------------	---

Return Description	Returns success after the Results Only vector argument is set as the cross product of the two input vectors.
--------------------	--

Remarks	The cross product is a function that takes two vectors and produces a vector that is perpendicular to them. As an example the cross product a vector along the x-axis and one along the y axis is a vector parallel to the z-axis.
---------	--

Vector Scaling

Function Description	Scales an input vector by a factor and outputs the scaled vector in a Results Only vector argument. Specify the input vector by entering the components or by referencing a vector from another MP step argument.
Return Description	Returns success after the Results Only vector argument is scaled.
Remarks	Scaling a vector is done by multiplying each component by a scaling factor.

Vector Normalize

Function Description	Normalizes a general input vector into one with unit length. The resultant vector is output into a Results Only argument.
Return Description	Returns success after the Results Only vector argument is scaled.
Remarks	It is often useful to have a unit normal vector to simplify functions. The normalized vector is computed by dividing n by $ n = \sqrt{x^2 + y^2 + z^2}$. When $ n =1$, then the coefficients x, y, z of n are the direction cosines of the angles that the vector n makes with the xyz -axes.

Vector Magnitude (Length)

Function Description	Computes the magnitude (or length) of a vector. Specify the input vector by entering the components or by referencing a vector from another MP step argument. The magnitude is output into a Results Only double argument. Reference this resultant argument from subsequent steps to use it.
Return Description	Returns success after the magnitude of the vector is computed and saved into the Results Only argument.
Remarks	The length of the vector is computed by taking the square root of the sum of the squares of each vector component $ n = \sqrt{x^2 + y^2 + z^2}$.

Instrument Operations...

Get Last Instrument Index

Function Description	Gets the instrument index of last instrument in the active collection. The index is set into a Results Only integer argument.
Return Description	Returns success after setting the index value into the Results Only argument.
Remarks	Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Move Instrument to Another Collection

Function Description	Moves an instrument from the active collection to a destination collection. Specify the index of the instrument to move and the name of the destination collection.
Return Description	Returns success after moving the instrument to the destination collection. Failure is returned when an instrument index is not found.
	Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3. The instrument index in this function relates to the active collection.
Remarks	When the destination collection does not exist in the job the step creates a collection with the name and then moves the instrument to the collection.

Save Instrument Configuration

Function Description	Saves an instrument's configuration settings to a file. Specify the instrument's index from within the active collection and a path/filename for the configuration file.
Return Description	Returns success after the configuration file is saved. Failure is returned when the specified instrument index is not found.
Remarks	Configuration files save measurement profiles, targets, reflectors, offsets and other instrument specific settings. An active instrument interface is required to save the configuration file.

Load Instrument Configuration

Function Description	Loads a configuration file too an active instrument interface. Specify the instrument's index from within the active collection and a path/filename to the configuration file.
Return Description	Returns success after the configuration file is loaded. Failure is returned when the specified instrument index is not found.
Remarks	Configuration files load measurement profiles, targets, reflectors, offsets and other instrument specific settings. An active instrument interface is required to load the configuration file.

Point At Target

Function Description	Points an instrument at a target. Specify an existing point and an instrument index from within the active collection. A connected and active instrument's interface is required to point the instrument. For instruments with active control (e.g., Trackers) the instrument points along the ray defined from the instrument's origin and the point.
Return Description	Returns success after the target ID is found and the instrument is pointed at the target. Failure is returned when the specified point name or instrument index is

not found.

Configure the point name with the Target ID argument. Enter, reference or use the dialog entry form to specify the target name format e.g., Collection::Group::Target, Group::Target, or just the Target.

The instrument interface must be active for this function. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3. The instrument index in this function relates to the active collection.

An optional HTML file can be specific to provide instruction and guide the user to the specific target. The file is opened in a browser with Ok and Cancel buttons.

Remarks

Measure Single Point Here

Function Description Measures a single point from an instrument. Specify the point name and instrument index. The instrument interface must be active for this function. A "Measure Immediately" flag allows the function to trigger the measurement or set the interface up for the user/operator to initiate the measurement.

Return Description Returns success after the target is measured. Failure is returned when the instrument index is not found or the target measurement fails.

Configure the target name in the instrument interface with the Point Name argument. Enter, reference or use the dialog entry form to specify the target name format e.g., Collection::Group::Target, Group::Target, or just the Target.

Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3. The instrument index in this function relates to the active collection.

The status of the Measure Immediately flag has different effects depending on the instrument. Measurement systems with integrated triggers (e.g., hand held probes) let the user initiate the measurement. Systems setup with stationary targets (e.g., a tracker locked on an SMR) is triggered when the Measure Immediately flag is set to TRUE.

An optional HTML file can be specific to provide instruction and guide the user to measure the target. The file is opened in a browser with Ok and Cancel buttons.

Remarks

'Build' Target

Function Description Opens a build target window with real-time feedback on the target's position. Specify the instrument index, measured and nominal point names, and tolerances. The instrument interface must be active for this function.

Return Description Returns success after the target is measured. Failure is returned when the nominal point can not be found.

Remarks

The instrument must be in the active collection. Specify its index with the

Instrument ID argument.

Configure the output and nominal point names with the Target ID arguments. Enter, reference or use the dialog entry form to specify the target name format e.g., Collection::Group::Target, Group::Target, or just the Target name.

An optional HTML file can be specific to provide instruction and guide the user to build the target. The file is opened in the build target dialog.

Measure Existing Single Point

Function Description	Measures an existing single point from an instrument. If the instrument is capable or being driven, the function commands the instrument to point to at the existing point location. The instrument interface must be active for this function.
	Specify the existing point name, the group name for the measured point and instrument index. A "Measure Immediately" flag allows the function to trigger the measurement or set the interface up for the user/operator to initiate the measurement after it is driven to the existing point. A Results Only field returns the name of the measured target.
Return Description	Returns success after the target is measured and the resulting point name is posted to the Resulting Point Name argument. Failure is returned when the existing point name is not found, the instrument index is not found or the target measurement fails.
Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
	Configure the existing point name and a group name for the measured point. The instrument interface is configured with the target name from the existing point and group name for the measured point. Enter, reference or use the dialog entry form to specify the existing point name e.g., Collection::Group::Target, Group::Target, or just the Target name. After the target is measured the point's name is set into a Results Only argument. Reference this argument from subsequent steps to use it.
	The status of the Measure Immediately flag has different effects depending on the instrument. Measurement systems with integrated triggers (e.g., hand held probes) let the user initiate the measurement. Systems setup with stationary targets (e.g., a tracker locked on an SMR) are triggered when the Measure Immediately flag is set to true. When the flag is set to false the operator initiates the measurement.
	An optional HTML file can be specific to provide instruction and guide the user to measure the target. The file is opened in a browser with Ok and Cancel buttons.

Measure Existing Single Point (Manual Guide)

Function	Measures an existing single point from an instrument. The operator is
----------	---

SpatialAnalyzer User Manual **623**

Description	<p>expected to manually get the instrument ready to measure the point.</p> <p>Specify the existing point name, the group name for the measured point and instrument index. The instrument interface must be active for this function. A "Measure Immediately" flag allows the function to trigger the measurement or set the interface up for the user/operator to initiate the measurement. A Results Only field returns the name of the measured target.</p>
Return Description	<p>Returns success after the target is measured and the resulting point name is posted to the Resulting Point Name argument. Failure is returned when the existing point name is not found, the instrument index is not found or the target measurement fails.</p> <p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p> <p>Configure the existing point name and a group name for the measured point. The instrument interface is configured with the target name from the existing point and group name for the measured point. Enter, reference or use the dialog entry form to specify the existing point name e.g., Collection::Group::Target, Group::Target, or just the Target name. After the target is measured the point's name is set into a Results Only argument. Reference this argument from subsequent steps to use it.</p> <p>The status of the Measure Immediately flag has different effects depending on the instrument. Measurement systems with integrated triggers (e.g., hand held probes) let the user initiate the measurement. Systems setup with stationary targets (e.g., a tracker locked on an SMR) is triggered when the Measure Immediately flag is set to true. When the flag is set to false the operator initiates the measurement.</p>
Remarks	<p>An optional HTML file can be specific to provide instruction and guide the user to measure the target. The file is opened in a browser with Ok and Cancel buttons.</p>

Measure Existing Single Point and Compare

	<p>Measures an existing single point from an instrument. After measuring the target a point to point compare is computed between the existing and measured point. A tolerance test is available to flag measured points that are more than a specific amount from the existing point.</p> <p>If the instrument is capable or being driven, the function commands the instrument to point to at existing point location. The instrument interface must be active for this function.</p> <p>Specify the existing point name, the group name for the measured point and instrument index. A "Measure Immediately" flag allows the function to trigger the measurement or set the interface up for the user/operator to initiate the measurement after it is driven to the existing point.</p>
Function Description	<p>Results Only fields return the name of the measured target, the vector between the existing and measured points, delta X, delta Y, delta Z, and vector</p>

	magnitude.
Return Description	Returns success after the target is measured and is the magnitude of the difference between the existing and measured points is less than the tolerance. Partial success is returned if when the point is measured but fails the tolerance test. Failure is returned when the existing point name is not found, the instrument index is not found or the target measurement fails.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
	Configure the existing point name and a group name for the measured point. The instrument interface is configured with the target name from the existing point and group name for the measured point. Enter, reference or use the dialog entry form to specify the existing point name e.g., Collection::Group::Target, Group::Target, or just the Target name.
	The status of the Measure Immediately flag has different effects depending on the instrument. Measurement systems with integrated triggers (e.g., hand held probes) let the user initiate the measurement. Systems setup with stationary targets (e.g., a tracker locked on an SMR) are triggered when the Measure Immediately flag is set to true. When the flag is set to false the operator initiates the measurement.
	An optional HTML file can be specific to provide instruction and guide the user to measure the target. The file is opened in a browser with Ok and Cancel buttons.
	Specify the tolerance for the magnitude of the point to point difference between the existing and measured points. The test is ignored when the tolerance is 0.0.
Remarks	After the target is measured a vector and the component delta's between the existing and measured points is loaded and set into Result Only fields. Point's name is set into a Results Only argument. Reference these arguments from subsequent steps to use them.

Add New Instrument

Function Description	Adds a new instrument to the active collection. Select the instrument to add from a list. The instrument index within the active collection is returned in a Results Only argument.
Return Description	Return success.
Remarks	Reference the instrument's index in the Results Only argument from subsequent steps to ensure functions reference the added instrument.

Initiate Servo-Guide

Function	Starts an interactive build mode with a tolerance for a sequence of points. The
----------	---

Description	<p>instrument interface must be active for this function.</p> <p>The nominal points are setup in a Point Name Reference List. This mode actively pans and zooms SA's graphics screen to help guide an operator to series of nominal build points.</p> <p>The tolerance is portrayed in the graphics view as a red translucent sphere or cylinder when the instrument's probe tip is not within the tolerance zone. When the instruments probe tip is inside the tolerance zone the sphere or cylinder color changes to green.</p> <p>Based on the color of the tolerance zone the operator/user initiates/triggers the point measurements.</p> <p>Specify optional prefixes to be added to each build point group and target name.</p>
Return Description	<p>Return success after all the build points are measured. Failure is return if a point measurement fails or the tolerance test is not met on the difference between a nominal and measured point.</p> <p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p>
Remarks	<p>After a point is measured the next point in the Point Name Reference List is automatically loaded and the build process continues. This mode ends after the last point in the list is measured.</p>

Watch Point To Point

	<p>Opens a Watch Window on a reference point. The watch window provides real-time updates of the differences between a reference point and current probe position.</p> <p>The instrument interface must be active for this function. When this step executes the Watch Window is opened and the measurement process is started.</p> <p>Specify the instrument index, reference point name, tolerance, and measurement mode (i.e., profile Watch Update) for the interface.</p> <p>A "Pause MP Until Closed" flag allows the function to pause the MP until the watch window is closed. When the flag is false the Watch Window is opened and the measurement process started the MP continues letting it process additional steps.</p>
Function Description	
Return Description	<p>Return success after the build point is measured. Failure is return if the point measurement fails or the tolerance test is not met on the difference between the nominal and measured point.</p>
Remarks	<p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example</p>

the fourth instrument in a collection has an index of 3.

Watch Point To Objects

	<p>Opens a Watch Window between one or more objects and the current probe position. The watch window provides real-time updates of the differences between an object and current probe position. The object closest too the probe position is automatically selected. Configure the probe offset properties for the Watch Window with the Projection Options argument.</p> <p>The instrument interface must be active for this function. When this step executes the Watch Window is opened and the measurement process is started.</p> <p>Specify the instrument index, Collection Object Name Ref List, Projection Options, tolerance, and measurement mode (i.e., profile Watch Update) for the interface.</p> <p>A “Pause MP Until Closed” flag allows the function to pause the MP until the watch window is closed. When the flag is false the MP continues after the Watch Window is opened and the measurement process is started. This lets the MP process additional steps.</p>
Function Description	
Return Description	<p>Return success after the build point is measured. Failure is return if the point measurement fails or the tolerance test is not met on the difference between the closest object and measured point.</p>
	<p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p>
Remarks	<p>The Collection Object Name Ref List is constructed in the Collection Object Name Editor. The sequence of names in the list is how objects are input into functions that require a list of object names.</p>

Watch Closest Point

	<p>Opens a Watch Window on one or more groups of points. The Watch Window provides real-time updates of the differences between a nominal point and current probe position. The nominal point closest to the probe position is automatically selected and used for the point to point delta shown in the Watch Window.</p> <p>The instrument interface must be active for this function. When this step executes the Watch Window is opened and the measurement process is started.</p> <p>Specify the instrument index, Group Name Ref List, tolerance, and measurement mode (i.e., profile Watch Update) for the interface.</p>
Function Description	<p>A “Pause MP Until Closed” flag allows the function to pause the MP until the watch window is closed. When the flag is false the Watch Window is opened and the measurement process started the MP continues letting it process</p>

additional steps.

Return Description	Return success after the build point is measured. Failure is return if the point measurement fails or the tolerance test is not met on the difference between the nominal and measured point.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The Group Name Ref List is constructed in the Group Name List Editor. The sequence of names in the list is how objects are input into functions that require a list of group names.

Start Instrument Interface

Function Description	Starts an instrument interface based on the instrument type. Specify the instrument index within the active collection. A tracker starts a Laser Tracker interface while a portable CMM starts the Arm interface.
	Specify the index with the Instrument ID argument.
Return Description	Returns success after the interface is started. Failure is returned when the index is not found or the connection to the instrument is not successful.
Remarks	Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Stop Instrument Interface

Function Description	Stops an active instrument interface based on the instrument's index. Specify the instrument index within the active collection.
	Specify the index with the Instrument ID argument.
Return Description	Returns success after stopping the interface. Failure is returned when the index is not found.
Remarks	Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Verify Instrument Connection

Function Description	Function checks if an instrument index is connected. It returns in Boolean Results Only argument. The value is either TRUE or FALSE depending on the interface's status. Specify the index with the Instrument ID argument.
Return Description	Returns success after checking the interface status. Failure is returned when the index is not found.
Remarks	Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Configure and Measure

Function Description	Configures an interface to a specific measurement mode and target name then initiates the measurement. Specify the point name and instrument index. The instrument interface must be active for this function. A "Measure Immediately" flag allows the function to trigger the measurement or set the interface up for the user/operator to initiate the measurement.
Return Description	Returns success after the target is measured. Failure is returned when the instrument index is not found or the target measurement fails.
Remarks	<p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p> <p>Configure the target name for the measured point. Enter, reference or use the dialog entry form to specify the existing point name e.g., Collection::Group::Target, Group::Target, or just the Target name.</p> <p>The status of the Measure Immediately flag has different effects depending on the instrument. Measurement systems with integrated triggers (e.g., hand held probes) let the user initiate the measurement. Systems setup with stationary targets (e.g., a tracker locked on an SMR) are triggered when the Measure Immediately flag is set to true. When the flag is set to false the operator initiates the measurement.</p>

Measure

Function Description	This function is effectively pressing the measure button on the instrument interface. It initiates the measurement with the interface's current target name and measurement mode. Specify the instrument index. The instrument interface must be active for this function.
Return Description	Returns success after the target is measured. Failure is returned when the instrument index is not found or the target measurement fails.
Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Show/Hide Instrument Interface

Function Description	Shows or hides an instrument's interface based on the state of a Boolean flag. The flag is called "Minimize Interface?" When true the interface is minimized. A false state maximizes the interface.
Return Description	Returns success if the interface is up and running. Failure is returned if the index is not found in the active collection or the instrument interface is not active.
Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example

the fourth instrument in a collection has an index of 3.

Locate Instrument (Ref. Tie-In)

	<p>Opens a dialog to make locating an instrument station to a reference group of points easy for the operator/user. The dialog helps guide the user measuring the reference targets. As soon as the instrument is located (e.g., after the third point) the user can initiate auto-measurement for all the rest of the points.</p> <p>Specify the instrument index, reference group name, name of group of points to measure, tolerance and whether the points should be automatically measured.</p>
Function Description	<p>The measured points are put in the group identified by the "Actual Group Name" argument. Each measured target is named by the corresponding reference target name.</p>
Return Description	<p>Returns success after the instrument is located and the fit RMS is less than the tolerance. Failure is returned when any of the inputs (e.g., instrument index or reference group) are not found or the tolerance test fails.</p> <p>After the third target is measured the function automatically transforms the all the measurements to reference points. Immediate feedback on the dialog as to how well they fit is easy to use. After the third point; the transform and statistics are updated automatically for each target measurement.</p> <p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p> <p>The tolerance test is ignored when the tolerance is 0.0.</p> <p>An "Auto Survey" Boolean flag allows the function to automatically initiate the auto-survey. When true the instrument initiates the measurements without waiting to be guided to each point. For this to work the instrument's location and orientation is at least roughly known. This mechanism is frequently used during auto-measurement cycles to keep an instrument oriented to the references. Test with the tolerance to know if the references have changed beyond an acceptable level.</p>
Remarks	

Locate Instrument (Group to Surface Quick Align)

	<p>Locates an instrument to an object with a Quick Align to a group of surface points. The reference point group is setup with at least four (preferably six or more) points on a surface. After the points are measure the function locates the instrument to the points then automatically does a "Points to Surface" best fit and compensates for the target offset.</p>
Function Description	<p>Specify the instrument index, name for the measured point group, name of the group containing the surface points, name of the surface to best too, any other object to transform with the instrument, and RMS and Max fit tolerances.</p>
Return	<p>Return success after locating the instrument and passing the tolerance tests. Failure is returned if any of the inputs are not found or all the tolerance tests</p>

Description	<p>fail. Partial success is returned when at least one but not all tolerance tests fail.</p> <p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p> <p>The tolerance tests are ignored when the input values are 0.0.</p>
Remarks	Results Only arguments for the fit RMS and Max Error are provide after the fit is completed. Reference these arguments from subsequent step to use them.

Locate Instruments (USMN)

	<p>Locates instruments using a USMN orientation between the specified instruments and a reference point group. The instruments to be included in the orientation are specified by index within the active collection. An optional argument for a nominal point group is available. Specify the name of the composite group of points. This group of points is the optimal set of points based on all of the inputs into the solution. An option to show the USMN dialog, or only show it when the RMS or max tolerances are violated is provided. Another option to use the "Auto Reject Outliers and Resolve" feature is available.</p> <p>Set the flag named AutoReject Outliers and Resolve to true to have the solution automatically eliminate observations with rankings above 100%. The process will resolve until the instruments are located and all the observations rankings are below 100%. When this flay is false the solution is performed without eliminating outliers.</p> <p>The Show USMN Dialog has a number of options. It can be configured to never show, always show or only show when the fit tolerances tests fail. Configure this argument to match the application requirements.</p>
Function Description	Set the tolerance requirements for the Max Error and RMS error values.
Return Description	<p>Return success after solving the network, locating the instruments and passing the tolerance tests. Failure is returned if any of the inputs are not found or all the tolerance tests fail. Partial success is returned when at least one but not all tolerance tests fail.</p> <p>The instruments must be in the active collection. Specify their indexes with the Instrument ID List argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p> <p>The tolerance tests are ignored when the input values are 0.0. Options to show the USMN dialog, or only show it when the RMS or max tolerances are violated are provided. Another option to use the "Auto Reject Outliers and Resolve" feature is available.</p>
Remarks	Results Only arguments for the fit RMS and Max Error are provide after the fit is completed. Reference these arguments from subsequent step to use them.

Get Instrument Transform

Function Description	Gets an instrument's location and orientation. A Results Only frame argument is provided with the instrument's transform relative to an input reference frame. Specify an instrument index and the name of a reference frame.
Return Description	Returns success after setting the Results Only frame for the instrument's position relative to the reference frame.
Remarks	<p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p> <p>The Results Only frame can be decomposed into position and rotation vectors or down to individual x, y, z, roll, pitch, and yaw components. Use the functions in Construction Operations >> Other MP Types >> Deconstruct Frame section.</p>

Set Instrument Transform

Function Description	Sets an instrument's location and orientation by specifying a transform. The instrument's transform values are set relative to a reference frame. Name the reference frame and instrument's index.
Return Description	The transform is entered or referenced from other transform arguments within the MP.
Return Description	Returns success after setting the instrument location with the transform. Failure is returned when the index or reference frame are not found.
Remarks	<p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p> <p>As an example use World as the reference frame and set the transform values to the first point in the reference group. The instrument's base frame is moved to the first reference point.</p> <p>Specify the number of steps to animate the move with. As an example a value of 2 shows the instrument at mid-point and then at the end point of the move.</p>

Get Instrument Weather Setting

Function Description	Gets an instrument's current weather settings. The settings are returned as double values in Results Only arguments. Values for temperature, pressure, and humidity are available in the Results Only arguments. The function also provides a Results Only Boolean flag indicating whether the values were read from an integrated weather station or manually set.
Return Description	Returns success after getting the weather values from the interface. Failure is returned if the instrument index is not found in the active collection or the interface is not active.
Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example

the fourth instrument in a collection has an index of 3.

The instrument interface must be active to get the weather values from the interface.

The weather value units are Fahrenheit, mmHg, and percentage of relative humidity.

Set Instrument Weather Setting

Function Description	Sets an instrument's weather settings. The settings are sent as double values to the interface.
Return Description	Returns success after setting the weather values in the interface. Failure is returned if the instrument index is not found in the active collection or the interface is not active. The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3. The instrument interface must be active to set the weather values in the interface.
Remarks	The weather value units are Fahrenheit, mmHg, and percentage of relative humidity.

Compute CTE Scale Factor

Function Description	Computes a scale factor based on a material CTE, initial and final temperatures. The scale factor is available as a double value in a Results Only argument.
Return Description	Returns success after computing the scale factor and making it available in the double Results Only argument.

The measurement plan step indicates Fahrenheit as the required units. Celsius values can also be used. The important aspect of the units is this function is that they are consistent. When the CTE value is specified in Fahrenheit then the initial and final temperature must be expressed in Fahrenheit.

Reference the computed scale factor argument from subsequent steps to use it.

Example: Aluminum

Remarks	CTE value	0.0000135 in/in/F	0.0000245 mm/mm/C
	Initial Temperature	85 F	29.4 C
	Final Temperature	68 F	20 C
	Scale Factor	0.99977905	0.99976975

Set (multiply) Instrument Scale Factor (CAUTION!)

Function Description Multiply a scale factor to an instrument's current scale factor. Specify the instrument's index (from within the active collection) and the scale factor to multiply to the instrument's existing scale factor. The resultant scale factor is the product of the two (i.e., current sf * sf = new sf).

Return Description Returns success after the instrument is scaled. Failure is returned when the instrument index is not found.

The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Remarks The new scale factor will be a product of the scale factor argument and the instrument's current scale factor. For example: the scale factor was 0.9995 and the scale factor argument is 0.9991. The resultant scale factor is 0.9986.

Set (absolute) Instrument Scale Factor (CAUTION!)

Function Description Sets a scale factor to an instrument's current scale factor property. Specify the instrument's index (from within the active collection) and the scale factor to set the instrument's scale factor. Changes to the scale factor cause the instruments targets to be re-computed.

Return Description Returns success after the instrument's scale is set. Failure is returned when the instrument index is not found.

The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Remarks The new scale factor is the instrument's new scale factor.

Get Instrument Scale Factor

Function Description	Gets an instrument's scale factor and returns it in a double Results Only argument. Specify the instrument's index (from within the active collection) and the scale factor is returned in the Results Only argument.
Return Description	Returns success after the instrument's scale is returned. Failure is returned when the instrument index is not found.
Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Transform Instrument - Frame To Frame

	Move an instrument from its current position by the relative delta between two frames.
Function Description	Specify an instrument index, initial and final frame names. The Number of Step argument controls how many intermediate positions the instrument is shown in the graphics view. Use this value to control movement animation.
Return Description	Returns success after moving the instrument with the delta transform. Failure is returned when the instrument index, initial or final frame names are not found.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
	As an example use "World" as an initial frame and a "Part" frame as the destination frame. When applied to an instrument; it is moved from where it is currently to a position/orientation the difference between World and Part.
Remarks	Specify the number of steps to animate the move with. As an example a value of 2 shows the instrument at mid-point and then at the end point of the move.

Transform Instrument by Delta

	Move an instrument from its current position with a World Transform Operator. A World Transform Operator is an output argument from Best Fit Transformation solutions. They portray a delta transform using the World frame as a base. The advantage of this operator is that it can be applied to move objects consistently. Function applies the scale from the World Transform Operator.
Function Description	Specify an instrument index and a World Transform Operator. The World Transform Operator is typically referenced from a Best Fit Transformation function.
Return Description	Returns success after moving the instrument with the World Transform Operator. Failure is returned when the instrument index or transform operator are not found.

Remarks The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3. The scale factor from the World Transform Operator is applied to the instrument.

Instrument Operational Check

Initiate an instrument operational check in the interface. Specify the instrument index and a string for the operational check. Not all Operational Checks apply to e instrument type.

Function After initiating the Operational Check the step waits until it is finished or closed
Description before continuing.

Return Returns success after completing the operational check. Failure is returned
Description when the index is not found or the operational check is not recognized by the interface.

The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

The instrument interface must be active to for it to perform the operational check.

The Command String argument does not need quotation marks around it in the step.

Laser Tracker "Instrument Operational Check" Command Strings Table

Command String	Description
"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 []
"Motors On"	Turn motors on
"Motors Off"	Turn motors off
"Start"	Re-start tracker
"Home"	Home tracker
"Two Face Here"	Perform Two Face ops check at current location
"ADM/IFM Here"	Perform ADM ops check (w.r.t. IFM) at current location
Remarks "IFM"	Start IFM ops check/cal. (two station or scale bar) at

	current location
"Closure"	Perform Closure check at current location w.r.t. current home
"Drift"	Start Drift ops check
"ADM"	Start ADM ops check/cal. UI for multiple locations (with respect to IFM)
"ADM Drive"	Equal to pressing the 'ADM Drive' button
"ADM Reset"	Equal to pressing the 'ADM Reset' button
"Ball Bar"	Start Ball Bar ops check
"Reflector Center"	Calls the Reflector Center dialog

Fabricate Observations

Function Description	Creates simulated measurements from an instrument to each point in a group of points. A Boolean flag called "Introduce Instrument Error" controls whether random error is injected on the measurements.
Return Description	Returns success after creating the simulated measurements. Failure is returned when the instrument index is not found.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	Instrument uncertainty estimates are used to bound the amount of random error introduced to each component (e.g., Hz, EI, and R) of measurement.

Set Instrument Measurement Mode/Profile

Function Description	Sets an instrument interface into a specific measurement mode profile. Specify the instrument index and the name of the measurement mode/profile. Custom measurement modes are support by calling them by name.
Return Description	Returns success after changing the measurement profile. Failure is returned when the instrument index is not found or the interface is not active.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
	The instrument interface must be active to for this function to set the measurement mode/profile.
Remarks	The measurement mode/profile string does not require quotation marks. Call custom user-defined measurement profiles by name.

Example Laser Tracker “Measurement Mode/Profiles” Table

Mode/Profiles	Description
Watch Update	Temporal scan sending updates to SA
Single Pt. To SA	Discrete point measurement sent to SA
Stable Pt. To SA	Stable point measurement sent to SA
Spatial Scan	Spatial scan with beam break termination sending points to SA
etc.	

Example Portable CMM “Measurement Mode/Profiles” Table

Mode/Profiles	Description
Discrete Points	Discrete point measurement sent to SA
Stream Points	Spatial scan sending points to SA
Patch (Projected Pt.)	Measure a patch on a surface then project a point to the surface
Pin	Measure a plane then at least 3 points on the pin to construct a circle/pin on the plane.
etc ...	

Example Portable Laser Radar “Measurement Mode/Profiles” Table

Mode/Profiles	Description
Tooling Ball	Tooling ball measurement mode
Surface Point	Measures a surface point, the Enhanced measurement mode is used with a timeout of 300 seconds
Surf-Vector Intersection	Raw SVI Measurement mode, projects to the nominal vector
mirroron <mirror name> mirroroff	Commands the interface to shoot through a mirror transform. Call the mirror by name. Applies the mirror transform to each measurement made after calling it. Turn the mirror transform mapping off with the mirroroff command.
etc ...	

Instrument interfaces have a default set measurement modes/profiles and the

customized profiles. Set the instrument interface into a measurement mode/profile using the name of the profile.

Set Instrument Group and Target

Function Description	Sets an instrument interface's point naming parameters. Specify the instrument index and the complete Collection::Group::Target name.
Return Description	Returns success after changing the Collection::Group::Target name is set in the interface. Failure is returned when the instrument index is not found or the interface is not active.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
	The instrument interface must be active to for it set the point name.
Remarks	Directly enter, use the dialog entry form for a target name or reference the point name argument. The active collection is used by default when the collection name is not included.

Set Instrument Targeting

Function Description	Sets an instrument interface's point naming parameters. Specify the instrument index and the Target name.
Return Description	Returns success after changing the target name is set in the interface. Failure is returned when the instrument index is not found or the interface is not active.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
	The instrument interface must be active to for it set the target name.
Remarks	Directly enter or reference the string with the target name.

Scan within perimeter

Function Description	Commands an instrument interface to collect a group of points within a perimeter. Use or specify a parameter set name to control the sampling within the perimeter.
Return Description	
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to scan within the perimeter.

Edge Scan Measurement

Commands an instrument interface to scan for an edge of a part. The instrument measures a series of points across the part edge and then computes (e.g., interpolates) a point on the edge of the part.

Provide two seed points to start the edge find. One point near the edge and one in the direction toward the edge. The scanner starts at the point near the edge and scans along a vector between the two points. Specify a parameter set name to control the sampling for the edge (e.g., how close and type of measurement to use).

Function
Description

Specify the Group and Target name for the resultant edge point.

Return
Description

The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Remarks

The instrument interface must be active to for it to scan for the edge feature.

Track Tape Measurement

Commands an instrument interface to scan for a series of retro-reflective targets laid out across a part surface. The instrument effectively tracks along the tape and finds each retro-reflective target. A sequentially named group of measured points are the output from this measurement process.

Provide an initial point on the tape to start the search for the initial retro-reflective target. A point on the part is used to orient the scan for the retro-reflective target. A point in the general direction of the next point is used to direct the scan along a vector from the initial point to search for the next target. Use a point beyond the last retro-reflective target to stop the search. The measurement mode is triggered to stop when the measurement ray moves past the termination point.

Specify a parameter set name to control the sampling for the retro-reflective tape targets (e.g., how close and type of measurement to use).

Function
Description

Specify the Group and an initial Target name for the tape targets. The point name is incremented as new tape targets are measured.

Return
Description

The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Remarks

The instrument interface must be active to for it to scan for the tape targets.

Auto Measure Points

Function Description	Command an instrument to automatically measure a group of nominal points. A group of measured points with the target names of their nominal points are output from this function.
	Each nominal point is used as a seed point for measurement. The instrument interface sets its target name with an "Actual Group Name" and the nominal point's target name. The instrument is driven to the nominal point coordinates. A target search is started. After the target is located and measured the actual point is sent to SA. The process is repeated for each point in the nominal group.
Return Description	
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to drive to the nominal points.

Auto-Measure Vectors

Function Description	Command an instrument to automatically measure a nominal (i.e., drive) vector group. A vector group of the differences along the drive vector between the nominal and measured points is output from this function. Each measured vector is named from the nominal drive vectors.
	The begin point on each drive vector is used as a seed point for measurement. The instrument interface sets its target name with an "Actuals Group Name" and the nominal vector's name. The instrument is driven to the initial seed point coordinates. A target search is started. After the target is measured it is projected to the nominal drive vector.
	When the Boolean flag "Project Point to Vector" is set to true the new projected point on the nominal drive vector is then used as the point to measure. The process is repeated until the measured point is on the nominal drive vector.
	A vector between the nominal point and the measured point is created in SA. The process is repeated for each nominal drive vector.
	An angle tolerance argument is used to skip vectors that project away from the instrument's line of sight. Skipping vectors that can not be measured saves time and reduces measurement error.
Function Description	High and low tolerance arguments are used to set the colorization of the measured vector group.
Return Description	
Remarks	Control the resolution of the measurement process with specific vector group

measurement mode/profiles within the instrument interface.

The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

The instrument interface must be active to for it to measure the points and drive along the nominal vector.

Auto-Correspond Closest Point

Function Description	A measurement mode that automatically names measured points based on which reference point it is closest too. Using a reference group of points, the function searches for the one that is closest too each point as it is measured. Once found, the target name from the reference point is used with the “Actuals Group Name (to be measured)” to name the measured point.
Return Description	
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to measure the points.

Drift Check

	Determine how much an instrument has moved relative to the object using this function. The function opens an easy to use dialog to measure a group of points and compares the point to point comparison results against a tolerance.
Function Description	Specify the instrument index, group name of the reference group, a name for the group of measured points and a maximum delta tolerance for the point to point comparison.
	Return success after the points are measured and the tolerance test is passed. Failure is returned when the instrument index is not found, the interface is not active or the tolerance test is failed.
Return Description	Partial success is returned when all the points are not measured.
	The dialog makes it easy to select the point to check and when setup for auto-measuring targets the process can be completed with the Auto-measure button.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to make the point measurements.

Measure Nominal Feature

Function Description	Measures a feature (e.g., circle/hole) using an instrument interface. Automatically drives the instrument to the feature and initiate measurements. A measured feature is created. Specify the instrument index and name of the nominal feature. The (like kind) measured feature is named based on the Resulting Point Name argument.
Return Description	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to measure the feature.

Guide Objects in 6D based on Point Measurements

Function Description	<p>6D build function using a 3D measurement system. Specify the instrument index along with destination and moving reference groups. The interface displays the 6D (e.g., dx, dy, dz, dRoll, dPitch, dYaw) of the difference between the destination and moving reference groups.</p> <p>List all of the objects to be moved with the moving reference group (e.g., surfaces, lines, frames, planes, circles, etc.). An Object Name Reference List argument is used to enter or reference object names.</p> <p>Identify a third group of points which is continuously measured from an initial position. The third group is called the Initially surveyed group. As the initial surveyed group position and orientation is updated the moving group and list of objects are moved along with them.</p> <p>The 6D update (position/rotation) on the dialog is color coded for in/out of tolerance based on two tolerance vectors. Note that the tolerance vectors are optional. Specify the x, y, and z tolerances with the Position Vector Tolerance argument. Specify the roll, pitch, and yaw tolerances with the Position Vector Tolerance argument.</p>
Return Description	<p>Returns success after the dialog is closed. Failure is returned when the index is not found or the interface is not active.</p> <p>Using the moving reference as the Initially surveyed group works.</p> <p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p>
Remarks	The instrument interface must be active to for it to measure the points.

Align Two Targets with Axis (WCF - X)

Function	Align two targets with the working frame's X-axis. Two targets specified in function arguments are repeatedly measured. The targets are saved in a group
----------	--

Description	<p>named in the Initial Measured Group argument. Each cycle of the targets increments the group name.</p> <p>A vector is computed between the two measured targets after each measurement cycle of the two targets. The angular (Pitch and Yaw) differences between the vector and the working frame's X-axis is computed after each measurement cycle. The alignment difference is shown on the function's dialog as delta Pitch and Yaw angles. Their units are decimal degrees.</p> <p>Set vector tolerance argument's Y and Z components to colorize the Pitch and Yaw delta values for their tolerance condition.</p>
Return Description	<p>Returns success after the dialog is closed. Failure is returned when the index is not found or the interface is not active.</p> <p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p>
Remarks	The instrument interface must be active to for it to measure the two targets.

Crib Sheet Operations...

Run Crib Sheet

Function Description	Runs a crib sheet with an instrument referenced by index. Specify the collection name containing the crib sheet measurement sequence and the name of the crib sheet. The instrument index is relative to the active collection.
Return Description	<p>Returns success after the dialog is closed. Failure is returned when the index is not found or the interface is not active.</p> <p>The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.</p>
Remarks	The instrument interface must be active to for it to run the crib sheet functions.

Laser Projection...

Project Objects

Function Description	Function sends a list of objects (e.g., splines, lines, circles, vector groups) to a laser projector interface. Create a list of object names with the Object Name Reference List editor. Specify a laser projector instrument interface.
Return Description	Returns success after the sending the list of objects to the interface. Failure is returned when the index is not found or the interface is not active.
Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

The instrument interface must be active to for it to send the objects to the projector.

Stop Projection

Function Description	Stops the projection from a laser projector interface. Specify a laser project instrument index.
Return Description	Returns success after the sending the stop command to the projector interface. Failure is returned when the index is not found or the interface is not active. The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to stop the projection.

Advanced Instrument Operations...

Issue Instrument Actuator Command

Function Description	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. Specify the instrument index and the command string.
Return Description	Returns success after the sending the command string interface. Failure is returned when the index is not found or the interface is not active. The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3. The instrument interface must be active to for it to act on the actuator command function.
Remarks	The command strings are specific to the instrument interface.

Set Instrument Axes

Function Description	Set instrument axis positions. Typically used with Robot interfaces. Specify an instrument index and then a list of the joint angles for each joint. The sequence of the double values in the list correspond to the sequence of joints in the interface model. The Number of Step argument controls how many intermediate positions the instrument is shown in the graphics view. Use this value to control movement animation.
Return Description	Returns success after the setting the instrument axes positions. Failure is returned when the index is not found or the interface is not active.

Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
---------	---

Set Alignment Projector

Function	
Description	Sets alignment projector.
Return	
Description	
Remarks	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.

Cloud Viewer Operations...

Clear Cloud Viewer

Function	
Description	Clears an instrument interface's cloud viewer.
Return	Returns success after the clearing instruments interface's cloud viewer. Failure is returned when the index is not found or the interface is not active.
Description	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to clear the cloud viewer. The interface ignores this command when it does not support a cloud viewer.

Send Cloud To SA

Function	Sends all the visible points in a cloud viewer to SA. Specify the name for the cloud and the instrument index.
Description	
Return	Returns success after the sending the points in the cloud viewer to SA. Failure is returned when the index is not found or the interface is not active.
Description	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to send point clouds to SA. The interface ignores this command when it does not support a cloud viewer.

Set Filter

Function	Set a cloud viewer's Quality Filter value. Specify the instrument index and the Quality Filter number for the points in the cloud viewer. Points with a quality
----------	---

Description	threshold lower than the filter value are not shown.
Return Description	Returns success after the setting the Quality Filter for the cloud viewer. Failure is returned when the index is not found or the interface is not active.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to set filter value in the cloud viewer. The interface ignores this command when it does not support a cloud viewer.

Save Point Cloud File

Function Description	Saves points in a cloud viewer to a file. Specify the instrument index and path/file name. The data is saved to an ASCII file when the Boolean flag "Save as ASCII" flag argument is true. A false value saves the points to a binary file.
Return Description	Returns success after the saving the cloud points to file. Failure is returned when the index is not found or the interface is not active. Failure is also returned when the path/filename to the file are not valid.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to save the cloud points to the file. The interface ignores this command when it does not support a cloud viewer.

Load Point Cloud File

Function Description	Loads points from a file into an instrument interface's cloud viewer. Specify the instrument index and a path/file name.
Return Description	Returns success after the loading the cloud points to the cloud viewer from the file. Failure is returned when the index is not found or the interface is not active. Failure is also returned when the path/filename to the file are not valid.
	The instrument must be in the active collection. Specify its index with the Instrument ID argument. Instrument indexes are zero based. As an example the fourth instrument in a collection has an index of 3.
Remarks	The instrument interface must be active to for it to load the points to the cloud viewer. The interface ignores this command when it does not support a cloud viewer.

Utility Operations...

Set MP Step Mode

Function Description	Sets the MP step mode to either Single Step or Auto Run. Single Step mode allows a user to step through the script step by step. When in Auto Run mode MP runs each step one right after another.
Return Description	Returns success.
Remarks	A transition to Auto Run mode start after executing the step. When changing the mode to Single Step the MP pauses at the next step in the script.

Delay for specified time

Function Description	Starts a timer dialog and counts down for a specific period of time. Specify the amount of time and units for the value. The MP waits (e.g., is delayed) for the specified amount of time.
Return Description	Returns success after delaying for the specified period of time.
Remarks	Set or reference the amount of time to delay the Measurement Plan.

Speak To User

Function Description	Sends a string to SA's Speech Server. The Speech Server must be active for the string to be spoken.
Return Description	Returns success after speaking the string. Failure is returned when SA's Speech Server is not active.
Remarks	Enter or reference the string to speak from other MP step arguments.

Set Working Frame

Function Description	Makes a frame the Working Frame for the job. Specify the frame name by either entering the Collection::Object Name or referencing the Collection::Object Name from other MP step arguments.
Return Description	Returns success after making the frame SA's Working Frame. Failure is returned when the frame name is not found.
Remarks	

Set Working Color

Function Description	Set SA's working color. Specify the color by referencing it or picking from a color palette.
----------------------	--

Return	
Description	Returns success after setting the working color.
Remarks	

Set Object's Color

Function	Sets a list of objects to a color specified by a color argument. Specify the color by referencing it or picking from a color palette. Enter or reference a list of object names.
Description	
Return	Returns success after setting the objects to the selected color. Returns failure if the objects are not found.
Description	
Remarks	Use the runtime object selection function to get a current set of objects in a Collection::Object Name Ref List. Reference the Collection::Object Name List to set their color.

Set Working Color Auto Increment

Function	Set SA's color auto-increment property. Configure the Auto Increment flag to true and SA increments the working color after a new object is created. Set the flag to false and SA does not change the working color unless set by the user or the Set Working Color MP function.
Description	
Return	Return success after setting the Auto-Increment property.
Description	
Remarks	

Move Robot to Frame

	Sends a Robot to a Frame. Name the device and destination frame.
	A flag is available to control whether the step waits for the device to complete the move. When set to true the step waits for the device to acknowledge that it has completed the requested move.
Function	A Results Only argument of the actual transform is set after the move is complete.
Description	
Return	Returns success after sending the device to the named frame. Failure is returned when the device and destination frame name are not found.
Description	
Remarks	Used with Robots to move them from their current positions to a frame orientation. Use the Acknowledge Arrival flag to pause the MP letting the Robot complete the move before sending the next command.

Move Robot to Named Destination

Function	Sends a Robot to a named destination. Name the device and destination. Either enter the name and destination or reference them from other MP step
----------	---

Description	arguments. For example SetMotorSpeed, SetMotorNumber, or MoveMotorPositive25 are possible destination variables. Typically the interface to the system has a table of strings that reference commands to the controller. A flag is available to control whether the step waits for the device to complete the action. When set to true the step waits for the device to acknowledge that it has completed the requested command. A Results Only argument of the actual transform is set after the action is complete.
Return Description	Returns success after sending the device to the named destination. Failure is returned when the device and destination name are not found.
Remarks	Used with robots or motors to set/control their current state. Use the Acknowledge Arrival flag to pause the MP letting the Robot or motor complete the action before sending the next command.

Send MP Result to External Device

Function Description	Send success, failure, or partial success to another application. Specify the other application by identifying the computer running the application with its TCP/IP address and socket port. The IP address and socket port is where this step sends the packet too. Either pick or reference the result value to send in the packet. Choose Success, Failure, or Partial Success.
Return Description	Returns success after sending the packet. Failure is returned when the address or port is not valid.
Remarks	Used to signal process events and actions to other applications. Specifying the computer's IP address is valid when the other app is on the same machine. The other application must be up and running and listening for data on the specific port used by this step.

Send MP Step's Status to External Device

Function Description	Sends the results of a specific step to another application. Specify the computer running the other application with its TCP/IP address and socket port. Enter or reference the MP step ID. Its return value (e.g., Success, Failure, or Partial Success) is sent to the other application.
Return Description	Returns success after sending the packet. Failure is returned when the address or port is not valid.
Remarks	Used to signal process events and actions to other applications. Specifying the computer's IP address is valid when the other app is on the same machine. The other application must be up and running and listening for data on the specific port used by this step.

Delete Objects

Function Description	Deletes a list of objects from the SA job database. Specify a list of objects with the Object Name Ref List argument.
----------------------	---

Return Description	Returns success after deleting the objects in the list. Failure is returned when the objects are not found.
Remarks	Create or reference an existing Object Name Ref List argument.

Highlight Objects

Function Description	Highlights a list of objects from the SA job database. Specify a list of objects with the Object Name Ref List argument.
Return Description	Returns success after highlighting the objects in the list. Failure is returned when the objects are not found.
Remarks	

Highlight Point

Function Description	Highlights an individual point in the graphical view and in the database tree. A Boolean "Show Point" argument enables the function to set the show and hide properties. Specify by entering, referencing, or using the dialog to identify the point name.
Return Description	Returns success after setting the point display properties.
Remarks	Leave the point name empty to clear all the point highlight properties.

Set Special MP Mode

Function Description	Configure Measurement Plan and SA into a special mode. These modes are generally specific to particular applications.
Return Description	Return success.
Remarks	

Increment Point Name

Function Description	Increments a point name with an integer value. The resultant point name is provided in a Results Only Point Name field. Specify by entering, referencing or using the dialog entry form to set the Base Point Name. The increment value is added to the base name with SA's point naming convention. Reference the resultant point name from subsequent steps to use the incremented point name.
Return Description	Returns success after incrementing the point name.

Remarks

Set Interaction Mode

Configure SA and MP interaction modes. Set SA's interaction mode to Manual, Automatic, or Silent. The manual mode is the default.

Measurement Plan's interaction mode is set to one of the following. Halt on Failure Only, Halt on Failure or Partial Success, and Never Halt are the interaction mode options for measurement plan operation. Halt on Failure or Partial Success is the default mode.

Function
Description

The Measurement Plan Dialog Interaction Mode controls how much access a user has with SA menu choices while running an MP script. This augment sets SA parameters to either Block or Allow Application Interaction.

Return
Description

Returns success after configuring each of the interaction control parameters.

When running in Manual Mode, MP presents typical dialogs to the user and asks them to control the process. The automatic mode eliminates interaction dialogs where possible. While in silent mode user interaction dialogs are suppressed. The assumption is silent mode is that the Measurement Plan script manages the success or failure of each step in the process.

The interaction mode setting allows the script to automatically stop when a step fails. This mechanism is useful when debugging a complex process. When ready for production implementation changing the mode to Never Halt allows the error trapping and conditional testing logic to control the process.

Remarks

Setting the application interaction to Block access prevents user intervention. The Allow Application Interaction allows the user to control and manage changes during the process.

REVERSE DESIGN

This will present a reverse design process steps.

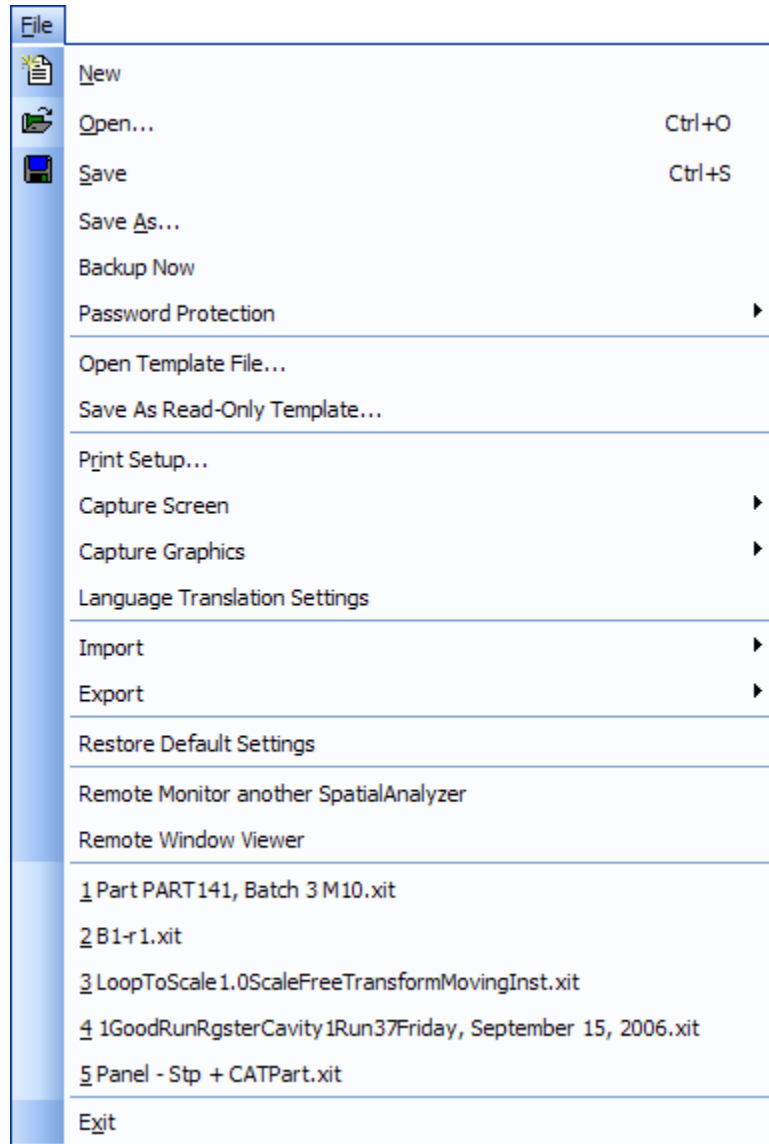
The process steps for reverse engineering the barrel surface in SA. Please note the process has quite a few parameters that can be tweaked to meet requirements. Regard the numbers shown here as suggestions.

1. Import the ACSII data into a Point Cloud
2. Open the point cloud properties dialog. Set the Draw increment to 1 ... this shows all the points in the cloud. Set the target offsets to there appropriate values if they have offsets.

MENU REFERENCE

This section will discuss each of the menus within **SpatialAnalyzer**. The intent is to provide a structured reference to each of the features of the program. As a result, there may be some overlap with other sections of the manual.

File



File Menu

New**Open...** **Ctrl+O****Save** **Ctrl+S****Save As...**

These menu items all reflect the standard Windows commands for document management. Each job in **SpatialAnalyzer** is stored in an individual job file. When these files are opened, they are represented in **SpatialAnalyzer** as a document.

Backup Now **Alt+SPACE**

This option makes a complete backup of the job in its current state. These backup files are placed in the “Analyzer Data\Backup” directory. The naming convention is to use the name of the job file and append the date and time of the backup.

Periodically, **SpatialAnalyzer** will automatically backup the current job file. These backups occur after a certain number of significant operations.

Since the backup files are complete copies of the job, you can go back to these files to recover from loss of data, user error, etc.

Open Template File...

Save As Read-Only Template... This option saves the current job with the .XTP extension. This indicates that the file is a template. The file is, however, simply a copy of the current job. It is a template only in that its intended use is that the user load it before beginning a new job.

Since there are a wide array of settings and customizations a user may configure within the **SpatialAnalyzer**, template files are necessary to avoid repeating the same option selections each time a new job is created.

Print Setup...

This is the standard Windows print setup option.

Print Screen

This option contains sub options for “to Printer” or “to File”. This function allows you to perform a screen capture.

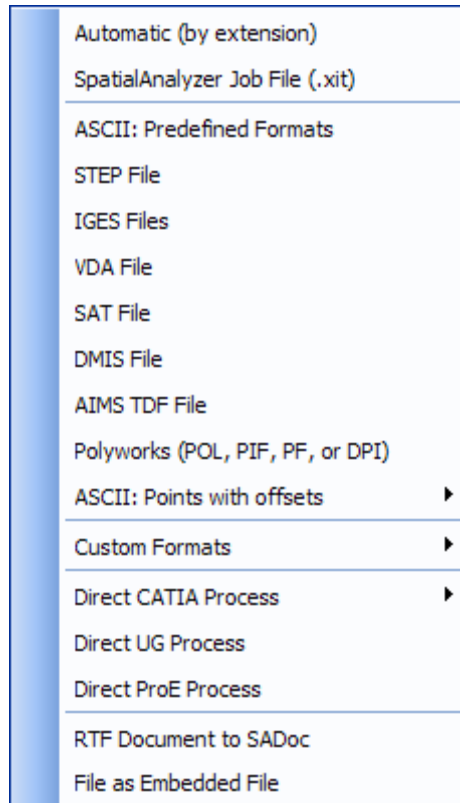
To Printer

This option will print the currently active graphical view. After you select the printer, you will be given several options. One of the options will be the resolution setting. Choose a low percentage initially and raise the percentage with caution. This is because the printing process with high resolution often fails due to printer memory, system memory, or printer driver quirks.

Capture to file

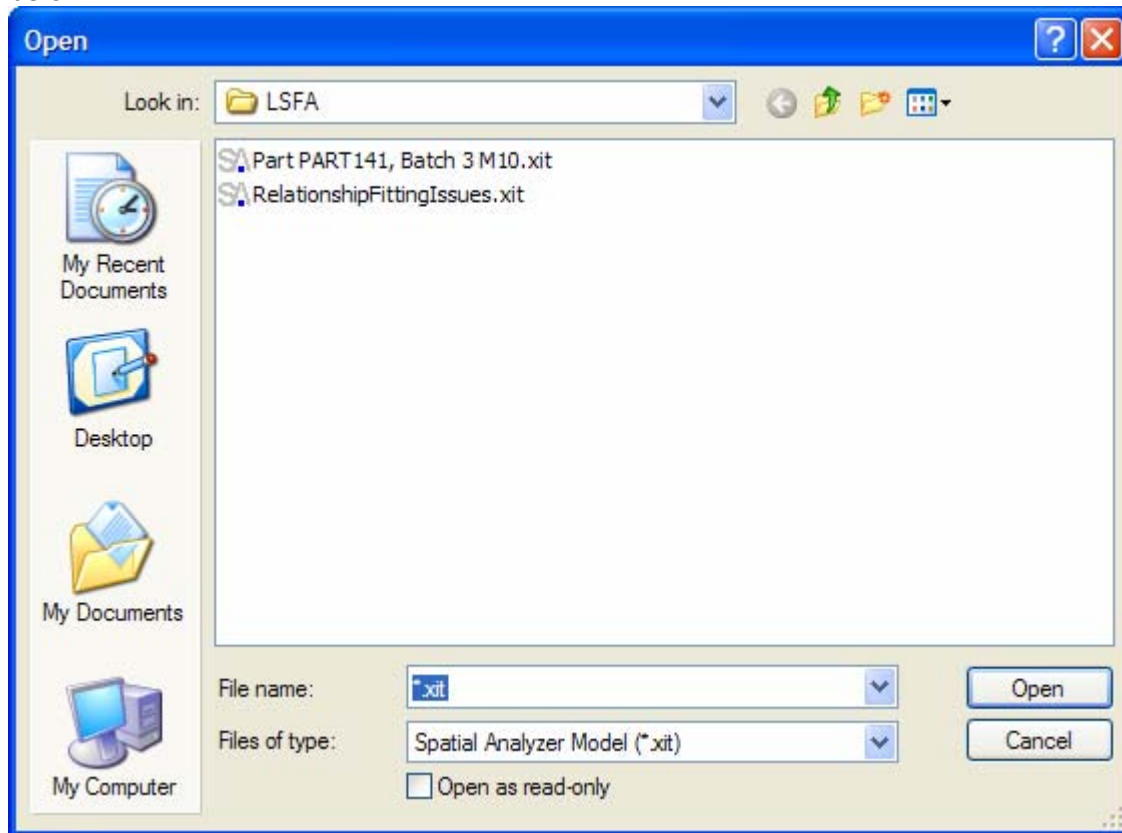
Import

This option allows you to import files into a **SpatialAnalyzer** job. The available file types are listed below.



File Import Menu

Each of the import options starts with a Windows file selection dialog as shown below.



File Selection Dialog

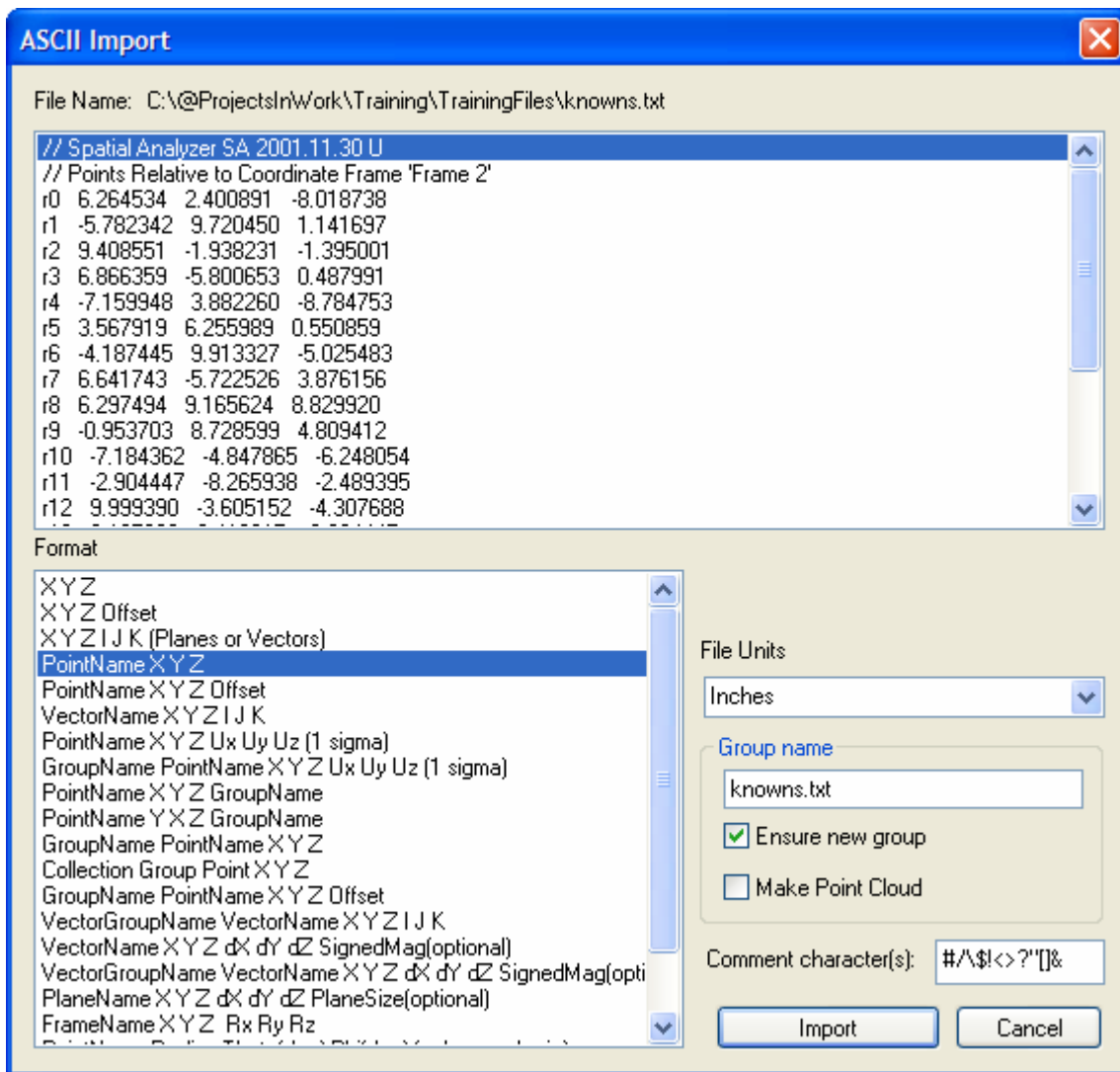
SpatialAnalyzer File (.xit)

This option allows you to merge another **SpatialAnalyzer** job file into the current job. After selecting the **SpatialAnalyzer** file to import, each collection in the file is imported into a collection of its own in the current job.

Initially, all of the objects (with the exception of the WORLD frame in this case) will be selected.

ASCII: Predefined Formats

This option will allow you to import an ASCII text file containing coordinate information in a variety of common formats. After selecting this option and picking the file to import, the following dialog will appear.



Predefined ASCII File Import Dialog

In this dialog, you will be able to configure **SpatialAnalyzer** to read the text file in a variety of formats. Select the line to start importing from in the file text list box. The specific point formats are listed in on the left of the dialog. Select the format that best represents the points to import.

If the file you are importing has intermixed comment lines; ensure that the character identifying a comment is listed in the Comment character's field. This field can also be used to filter out points from the import. As an example if you want to exclude all points starting with a 'E' from the import, include a 'E' in the Comment character's field. Any line starting with a 'E' is then treated as a comment.

The File Units option allows you to specify the units of the data in the file. If this setting is different that the units that are being used in the current job, the coordinates will be converted to the current **SpatialAnalyzer** job units as they are imported.

Name the object to import the points into ... note this field will be ignored if the points have group fields.

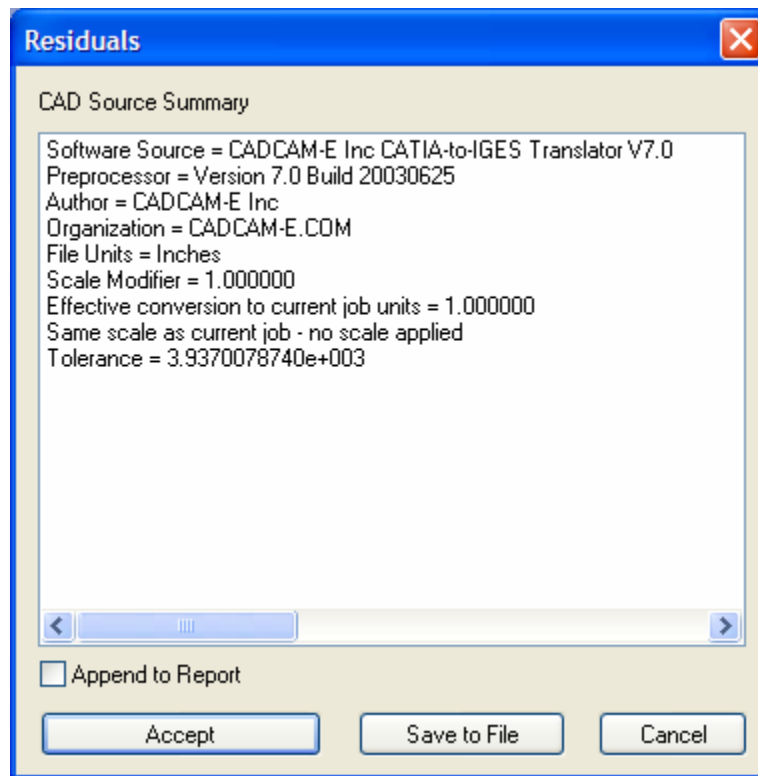
Select the Make Point Cloud if the number of points in the file is large.

When the importer is configured; select the Import button to process the file. All of the points, groups, and clouds are imported into the active collection.

STEP File

This option will import an ASCII formatted Standard for the Exchange of Product Model (STEP) file, ISO 10303. STEP is an international standard to support industrial automation systems and integration. It accomplishes that goal by establishing a common format for product data representation and exchange. The ISO TC184/SC4 committee controls the development of the STEP standard. ISO 10303 is actually a collection of standards called Application Protocols (AP). **SpatialAnalyzer** supports AP-203, which is the Configuration Controlled Design component of the STEP standard.

After selecting the file you wish to read a filter dialog as shown below, is presented to you. You are then able to select the type(s) of entities that you want to import into **SpatialAnalyzer**. You can also set the Group name in the Points edit box.



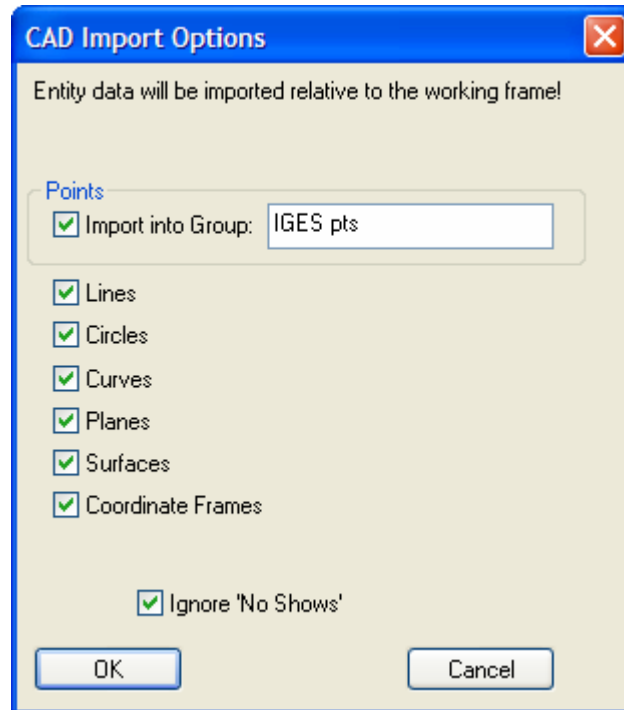
STEP Import Filter Dialog

After hitting the OK button, the entities are imported into the appropriate database objects. Each of the entities will appear in the Graphics and Database Tree views.

NOTE: All imported entities will be placed relative to the current working frame in **SpatialAnalyzer**.

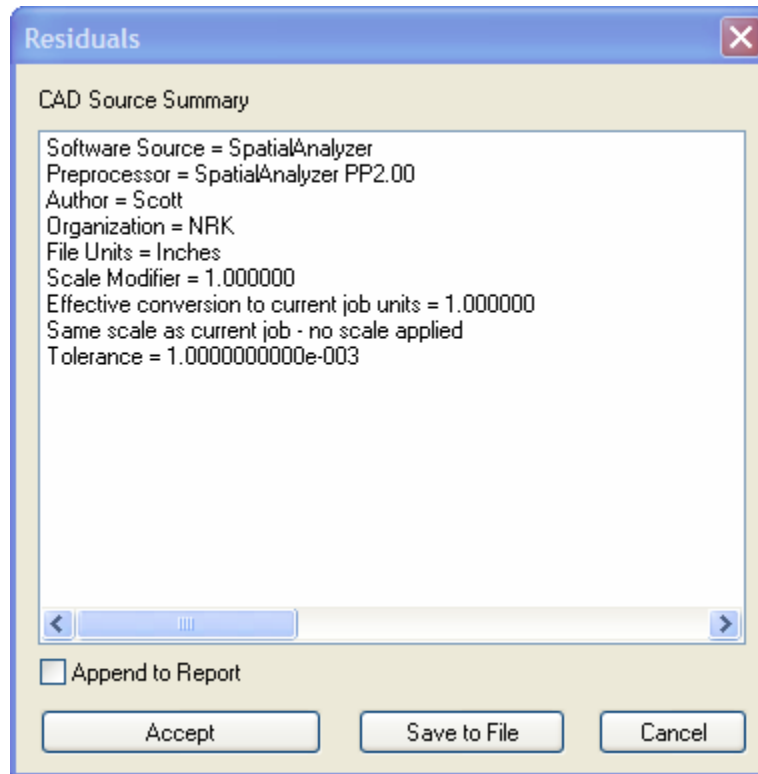
IGES Files

This option will import an ASCII formatted IGES file. After selecting the file you wish to read, you will have the opportunity to filter the import process based on the IGES file entity type information. Since **SpatialAnalyzer** preserves the type and name information found in an IGES file, you can base your type selection on the CAD modeling convention you are accustomed to. The type selection dialog is shown below.



IGES Import Filter Dialog

After selecting the entities to import the file is scanned to prepare a summary. This summary information will be presented as shown below.



IGES Import Summary

After reviewing the summary results, press **Accept** if you wish to continue with the Import process. The IGES Import Options dialog, shown above, will appear next. It will allow you to configure the point naming conventions as well as filter unwanted entities.

After this final selection, the import process will complete, and the IGES entities will be in your **SpatialAnalyzer** model.

NOTE: All imported entities will be placed relative to the current working frame in **SpatialAnalyzer**.

AIMS TDF File (Beta)

The AIM file import is a custom Measurement Plan Interface. It is able to scan data capability for the MClient Metronor-AIMS measurement plan interface.

Polyworks (POL, PIF, PF or DPI) File

Importing Polyworks Files (POL, PIF, PF, and DPI) option added. This option facilitates interfacing with instruments like laser scanners.

ASCII: Points with offsets

<Name> <X> <Y> <Z> <Offset>

2 Files: <Name> <X> <Y> <Z> + <Name> <Offset>

Custom Formats

Auto Callout File

A custom interface for automatically generating Callout views for a job file.

VSTARS .xyz file

This option allows you to import a file from a GSI Digital Photogrammetry system. These files are ASCII text files containing coordinate information.

xyz, IJK File (IJK)

A special case importer for files with coordinates and unit vectors.

IMETRIC File

1-D Data (DataMyte)

Direct CATIA Process (for V4, V5, CGR)

Importing CAD models directly into SA is accomplished with a third party library. Imports from V4 (.model), V5 (.CATPart and CATProduct) , and CGR files are supported.

The process works by first selecting the native CAD file. SA directly reads the file and converts the data in the file into SA objects resulting in a standard xit file. Selecting the file to convert is the only step you need to complete.

Before these functions are accessible the library must be procured from the supplier and installed with a license on your computer. Please contact sales@kinematics.com to acquire a license.

Direct UG Process

The Direct UG Process uses the same mechanisms as documented in the Direct CATIA Process.

Direct ProE Process

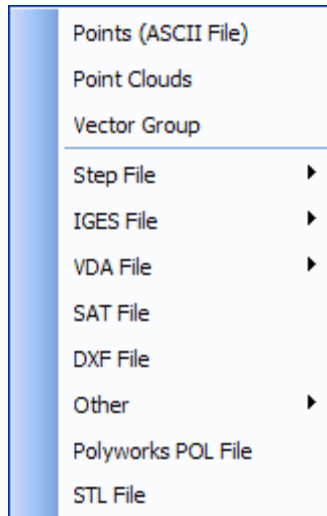
The Direct ProE Process uses the same mechanisms as documented in the Direct CATIA Process.

RTF Document to SADoc

This function imports a Rich Text Format (RTF) document into the SA job file. The document is saved as a SADoc on the treeview in the active collection. To open the document from SA; select it and then right-click on it to get its respective menu functions. Select the Show option from the menu to open the document.

Export

As with the import functions, **SpatialAnalyzer** is capable of exporting a number of different file formats. These are shown on the File Export menu in the figure below.

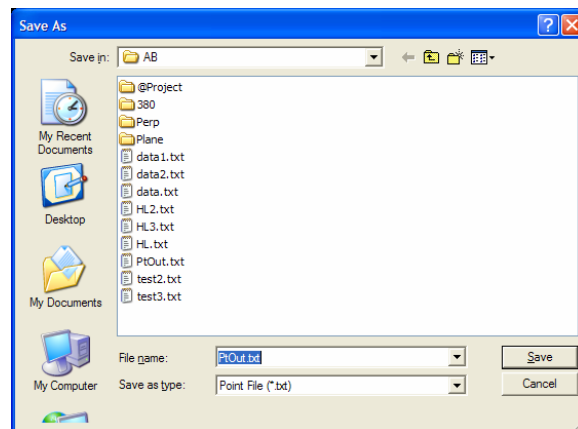


File Export Menu

Points (ASCII File)

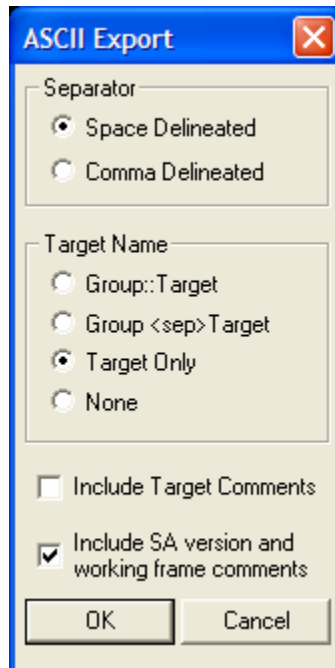
This option allows you to export coordinate values relative to the working coordinate frame. After selecting this option, you will be prompted to select the points that you wish to export. You can either click select them on the screen, drag a window on the screen by holding the SHIFT key, or pressing F2 to select from a tree-view.

After picking the points, select the file you wish to place them in, and **SpatialAnalyzer** will export the points.



File Export Save-As dialog

There are a number of common formatting choices for the exporting process. Select the profile that meets your needs. The ACSII Export format selection process dialog is shown in the figure below.



File Export Save-As dialog

Select either space or comma delimited and whether you need the group name added as a prefix to the point name. Export options for target comments and whether or not you would like to have the SA version and working frame adding as a header in the output file are available on the dialog.

Hit the OK button to export the points to the file.

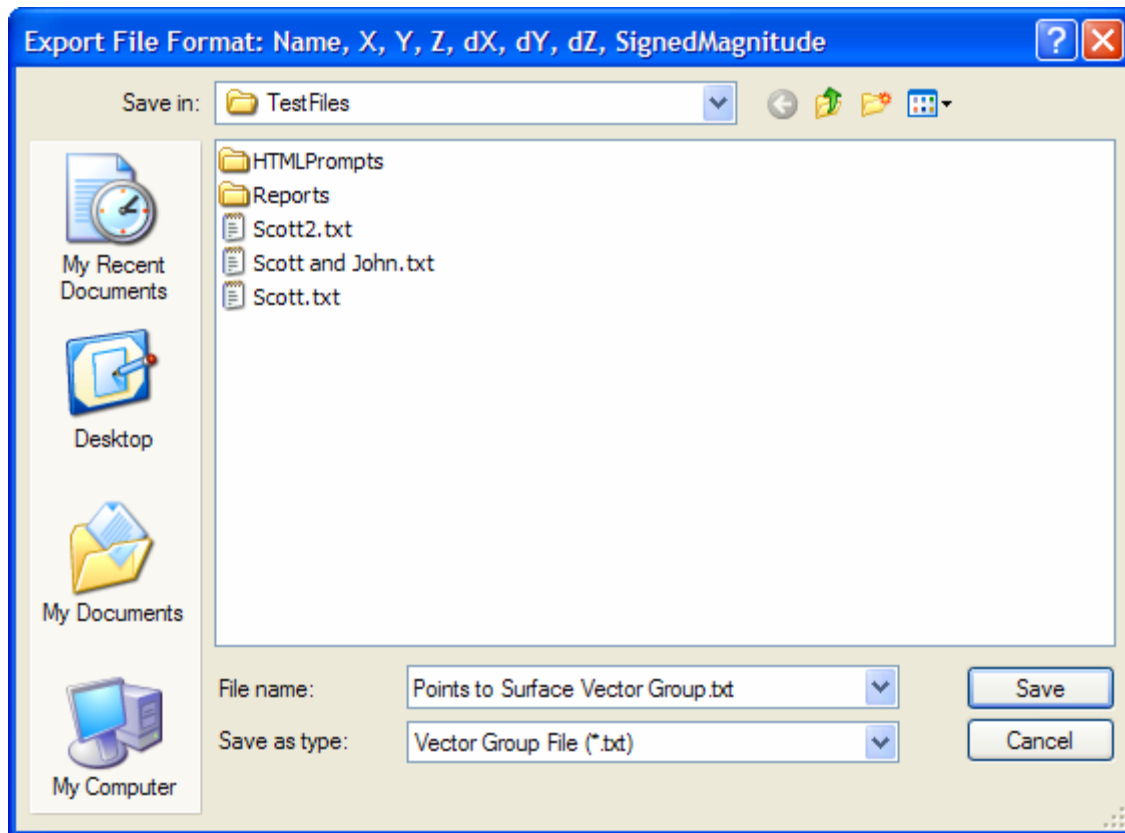
This same function (i.e., to export a point group) is also available by right-clicking on the point group in the treeview.

Vector Group

This option allows you to export a vector group relative to the working coordinate frame. After selecting this option, you will be prompted to select the vector group that you wish to export. You can either click select them on the screen, drag a window on the screen by holding the SHIFT key, or pressing F2 to select from a tree-view.

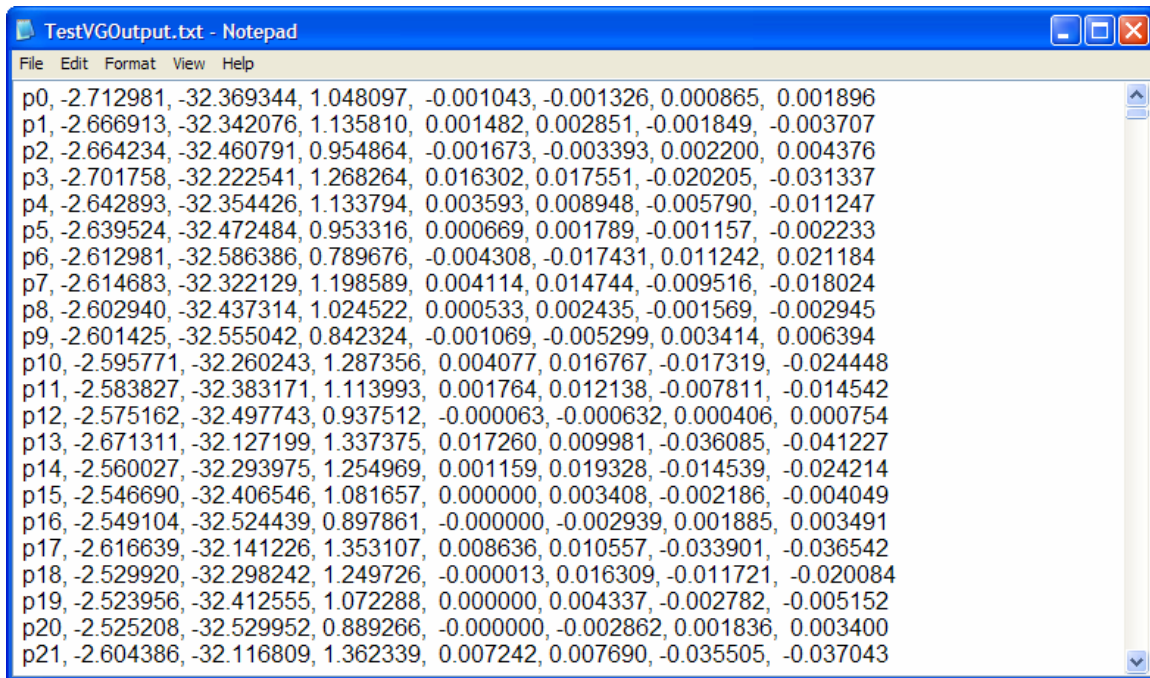
After picking the vector group, select the file you wish to place them in, and **SpatialAnalyzer** will export the vectors in a Name, X, Y, Z, dx, dy, dz and Signed Magnitude.

The X, Y, and Z values are the beginning vector points coordinates.



Vector Group Export Save-As dialog

After exporting the vectors to the file, it is opened for your review. An example is shown below.

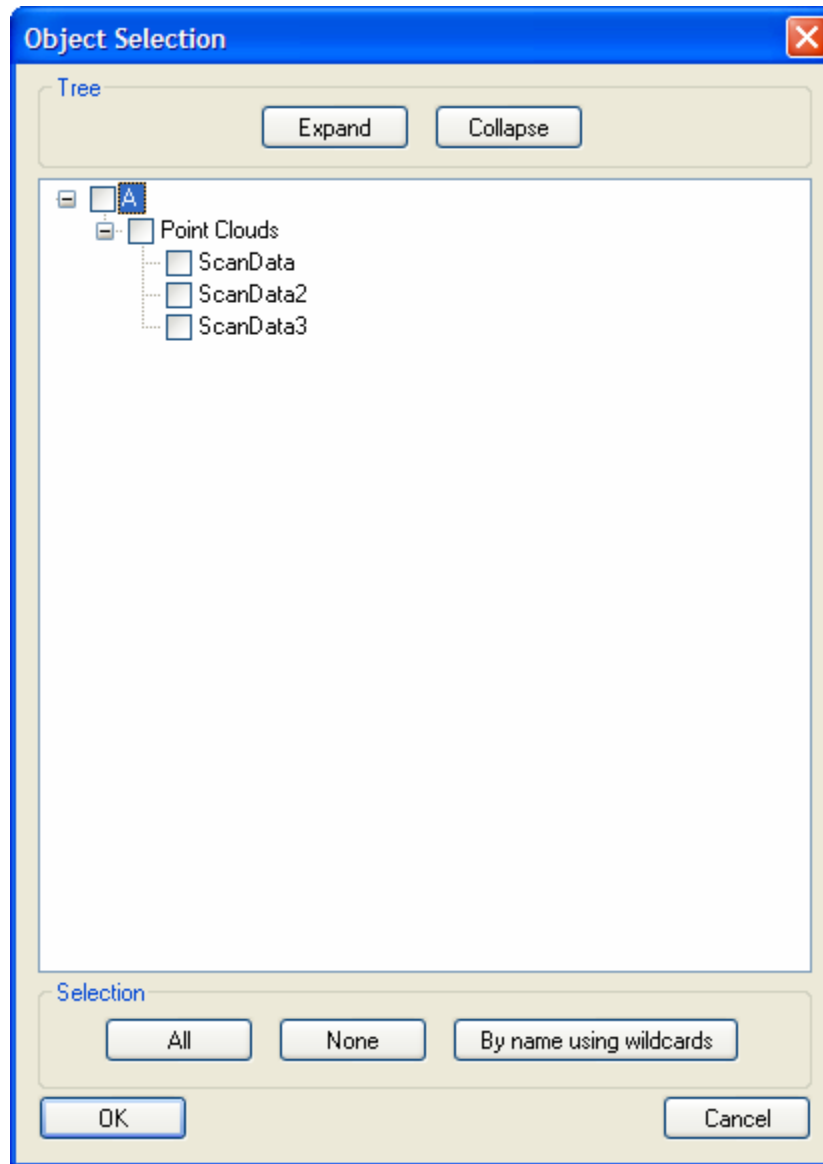


```
TestVGOutput.txt - Notepad
File Edit Format View Help
p0, -2.712981, -32.369344, 1.048097, -0.001043, -0.001326, 0.000865, 0.001896
p1, -2.666913, -32.342076, 1.135810, 0.001482, 0.002851, -0.001849, -0.003707
p2, -2.664234, -32.460791, 0.954864, -0.001673, -0.003393, 0.002200, 0.004376
p3, -2.701758, -32.222541, 1.268264, 0.016302, 0.017551, -0.020205, -0.031337
p4, -2.642893, -32.354426, 1.133794, 0.003593, 0.008948, -0.005790, -0.011247
p5, -2.639524, -32.472484, 0.953316, 0.000669, 0.001789, -0.001157, -0.002233
p6, -2.612981, -32.586386, 0.789676, -0.004308, -0.017431, 0.011242, 0.021184
p7, -2.614683, -32.322129, 1.198589, 0.004114, 0.014744, -0.009516, -0.018024
p8, -2.602940, -32.437314, 1.024522, 0.000533, 0.002435, -0.001569, -0.002945
p9, -2.601425, -32.555042, 0.842324, -0.001069, -0.005299, 0.003414, 0.006394
p10, -2.595771, -32.260243, 1.287356, 0.004077, 0.016767, -0.017319, -0.024448
p11, -2.583827, -32.383171, 1.113993, 0.001764, 0.012138, -0.007811, -0.014542
p12, -2.575162, -32.497743, 0.937512, -0.000063, -0.000632, 0.000406, 0.000754
p13, -2.671311, -32.127199, 1.337375, 0.017260, 0.009981, -0.036085, -0.041227
p14, -2.560027, -32.293975, 1.254969, 0.001159, 0.019328, -0.014539, -0.024214
p15, -2.546690, -32.406546, 1.081657, 0.000000, 0.003408, -0.002186, -0.004049
p16, -2.549104, -32.524439, 0.897861, -0.000000, -0.002939, 0.001885, 0.003491
p17, -2.616639, -32.141226, 1.353107, 0.008636, 0.010557, -0.033901, -0.036542
p18, -2.529920, -32.298242, 1.249726, -0.000013, 0.016309, -0.011721, -0.020084
p19, -2.523956, -32.412555, 1.072288, 0.000000, 0.004337, -0.002782, -0.005152
p20, -2.525208, -32.529952, 0.889266, -0.000000, -0.002862, 0.001836, 0.003400
p21, -2.604386, -32.116809, 1.362339, 0.007242, 0.007690, -0.035505, -0.037043
```

Vector Group Export example file

Point Clouds

Point cloud exporting is supported with an ASCII export of the X, Y, Z components for each point in the cloud. After selecting the file and its location into the SaveAs file dialog an Object selection dialog with all the Point Cloud in the job is presented (shown below.)



Cloud Selection Dialog

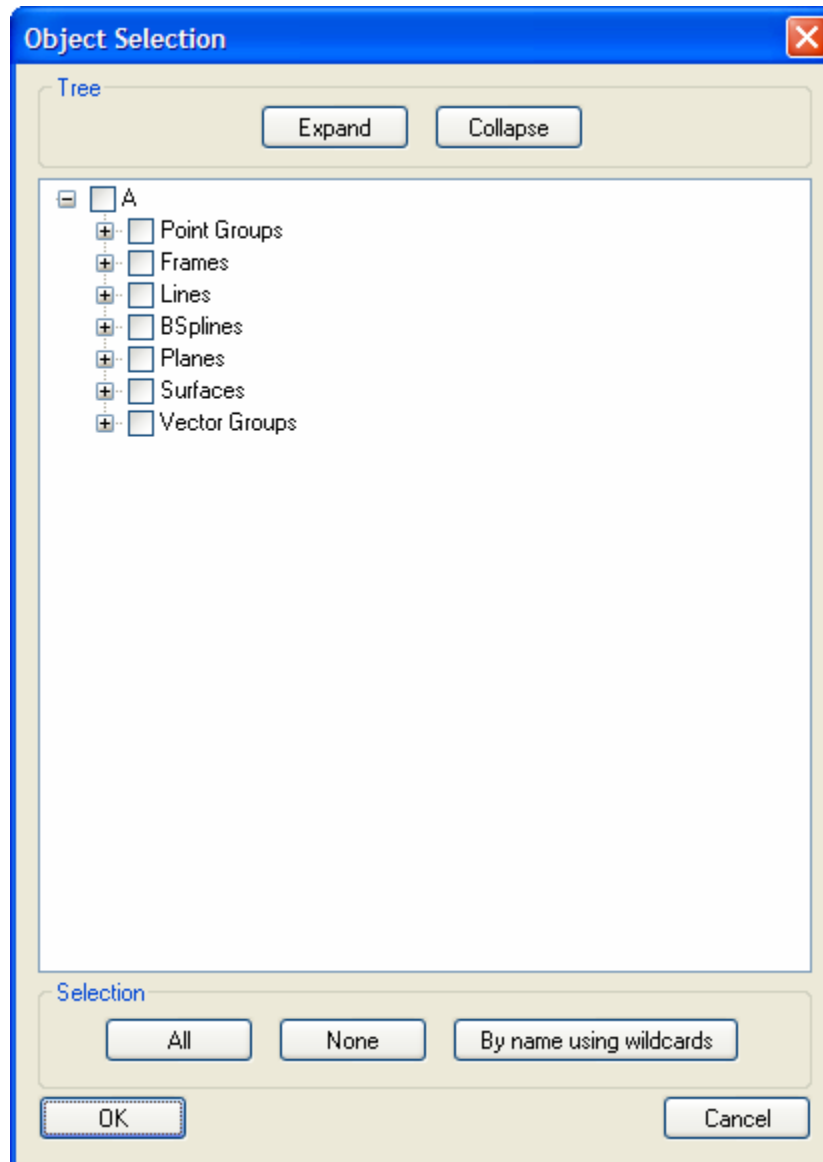
After hitting the OK button and the clouds of points are then saved to the file.

File Export (Entire Model)

Exporting objects from SA into a standard graphics file format is available in STEP, IGES, and VDA formats. The process is the same for each format. Enter the file name and its location into the SaveAs dialog and hit OK button on the dialog. Each of the compatible entities are then exported to the file.

File Export (Selected Objects)

After selecting this option, the first step is to select objects to export with any of the typical selection methods in SA e.g., graphical selection, treeview selection or the F2 selection dialog. The F2 selection dialog is shown below.



IGES Export Options Dialog

After selecting the objects to export choose a file and location for the export with the standard SaveAs dialog. The data is exported relative to the current Working Frame.

SAT File

This option will allow you to export parts of the **SpatialAnalyzer** job file into a SAT (i.e., Save As Text) format. An SAT file is used to exchange graphical entities between systems that use an ACIS based geometry engine. ACIS is an object-oriented geometric modeling toolkit designed for use as a geometry engine within 3D modeling applications.

After selecting the file location, all compatible SA entities are written to the SAT formatted file.

DXF File

This option will allow you to export points from the **SpatialAnalyzer** job file in a DXF format. After selecting the file location, you can specify which points to export to the DXF formatted file. You may choose the export either all of the points, or a specific group of points.

Points: Metric Vision Instrument Compensation Data Format

Exporting a point set to compensate the Metric Vision system is accomplished with this function. The selection control allows you to select the points to export and the location to save the file.

Imageware Cloud File

After selecting this option, you must select a location for the file. Once you do this, all of the points in the model will be written to disk as an ASC file. This file is capable of being imported directly into Imageware's Surfacar or Verdict package.

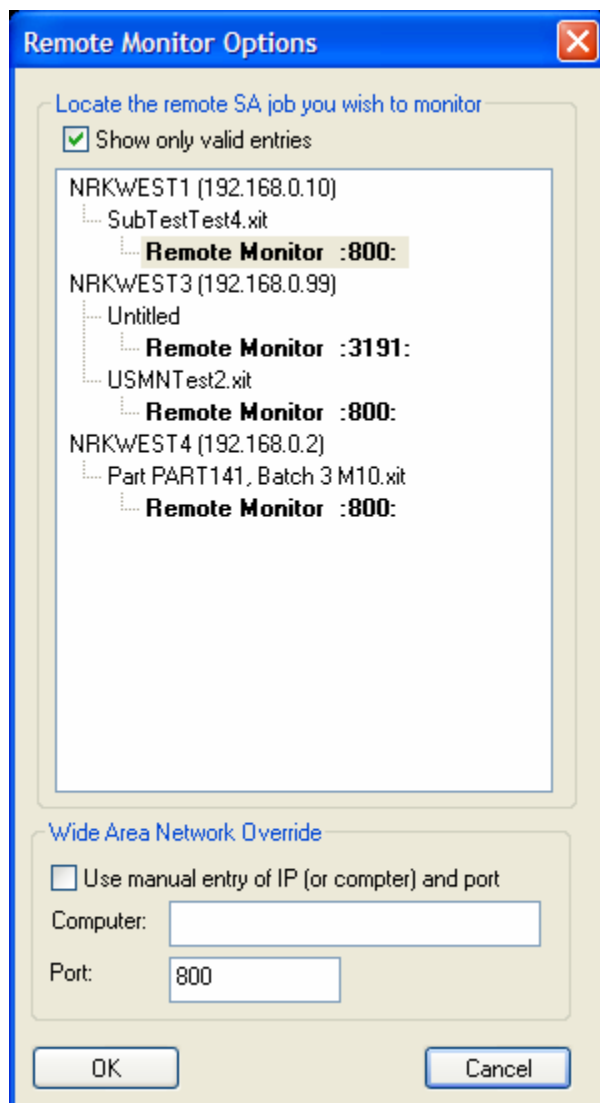
Polyworks POL file

Exporting a Polyworks POL Files is supported to facilitate SA real-time interface with instruments like laser scanners. Select the clouds to include in the export and then the file name and location. The point clouds are exported to the file in the POL format in the current working frame.

Remote Monitor another SpatialAnalyzer

Remote monitoring is a powerful feature in **SpatialAnalyzer**. It allows you to log into another **SpatialAnalyzer** job and view its entire contents. All you need to know about the remote job is the name of computer that is running **SpatialAnalyzer**.

This option asks you to select the job from a tree list of other machines running **SpatialAnalyzer** on your company's network. Next, it will locate the Analyzer, and transfer the currently active job to the current document in the local **SpatialAnalyzer**. When this occurs, notice that the time of the last update will be posted in the corners of the graphical window.

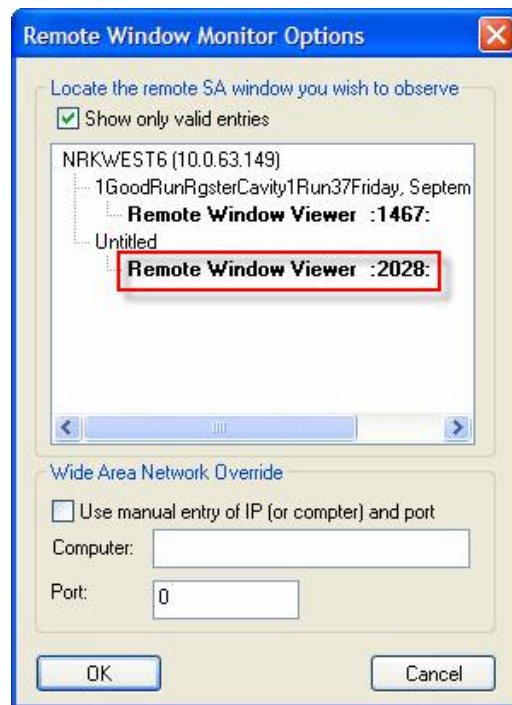


Remote Monitor Options Dialog

When you remote monitor, the current document in your local **SpatialAnalyzer** will be overwritten. If you wish to save the current job, be sure you do so before you initiate remote monitor mode.

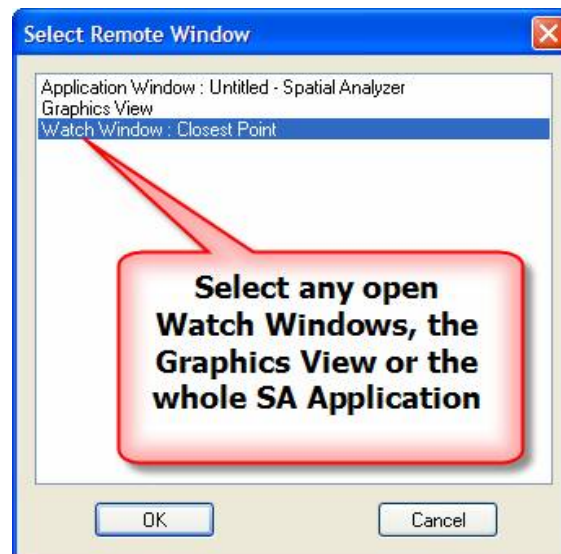
Remote Window Viewer

This mechanism is able to show Watch Windows, SA Graphics Window and the SA Application on other computers on the LAN running SA.



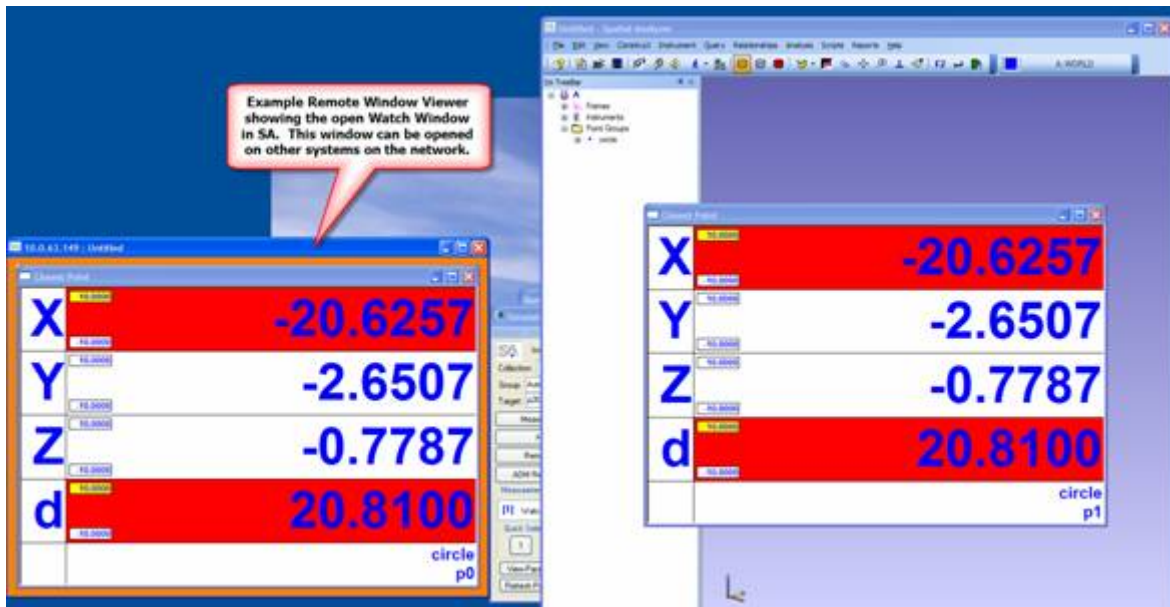
Remote Window Monitor Options Dialog

Select any open Watch Window, the Graphics View, or the whole SA Application graphical view.



Select SA Window to Watch Remotely

Example SA Remote Window Viewer shown below...



Example SA Remote Window functionality

This functionality works from any PC on the network including the system running SA. One option is to use a Tablet PC to run SA Remote Window Viewer on it.

Recent Files

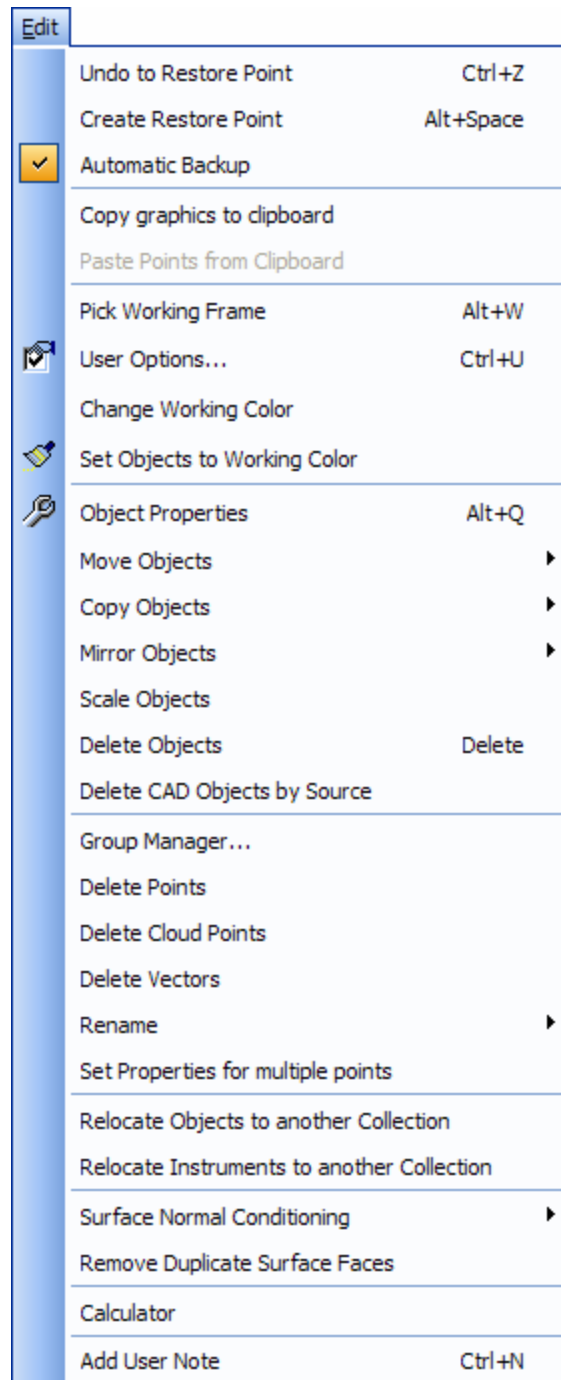
As with most Windows applications, **SpatialAnalyzer** keeps a list of the files you have most recently worked on at the bottom of the file menu. Selecting any of the files on the file menu will automatically load it into **SpatialAnalyzer**.

Exit

Exiting **SpatialAnalyzer** is accomplished like any other standard Windows application. **SpatialAnalyzer** will ask you if you want to save any changes before it completely exits.

Edit

The Edit menu in **SpatialAnalyzer** provides a set of capabilities to manipulate the properties of an entity within the **SpatialAnalyzer** database.



Edit Menu

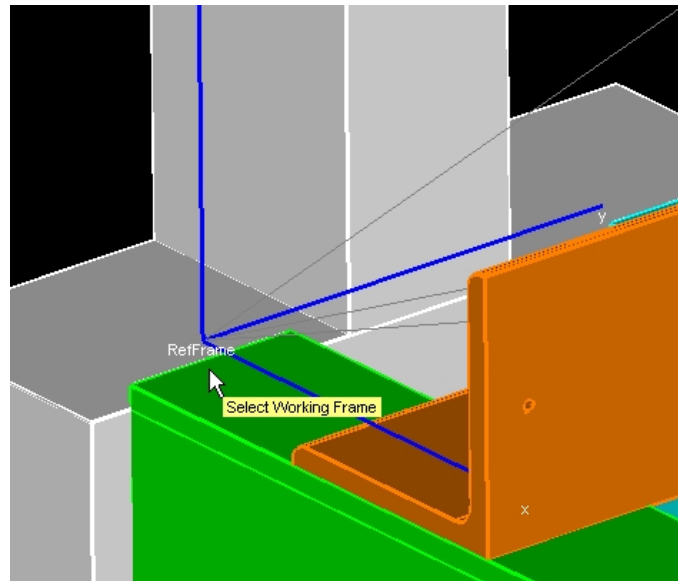
Copy graphics to clipboard

This option will place the graphics in the **SpatialAnalyzer** window into the Windows Clipboard. You may then paste these graphics into other applications.

Pick Working Frame

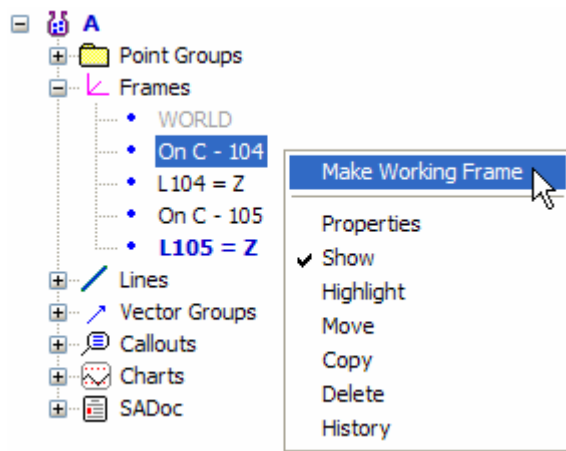
Alt+W

You will be prompted to select the frame that you wish to designate as the working frame. You may press F2 to pick the frame from list, or you can click on it in the graphical view as shown below. You can also change the working frame using the toolbar. There is a list of the frames in a pull-down combo box as well as a button.

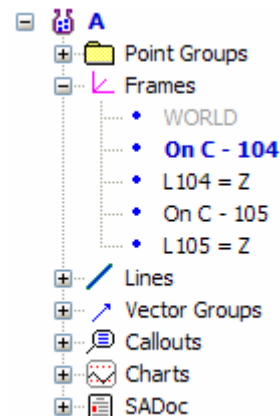


Pick Working Frame

The right-click menu for frames also supports changing the working frame. The figure below shows the menu and the process.



Right-click menu to Change the Working Frame



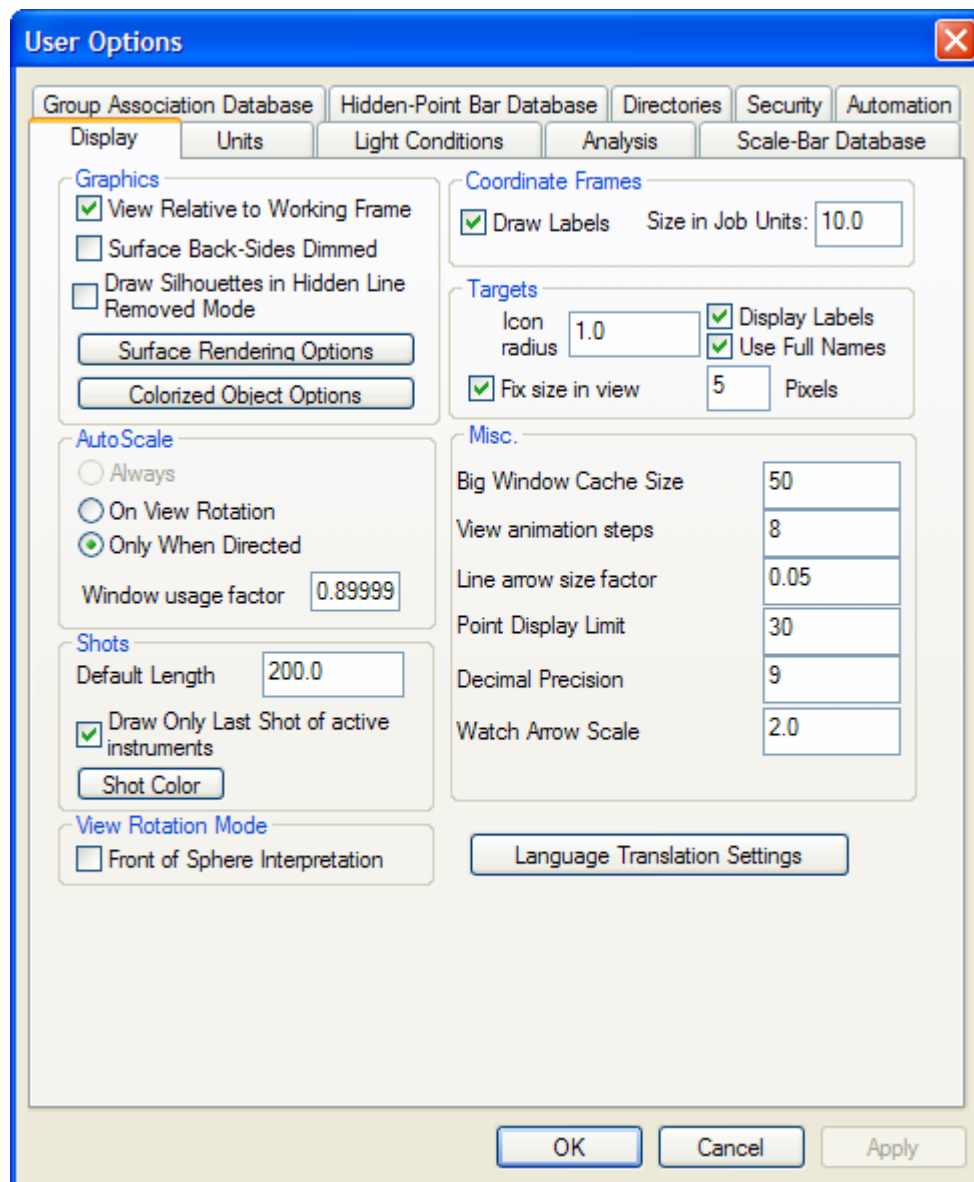
Right-click menu to Change the Working Frame

This action changes the working frame from the current one called, "L105 = Z" to the frame called "On C – 104." The results are shown in the figure above.

User Options...

Ctrl+U

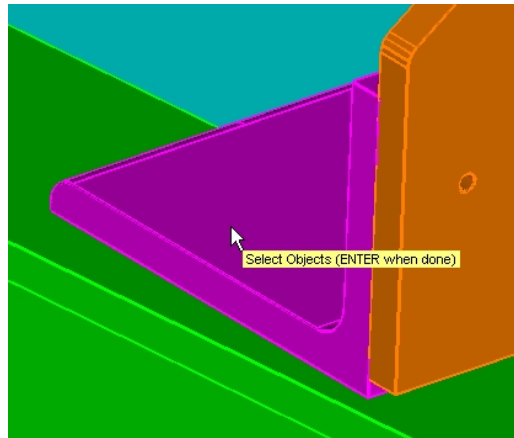
This menu item will display the User Options tab sheet. The options in this sheet are discussed in the of this manual.



User Options Dialog

Set Object's Color to Working Color

This option will prompt you to select the objects you would like to recolor. Once the objects have been selected, their color will be set to the current working color. The working color is set in the Color combo box on the toolbar.



Change Object's Color to the Working Color

Since groups of points are categorized as objects, you can color groups of points using the option as well.

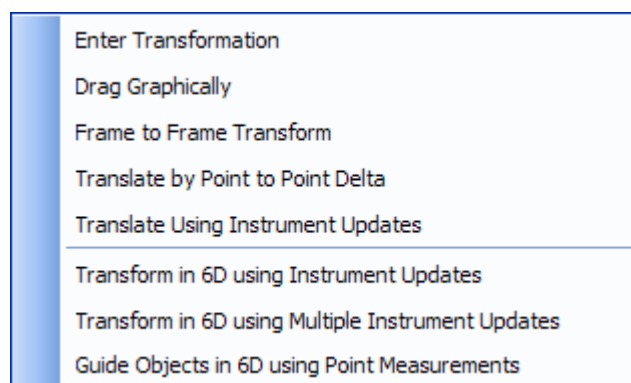
Object Properties

Alt+Q

You will be prompted to select an object. As usual, you can do this either by clicking in the graphical window, or by pressing F2 to select from a list of objects. Once the object is selected, you will be prompted with its property dialog. You will be able to change its properties and then update the object.

Move Objects

There is a sub menu allowing you to choose among different move object mechanisms. They include; either entering a transformation directly, transform from one frame to another, and using measurements directly from online instrument(s). A number of options are available for Moving Objects with instrument updates. Each option is detailed in the sections below.



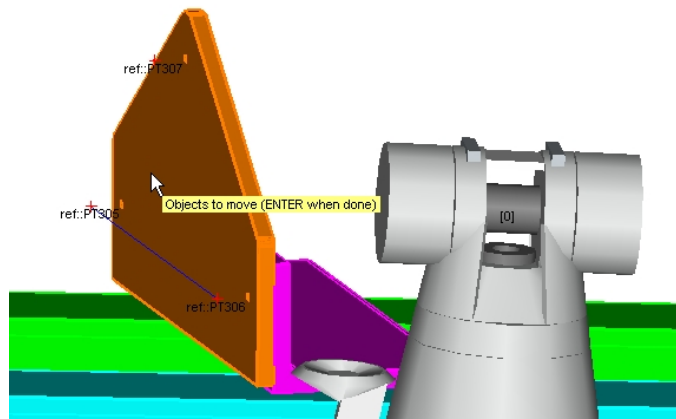
SpatialAnalyzer Move Objects Submenu

Enter Transformation

The first method for moving objects in **SpatialAnalyzer** is straightforward. **SpatialAnalyzer** prompts you to select the objects that you wish to apply the move to, as

shown in below. It then asks you to input the transformation parameters into the Move Objects Transformation Dialog shown below. You can see the effects immediately in the graphics view. **SpatialAnalyzer** lets you rotate, pan, and zoom the view during the process. Once you are satisfied, you can cancel or accept the results.

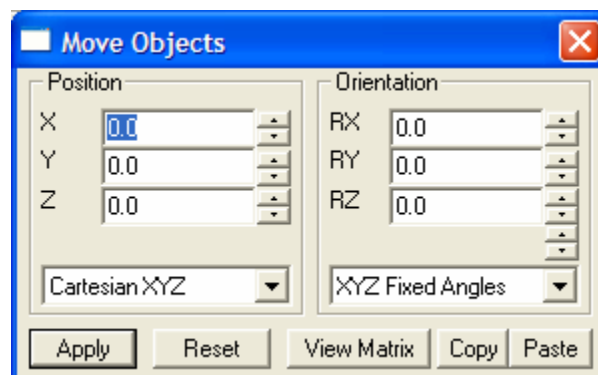
You can select the objects to be moved using a combination of selection methods. The selection methods include, double clicking (with the left-mouse button) on each object, trap-selecting objects (by holding a keyboard shift key and the left-mouse button down while dragging across the objects of interest), and by using the F2 object selection dialog.



SpatialAnalyzer Move Objects

Once selected, an object's color temporarily changes. The select object color depends on the current background color. If the background color is currently black, selected objects are temporarily shown as white. If the current background color is white, selected object's are temporarily shown as black. After selecting all of the objects that you want to apply the move to, hit the enter key to move to the next step.

In the case of an entered transformation, you will be given the transformation dialog and allowed to move the objects and observe the effects.




Move Objects Transformation Dialog

As you enter the transformation into the dialog, the objects will move in the graphics view. Please note that the transformations are always relative to the current Working Frame. The mechanics of the Move Objects Transformation Dialog are covered in greater detail in the Coordinate Frames and Spatial Transformations of this manual.

Drag Graphically

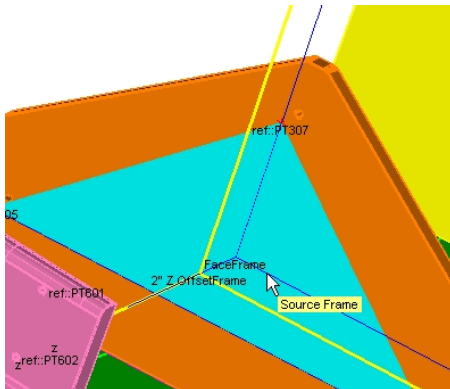
Moving objects graphically in SA is an efficient method to get object into an approximately location relative to one another.

The process allows you to select an object or objects and then move them with the mouse. Holding the left mouse button down while moving it translates the objects relative to the graphics view. Holding the right mouse button down while moving it rotates the objects; again relative the graphics viewport.

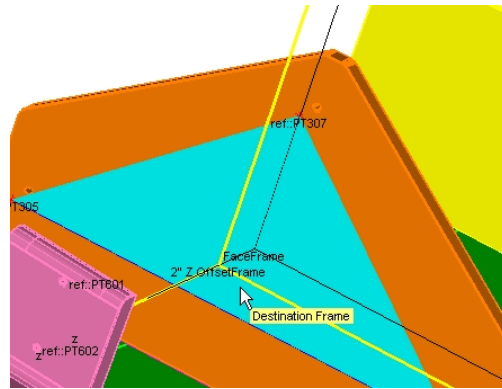
You can change viewports with the Viewport toolbar button  during this process. Changing the viewport makes orienting the objects in 3D easier. Select enter to end the process or the escape key to return the object(s) back to their original locations.

Frame to Frame Transform

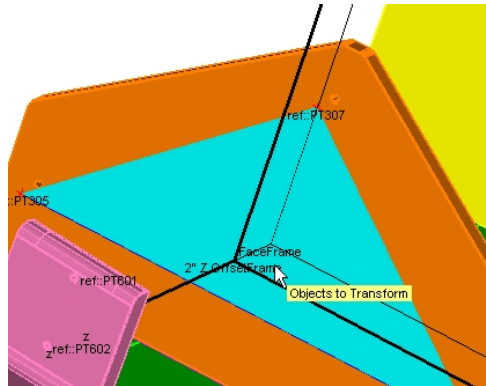
In the case of a frame-to-frame transformation, you will be asked to select two frames. First select, a source frame, and then a destination frame. The objects will be transformed by the relative difference between these two frames.



SpatialAnalyzer Select Source Frame



SpatialAnalyzer Select Destination Frame



SpatialAnalyzer Select Objects to Transform

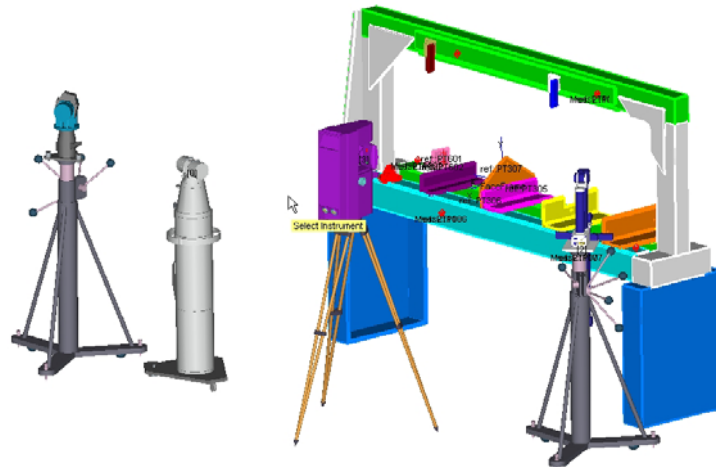
Selecting the objects to be moved is the next step. Hit the Enter key on the keyboard to conclude the selection step. The objects are then moved by the difference between the two frames. After the selected objects are moved, you will be asked if you wish to keep the results.

Translate Using Instrument Updates

Moving objects can be monitored in real-time using the Translate Using Instrument Updates function. This function will (real-time) translate an object or group of objects from the previously measured point to the current measured point position. It is by default a 3D movement.

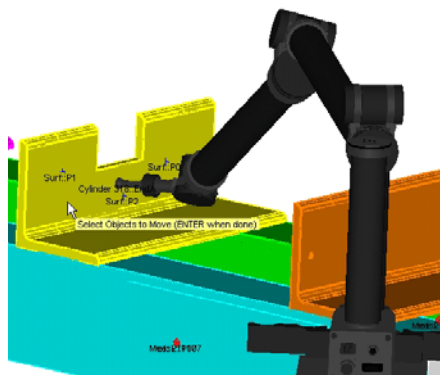
It can be used in a number of different applications. One application is Building components in an assembly. The process follows a step-by-step procedure to build one point at a time into their nominal locations.

The first step is to select the instrument that will be used to send the updates.



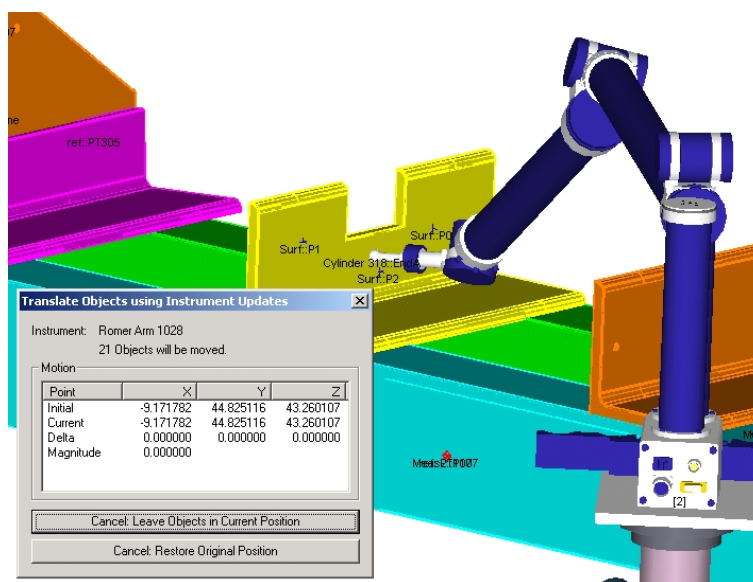
SpatialAnalyzer Select Instrument

Select the objects that are going to be moving.



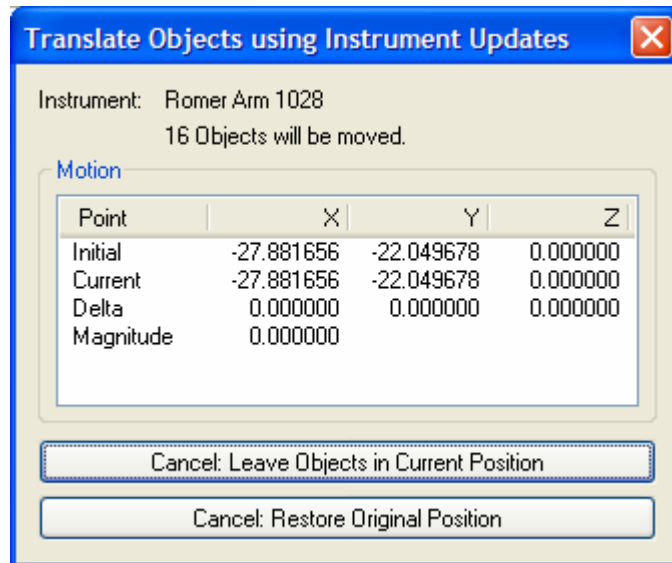
SpatialAnalyzer Select Objects

The system then starts polling the instrument interface and translates the graphical representation of the objects to the current measurement position. As you move the instrument, **SpatialAnalyzer** shows the graphical objects moving in real-time as shown below.



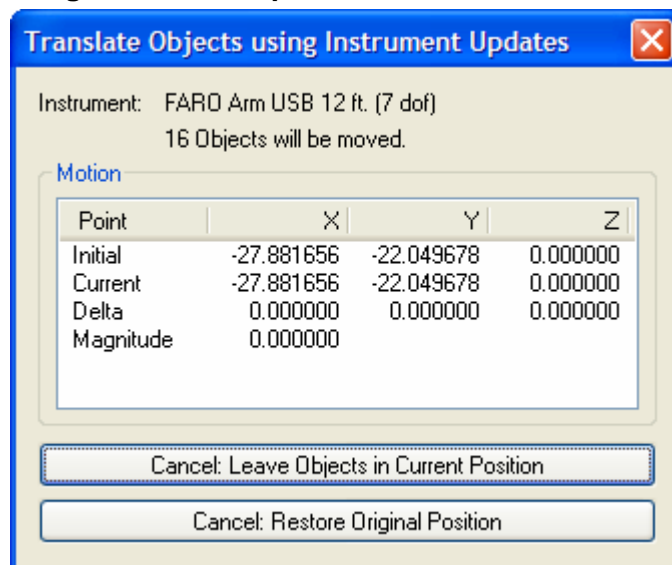
SpatialAnalyzer Translate Objects using Instrument Updates

An example process status and control is provided in a dialog shown below.



SpatialAnalyzer Translate Objects using Instrument Updates Dialog

Transform in 6D using Instrument Updates

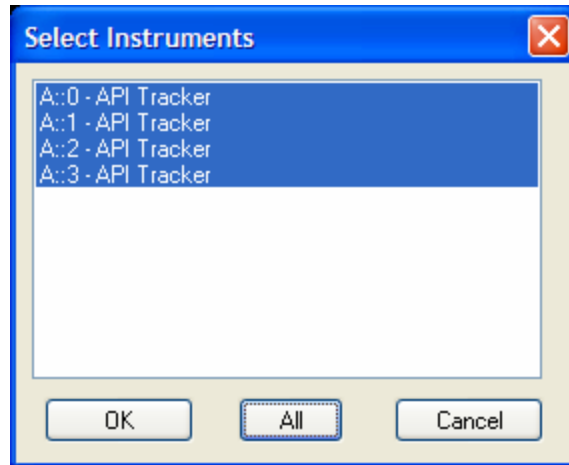


SpatialAnalyzer Translate Objects using Instrument Updates Dialog

Transform in 6D using Multiple Instrument Updates (Trans-Track)

For 3D instruments (e.g., Trackers) transforming objects in real time requires at least three on-line trackers. This functionality is also called Trans-Track. This function is provided as an optional module on your SpatialAnalyzer license.

To accomplish 6D tracking with multiple instruments select the instruments to use in the process (at least three are required.) The dialog below is an example of the instrument selection options.

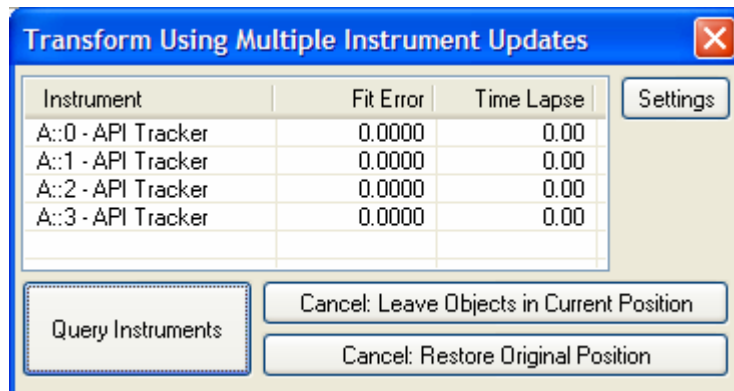


Select Instruments Dialog

Each instrument will monitor and update a point. The movements of the set of points are used together to update objects in 6D. Select one point per instrument. The point selection order must be consistent with the Instrument selection order e.g., select the points in the instrument sequence.

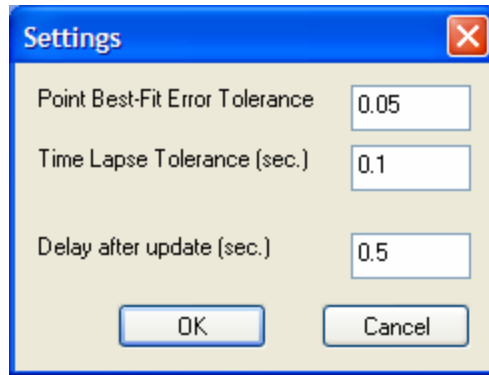
After defining the instruments and tracking points; select the objects that need to move with the point set. Select the object(s) with any standard SA selection mechanism.

The Trans-Track interface (shown below) indicates the instruments being monitored and the respective fit error for the point that it is tracking. The time lapse value indicates the time lapse between the measurements (i.e., synchronization error.)



Translate Objects in 6D using Multiple Instrument Updates Dialog

Trans-Track setting is accessible on the Settings dialog. Parameters of the best-fit threshold, time lapse tolerance and the delay after update are configured on this dialog.



Trans-Track Settings

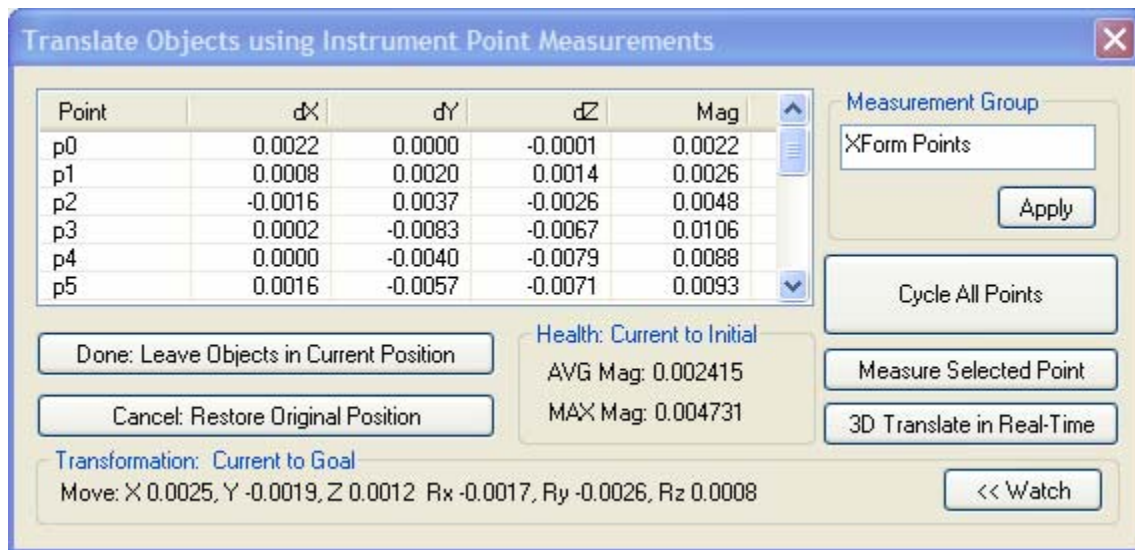
The Point Best-Fit Error Tolerance is used as a threshold to colorize Fit Errors that exceed the tolerance. It effectively flags the Instrument-Point pairings whose residuals are higher than the specified tolerance.

The Time Lapse Tolerance (sec.) defines the period of time during which all of the updates must be received. Only updates received during this period time are considered in the best fit solution to compute the new position and orientation. The Delay after Update (sec.) value is used in combination with the Time Lapse Tolerance parameter. The Delay is the period of time between update requests. Together they form the timing parameters for the Trans-Track process.

Guide Objects in 6D using Point Measurements

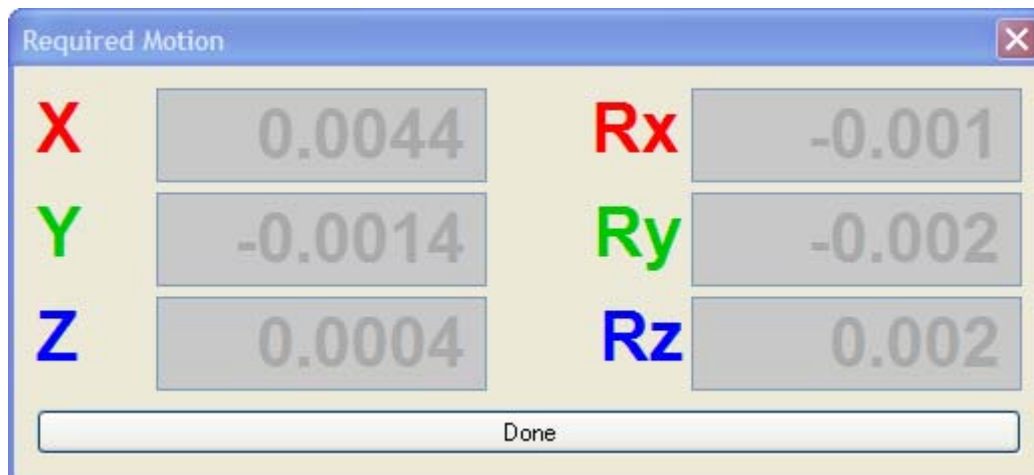
Guiding objects in 6D with a single 3D instrument requires a procedure where a sequence of at least three points is measured and then the objects are transformed to match the move by the point set. This process starts by selecting the instrument and the destination (goal) group that will be used as the reference. The next step is to select the Moving Group (attached to objects.) This group is repeatedly measured and compared against the destination (i.e., reference) group. The objects to move (which are attached to the moving group) are selected with any of the standard SA selection methods.

The dialog shown below is an example of the interface for the process.



Guide Object in 6D using Point Measurements Dialog

After the moving group point set is measured a best-fit transformation between the current and previous locations is solved. The transform between the prior and current point location is then applied to the selection objects. The delta between the two set can be watch dynamically change on the Watch window (shown below.)



Require Motion Watch Window Dialog

Transform by Minimizing Relationships

Relationships can be used for automatically updating dynamic changes between objects, make creating vector groups for analysis faster, provide a quick overview of the status of build operations and enable you to optimize your measurement directly against design constraints. This function is the interface for using relationships to optimize the differences between the objects in each relationship.

Relationships effectively provide a special type of spatial transformation capability. By simultaneously optimizing (i.e., best-fit) relationships i.e., the distance between (measured) points and known geometric shapes (e.g., points, lines, planes, and surfaces) the process fits measurements directly against their nominal features. The goal of the optimization is to find the spatial orientation of the measured points such that the relationship or group of relationships (i.e., distance between the points and their geometric shape) is minimized.

The on Relationships in the SA Users Manual details the process for transforming by minimizing relationships.

Copy Objects **Ctrl+C**

This menu item will allow you to copy multiple objects in the model. The copied objects each put in the active collection. After selecting the objects, you will be prompted to move the objects using the transformation dialog.

Remember that since groups of points are classified as objects, you can use this command to copy points. If they are measured targets, they will change their graphical depiction to represent points.

Mirror Objects

The mirror function works by multiplying a negative one to one of the components for each object you select to mirror. The mirror function is applied relative to an axial plane (e.g., X-Y, Y-Z, or Z-X) from a frame selected during the process.

When mirroring objects care needs to be taken concerning an object's (e.g., surface) normal direction. The mirroring process will generally flip the positive sense of an object.

Move

When moving objects with the mirror function copies are not made; however the normal directions for object can be inverted from their prior state.

Copy

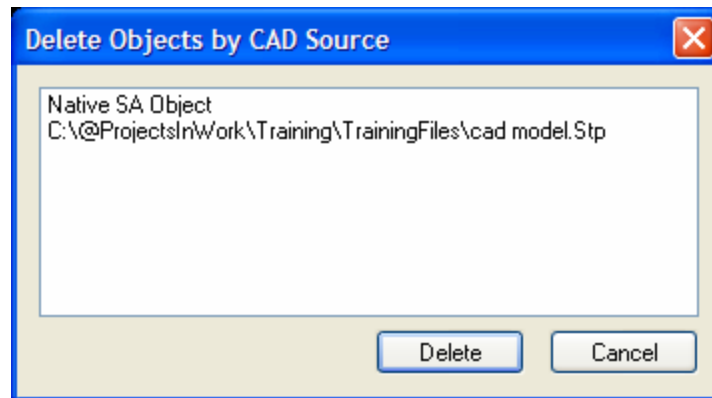
Copied objects are copied into the active collection. Care should be taken relative to the object's normal direction.

Delete Objects **Del**

This option will prompt you to select an object. After making your selection, it will prompt you for confirmation then delete the object.

Delete CAD Objects by Source

Cleaning up CAD elements after importing into a job file would be a challenging task. The function Delete CAD Objects by Source makes that process easier by providing a listing of each file import and the respective objects created. The dialog below shows an example of deleting CAD imported objects.

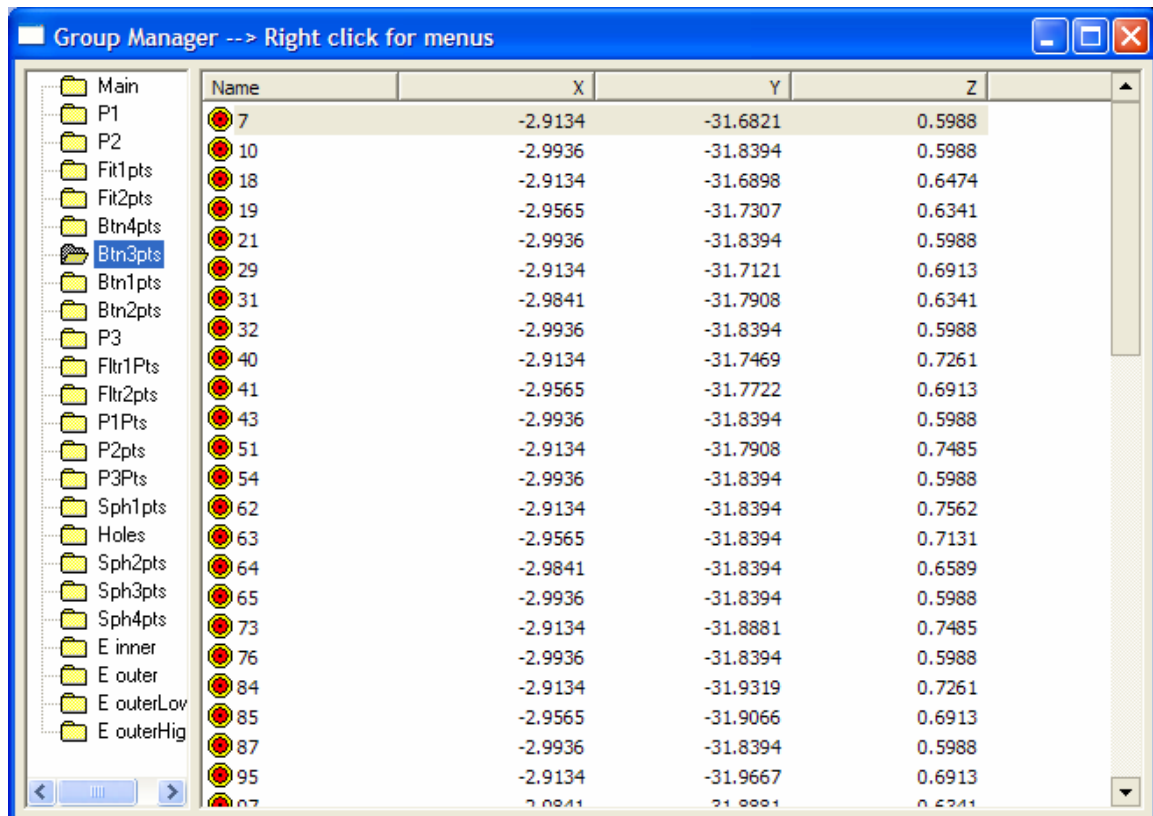


Require Motion Watch Window Dialog

To delete all of the entities that were created by importing the "cad model.Stp" file; select it and hit the Delete button. All of the Native SA objects can also be deleted with this function.

Group Manager...

The group manager option will display the group manager dialog shown below. This dialog may be used to move points between groups, copy points, delete points, rename points or create new groups.



Group Manager Dialog

Group Manager functionality is covered in detail in the on Group, Points and Targets.

Delete Points

After selecting this option, you will be prompted to select the points you wish to delete. After doing this and answering the confirmation, the points will be deleted. If the points are linked to measurements, the measurements will be removed as well.

Delete Cloud Points

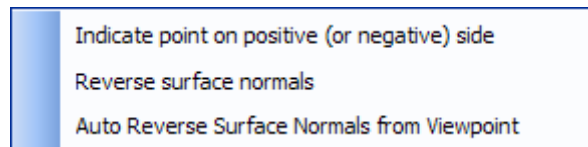
After selecting this option, you will be prompted to select the cloud points by graphically selecting them by holding the shift button down and drag the left mouse button to trap the points to delete. You can change views during the selection process without un-selecting the cloud point. After doing this and answering the confirmation, the points will be deleted.

Rename

You can either choose to rename a single point or a group of points.

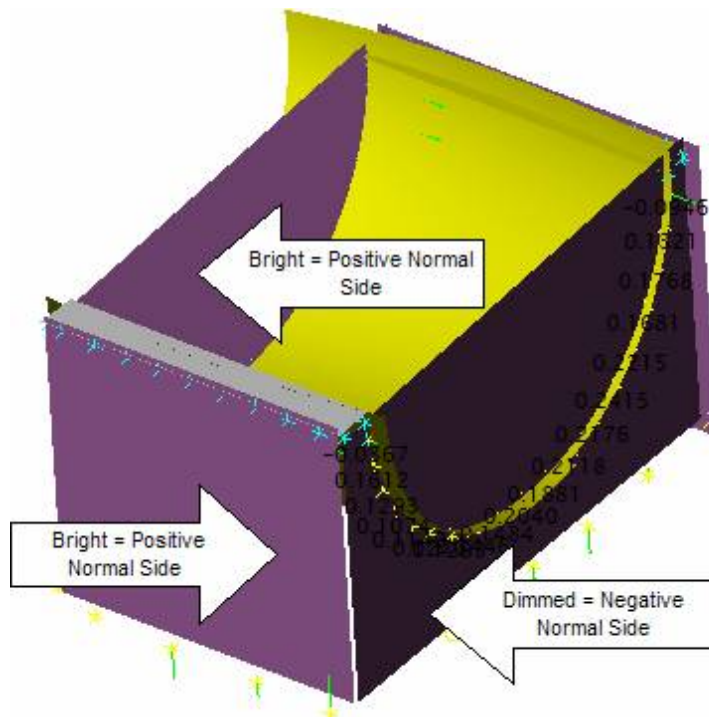
Surface Normal Conditioning

Use one of the Edit >> Surface Normal Conditioning functions to reverse the sequence (direction) of surfaces normals. The sub-menu of options is shown in the figure below.



Surface Normal Conditioning

SpatialAnalyzer shows the sense of the surface's normal direction by presenting the positive side of the surface as Bright and the negative side Dim. There is an example shown in the figure below.



Surface Back-Sides Dimmed

This behavior of showing the positive and negative sense of surface as bright and dim (respectively) can be turned on the User Options dialog. By turning off the Display option for showing Surface Back-Sides Dimmed all of the surfaces are then shown with the Bright color.

☒ Surface Back-Sides Dimmed

User Options >> Display Tab for Surface Back-Sides Dimmed

Indicate Point on Positive (or Negative) Side

This function allows you to select a point and then select the surfaces to condition. The next step is to indicate if the point lies on the positive or negative side of the surface. SA will then project the point on to each surface and if the projection indicates the point is on the alternate side then its surface normal is changed to be consistent with your request.

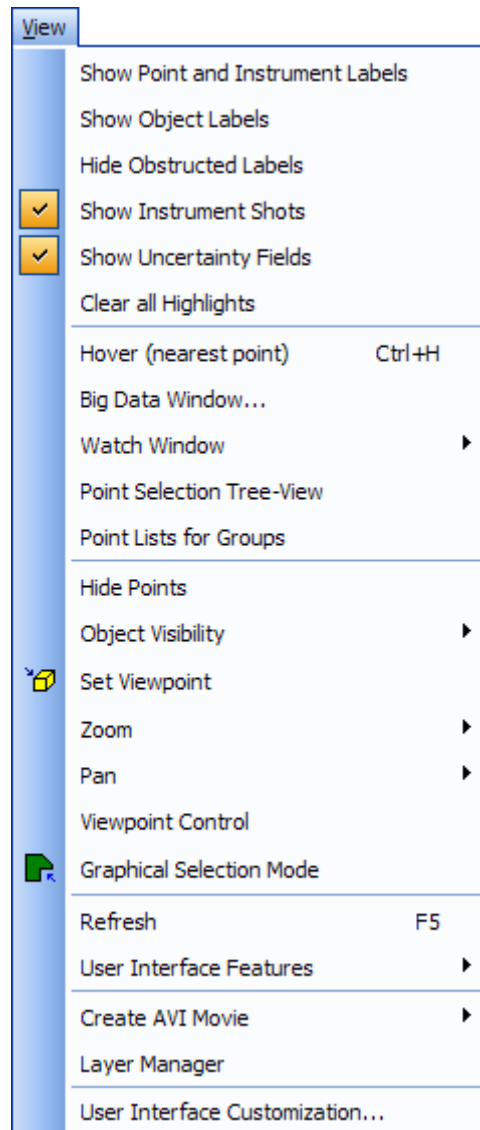
Reverse Surface Normal Direction

To reverse a number of surfaces at the same time select the Reverse Surface Normal Direction option from the Edit >> Surface Normal Conditioning menu. After selecting all of the surfaces that need to be changed SA reverses each selected surface's normal direction.

Calculator

This option spawns the Windows Calculator application.

View



View Menu

Show Labels

Control over the display of point and object labels in the graphics view is provided with the Show Labels toggle on the View menu. A check mark is shown on the left side of the menu, when the labels are visible.

Specific control relative to showing fully qualified object names (i.e., Group::Name) or just the point name is set on the Display tab of the User Options dialog. The User Option dialog is accessible from the main tool bar or you can select it from the Edit menu.

Show Instrument Shots

Toggling the visibility of last instrument shot in the graphics view is controlled using the Show Instrument Shots menu selection in the view menu. A check mark is shown on the left side of the menu when they are visible.

This display option can be set to either show all of the active shots or just the last shot by each instrument. **SpatialAnalyzer** defaults to only showing the last instrument shot. Control over showing all the shots or just the last shot is done on the Display tab in the User Option dialog.

Big Data Window...

This option will open the Big Data Window shown below. This window is used to display data as it is acquired from measurement devices. In addition to raw measurement data, this window will also display group comparison results.

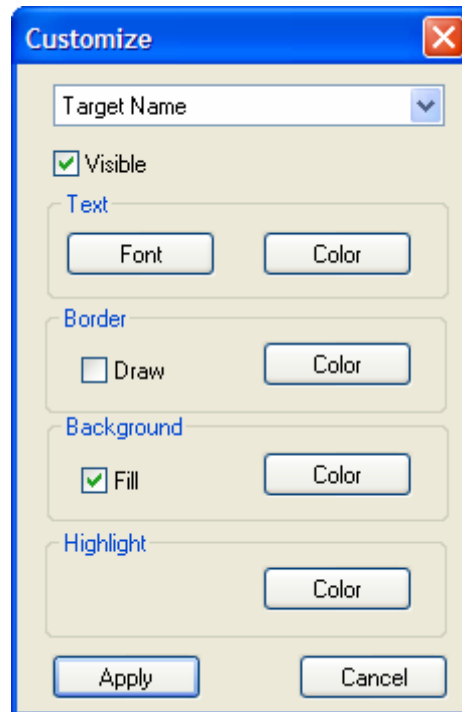


Big Data Window

This window is most often used with theodolite systems. It will display the pointing errors, worst apex angle, and other theodolite related information. It will also highlight and provide voice notification if the pointing error is out of tolerance (this tolerance is set in the user options area).

You can customize the contents, fonts, and sizing of the elements in the window. Select customize from the menu and the dialog shown below appears. Select the field you wish to customize from the combo box at the top of the dialog. Then set the options below and press Apply.

You can also resize the fields. Select ReSize from the menu. You will be able to move and resize the individual data fields by following the instructions under the “How to resize” item.



Big Data Customization Dialog

Watch Windows

This option allows you to create a series of watch windows. These windows read the live data coming from a selected instrument and compare it to a variety of geometrical features. Refer the titled Watch Windows for more information.

Point Selection Tree-View

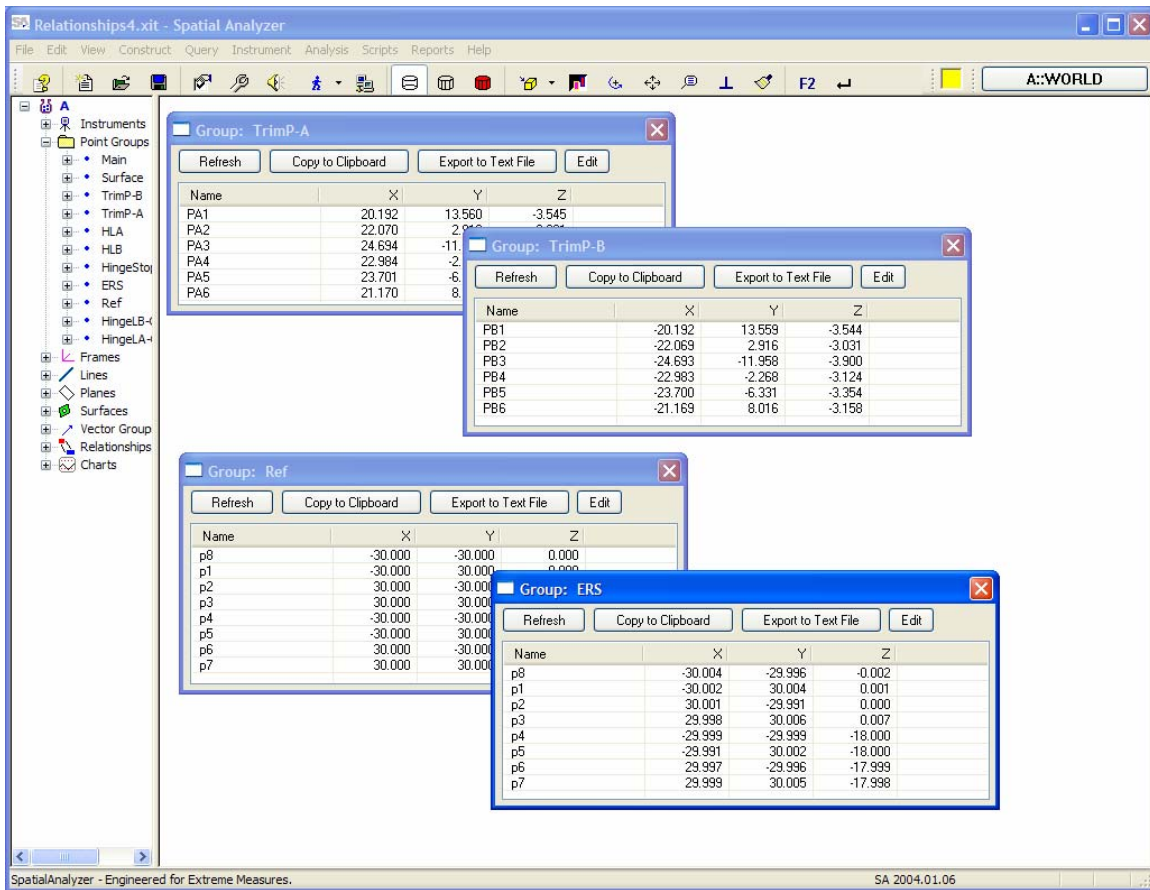
F2

If you are being prompted to select points, you can either select them graphically, or press F2 to select using a tree-view. This menu options provides another way of activating the tree-view.

In addition, you can double-click a target in the user interface at any time to display its properties. This menu option can be used to do this as well when point selection is not active.

Point Lists for Groups

Listing points within a group is accessible either from right-click menu on each group or from the View menu function called Point Lists for Groups. The figure below shows several groups of points, each listed in their own list control.

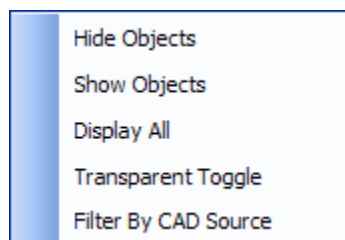


Point List for Groups

The Point List dialog has a convenient interface with the clipboard and exporting points to a file is accessible with the Export to Text File button. A point group editor is available by selecting the Edit button.

Object Visibility

This option contains a sub menu as shown in the figure below. These options will allow you to hide and show specific objects, or display all objects. In addition, there is an option that will switch an objects graphical representation to wireframe. This is particularly useful if you wish to “see through” an object to the objects inside.



Edit, Object Visibility Menu

Hide Objects

There are several methods for hiding objects in SA. The Hide Objects function allows you to select objects with any of the standard selection methods, including graphically, trapping, F2 selection tool and double clicking the items on the treeview. After selecting the object hit the Enter key to complete the process. Hitting the Escape key will cancel the function.

Show Objects

To show objects that were hidden you can use Show Object function and select the items with the F2 Selection Tool. Hit the Enter key to show the objects or the Escape key to cancel the operation.

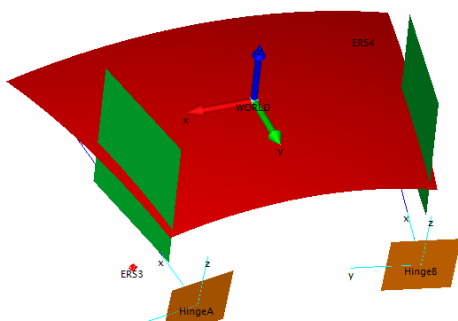
Another common method uses the right-click menu 'Show' option. Select the specific object (or any branch above the object) and then select the Show function.

Display All

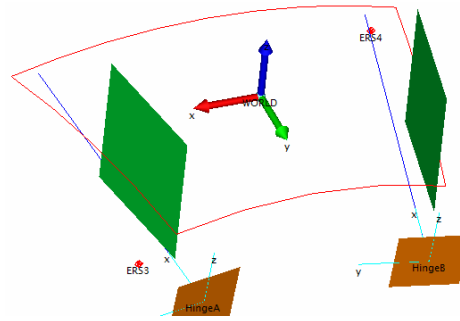
The Display All function works as its name indicates ... it turns the Show attribute on for each item in the SA database. All of the entities in the database will be visible.

Transparent Toggle

For objects that have shading properties the Transparent Toggle function will turn the shading function on and off. The figure (left) below shows the surface with its Transparent Toggle off (default.) The figure (right) shows the same surface with the Transparent Toggle in an on state.



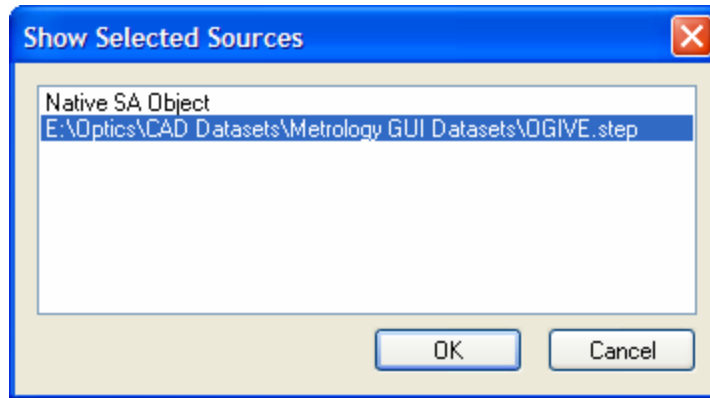
Transparent Toggle Off



Transparent Toggle On

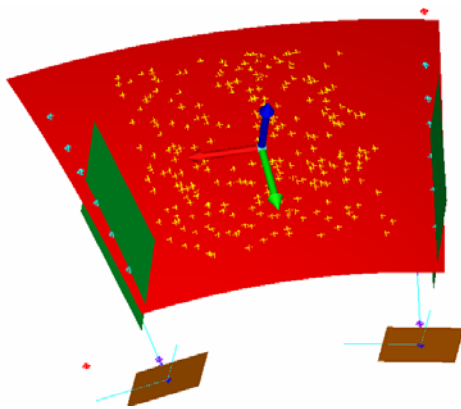
Filter By CAD Source

Turning the show attribute on or based on the source of the object is available with View >> Object Visibility >> Filter By CAD Source function. The example below the objects imported in the 'OGIVE.step' model will be visible while all of the Native SA Objects will be put in No Show state.

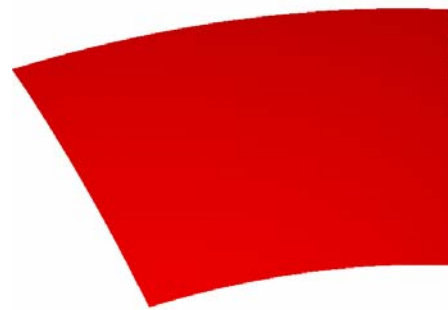


Show Selected Sources

The effect is shown in the two figures below. The figure on the left shows the model before filtering. On the right, is the model after filtering the Native SA Objects to a no show state. Note the dialog has you select the sources to show. You can select as many sources as you need to.



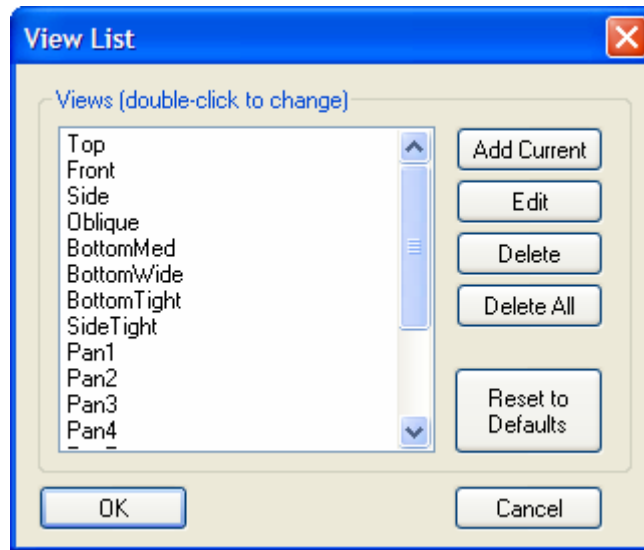
Before Filter by CAD Source



After Filter By CAD Source

Set View Point

The Set View Point function opens a dialog to create, set, and delete the views in the current SA job file. The dialog in the figure below shows an example of the views setup for a particular job.

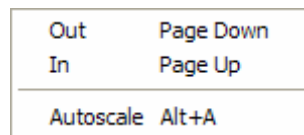


View List Dialog

To be able to return to a convenient view select the Add Current button and name the view. You can edit the name and whether or not you want the zoom settings restored with it by selecting the Edit option. Deleting view is also available along with the Reset to Default functions.

Zoom

This will access the zoom submenu shown below. You can zoom in or out as well as Autoscale. Also note the hotkey definitions that will allow you to perform these functions from the keyboard.

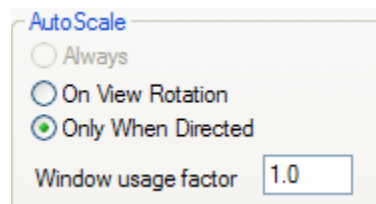


Zoom Sub-Menu

Autoscale

Alt+A

The autoscale function sets the viewport to just wide enough to see all the visible objects in the job. The exact scaling process can be more precisely controlled with the Window usage factor coefficient. The value is found on the Display tab of the User Option dialog. The group of setting related to this issue is shown in the figure below.



Autoscale group on Display Tab of User Options

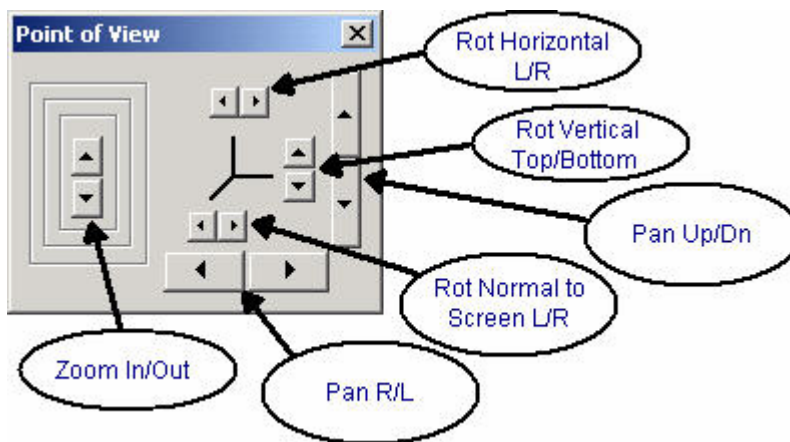
As an example; set the Window usage factor to 0.75 to have the maximum extents of the visible object use 75% of the graphics viewport.

Pan

The pan sub-menu contains options for up, down, left, and right. You can also use the arrow keys on the keyboard to perform these operations.

Viewport Control

Precisely controlling the view perspective is supported in the View menu using the Point of View Control dialog. Specific incremental adjustments (e.g., rotating 5 degrees about the X-axis⁹, etc.) can be made to the view using this dialog.



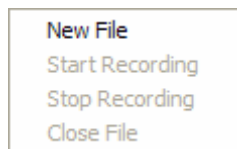
Point of View Control Dialog

Refresh


To force SA to refresh the graphics view and treeview controls use the Refresh function.

Create AVI Movie

Creating a AVI movie of your measurement process is available with the functions included in the Create AVI Movie submenu (shown in the figure below).



Create AVI Movie submenu

⁹ The translation and rotation increment is set in the User Options Dialog's , Units tab.

Setting up the AVI File and Video Capture Options

Select New File from the View >> Create AVI Movie submenu and enter a file name and location for the AVI movie file. Click OK to accept. The next step allows you to define the image capture options. You can choose between several screen areas, the output size, and the capture method. The playback rate defines the speed that movie plays. Care needs to be taken to understand how many frames is needed verse the running time of the movie.

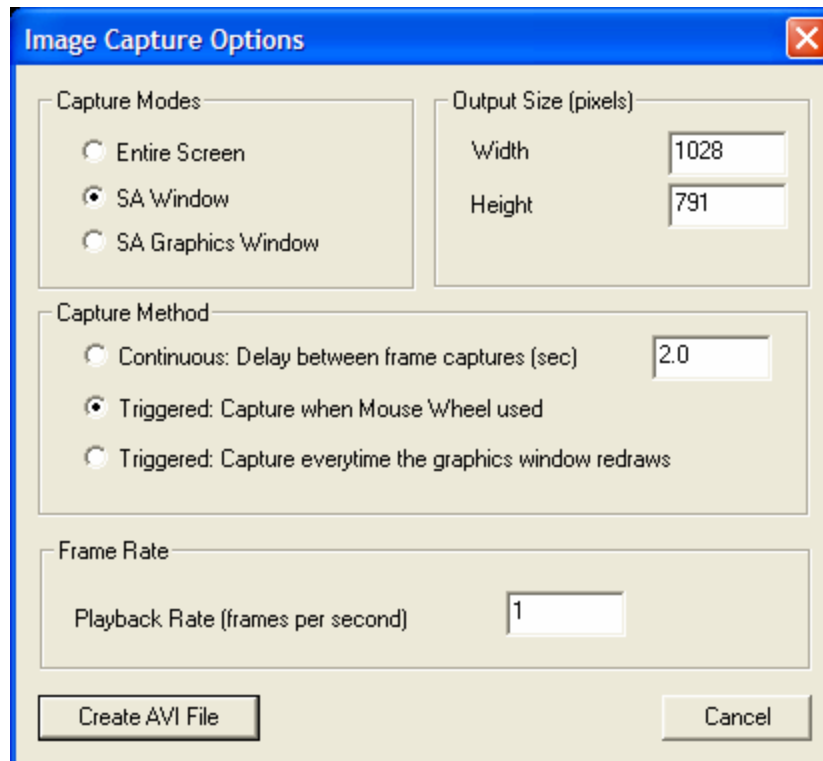
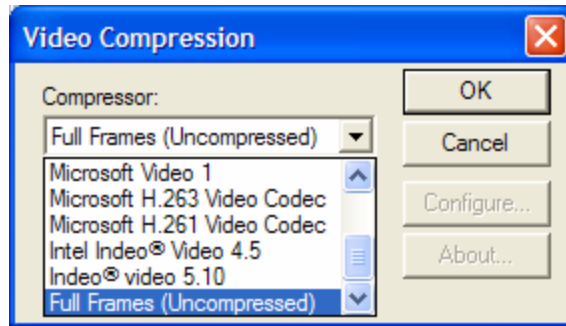


Image Capture Options for AVI Movies

Select Create AVI File when the image capture options are specified. The next step is to define the Video Compression Codec.¹⁰ The figure below shows the selection dialog. Select one among the choices of common codecs. If your computer does not have the codec installed it can be acquired over the internet or you can choose the default Full Frame (no compression.)

¹⁰ A codec is a scheme for coding and decoding large amounts of data. First the data are coded, then they are economical and uncomplicated to store. To view the coded data, they must first be decoded. (ref: <http://compsci.acad.emich.edu/>)



Video Compression Codec selection dialog

Start the AVI Movie Capture

Once the file and video capture setting are configured, start the capture process with the Start function on the submenu.

Stop the AVI Movie Capture

Stop or pause the AVI Movie capture with the Stop command.

Close the AVI Movie

Before you can play the AVI Movie in a player the file needs to be closed. Once file is closed the movie can be played in any compatible viewer.

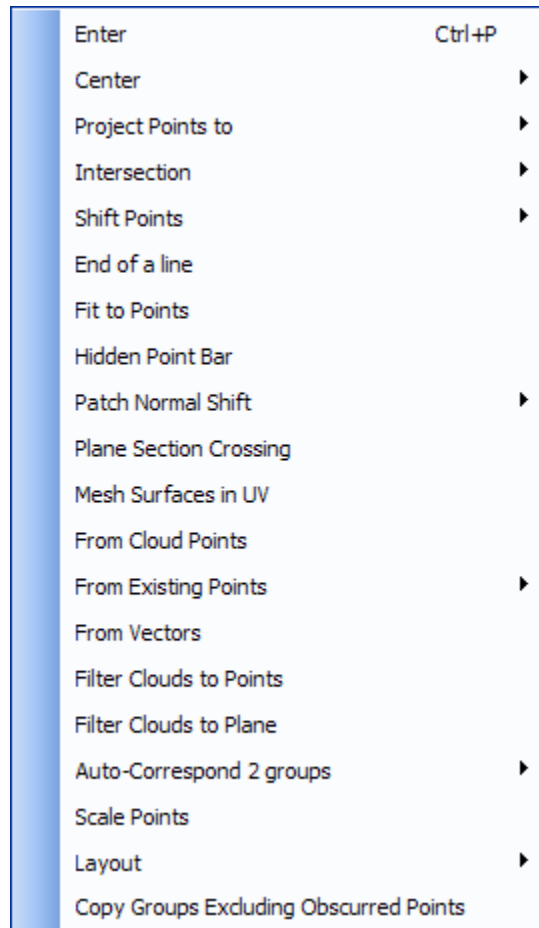
Construct

Construct	
Point(s)	▶
Frame	▶
Line	▶
Circle	▶
Plane	▶
Cylinders	▶
Spheres	▶
Paraboloid	▶
Cone	▶
Ellipse	▶
Re-Fit Geometry	▶
Curve	▶
Surface	▶
Polygonized Mesh	▶
Point Cloud	▶
Point Group	
Datums	▶
Perimeter	
Ellipsoid	▶
Boxes	▶
Vector Group	▶
Pyramid...	
Scale-Bar	▶

Construct Menu

Point(s)

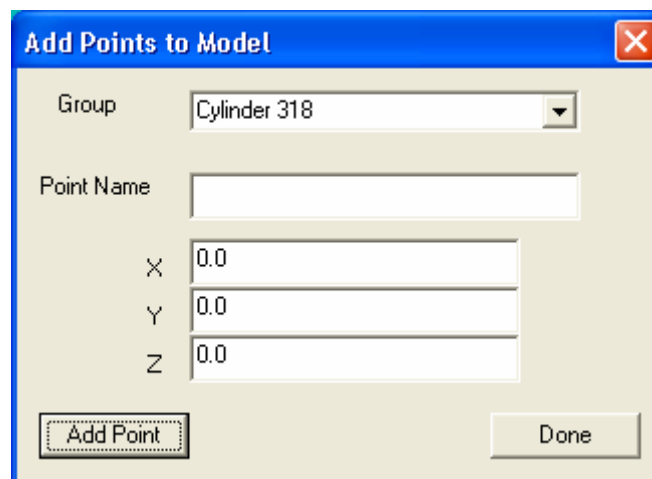
There are many options for constructing points. These are shown below.



Construct, Point Menu

Entering Coordinates

As with other types of geometries, you may create a point by entering its coordinates. To do this select Construct >> Point >> Enter. You will be prompted for a group and point name and coordinates in the Current Working Frame with the dialog shown below.



Add Points Dialog

Using this dialog a user may repetitively enter point coordinates without leaving the dialog.

Center Point Methods

A series of function that create points at the center of other objects can be a useful tool for metrology and analysis task. The descriptions below describe these functions.

Center of a Circle: You may create a point that is the center of a circle. To do this select the function from menu at Construct >> Point >> Center >> Circle. You are prompted to select the desired circle. Double click your left mouse button on the desired circle or select using the F2 function. Immediately, you will see a dialog showing you the coordinates of the center point in the Current Working Frame. Enter a point name for this new point.

Center of a Line: You may create a point that is the center of a line as well. To do this select Construct, Point, Center and Line. You will immediately be prompted to select the desired Line. Double click your left mouse button on the desired line or select using the F2 function. Immediately, you will see a dialog showing you the coordinates of the center point in the Current Working Frame. Enter a point name for this new point.

Center of a Sphere: You may create a point that is the center of a sphere. To use this function select Construct >> Point >> Center >> Sphere from the menus. You will immediately be prompted to select the desired sphere. Double click your left mouse button on the desired sphere or select using the F2 function. Immediately, you will see a dialog showing you the coordinates of the center point in the Current Working Frame. Enter a point name for this new point.

Center of a Cylinder: This function creates a series of points that are centered in the cylinder. There are two points at either end and another point at the midpoint. The points are labeled EndA, EndB and Center, respectively and are automatically put into a group with the same name as the cylinder.

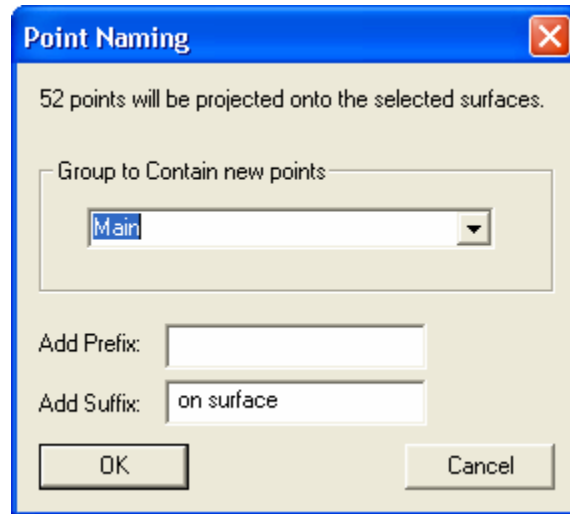
Point Projections Methods

A series of function that create points by projecting them to other objects can be a useful tool for metrology and analysis tasks. The descriptions below describe these functions.

Project Points to Surfaces Closest Point and to Curves: The functions projecting points to surfaces and curves follow the same procedure. The manual will describe the process for projecting points to surfaces. Please note that the same process is applicable for projecting points to curves.

Points are created by projecting them onto selected surfaces. The function searches the surface for the point that has minimum distance between each selected point and surfaces. The process is select Construct, Point, Project Points to, Surfaces, Closest Point. Select the desired surfaces that you want to project points too. Select the surfaces using the graphical techniques or with the F2 selection tool. Use the enter key to end this phase of the selection process. Select the points to project and press enter. Next a

Point Naming dialog is presented to allow the point naming to be controlled. Select or enter a new group name. You can add a point name prefix and/or suffix to each projected point. Note the prefix and suffix are optional.



Point Naming Dialog

The next step allows you to enter a Probe Offset value for each point. Each created point will be projected along a line between the initial point and the point on the surface by the amount entered into this dialog. A value of 0.0 means the created point will lie on the surface.



Probe Offset Value Dialog

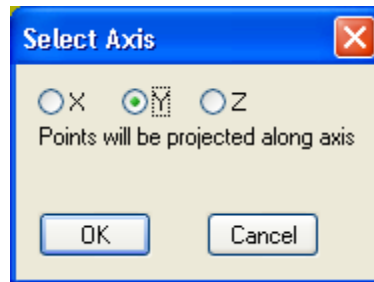
After the Probe Offset Value is entered the points are projected and created with the name and group attributes defined in the naming dialog.

Project Points to Surfaces Relative to Coordinate Axes Methods: The functions projecting points to surfaces relative to coordinate axes follow the same procedure. They differ in the type of axis used to define the direction and orientation of projection. The projection mechanisms are described in the sub-sections below.

The points are created by projecting them onto selected surfaces along a specific axial direction. If the projection intersects the surface more than once the function keeps the point closest to the initial point. Start the process by selecting Construct >> Point >> Project Points to >> Surfaces >> Relative to Coordinate Axes from the menu. Select the

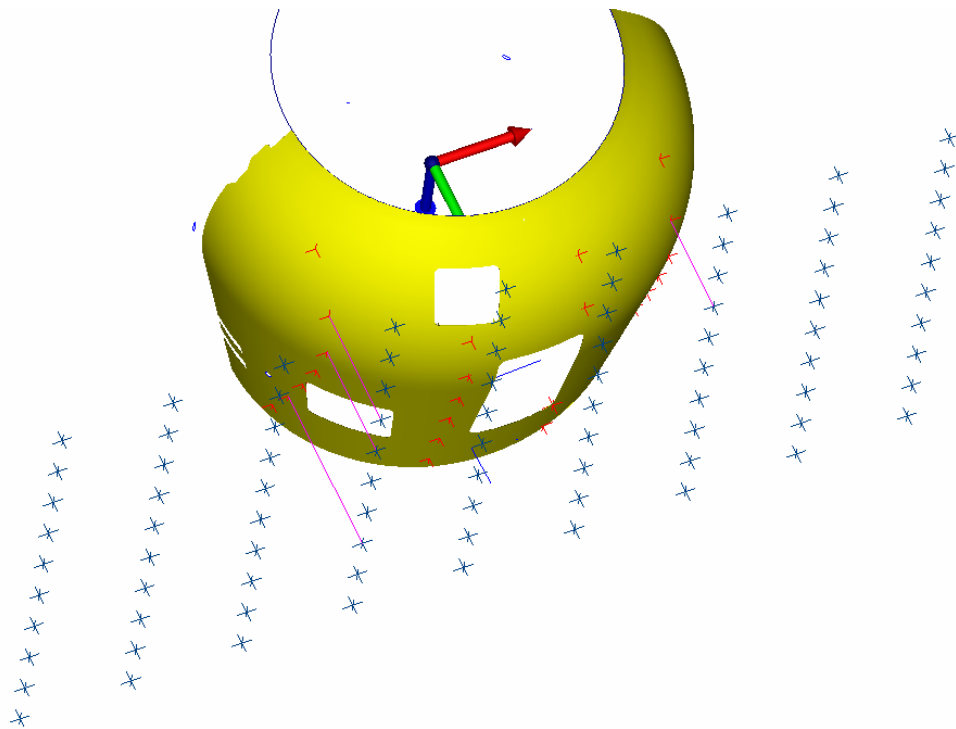
desired surfaces that you want to project points too. Use the enter key to end this phase of the selection process. Select the points to project and press enter.

Parallel to WCF Axis: The next step allows you to select the Axis that the points will be projected along. Each created point will be projected along a line parallel to the axis and passing through the selected surfaces. If the projection intersects the surfaces more than once, the point closest to the input point is created. If the projection does not intersect the surface then a point is not created.



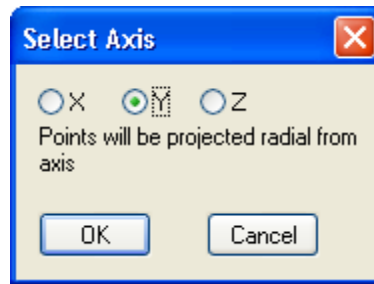
Dialog to select Parallel Projection Axis

The figure below show an example of points projected parallel to a current working frame axis. In this case the axis was the Y.



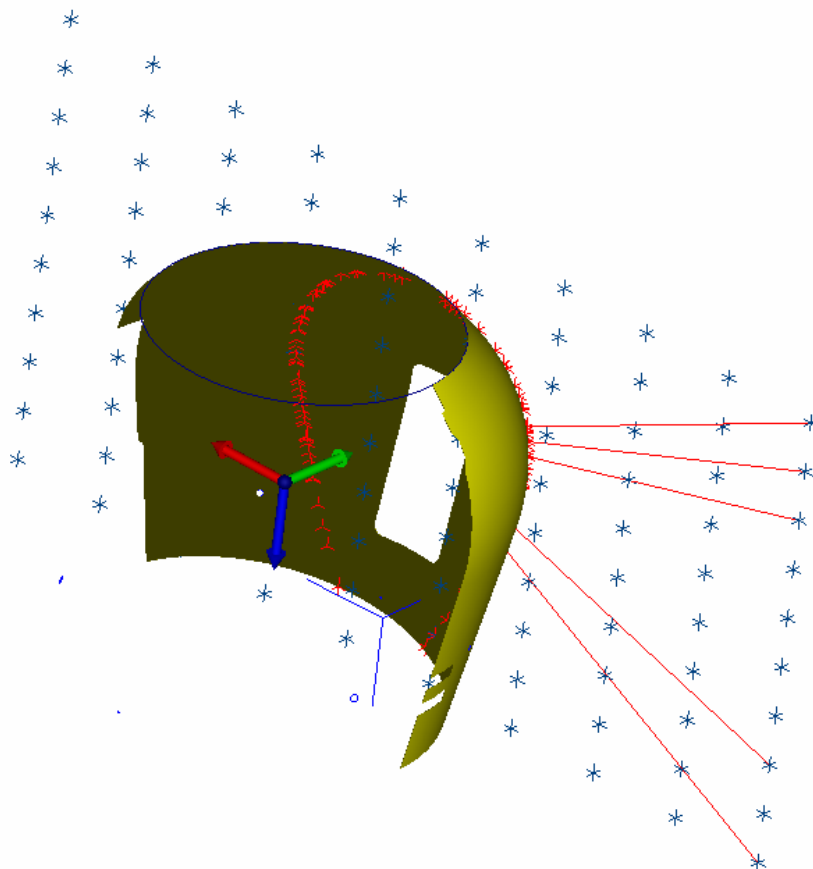
Points Projected onto Surfaces Parallel to Axis (e.g., Y-Axis)

Radial to WCF Axis: Select the axis that the points will be projected from radially. Each created point will be projected along a line extending radially from the selected axis and passing through the selected surfaces.



Dialog to select Radial Projection Axis

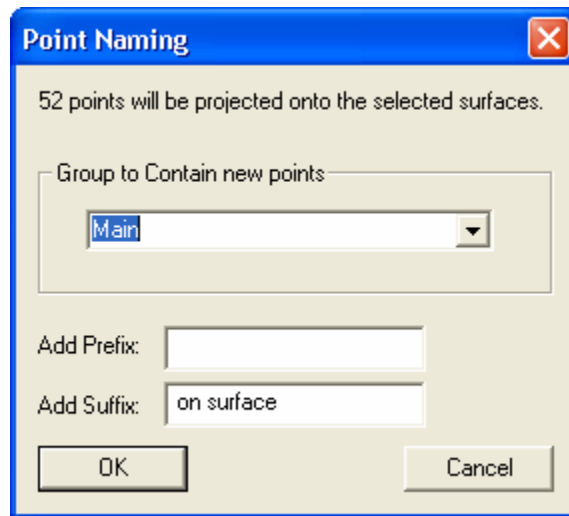
The figure below provides an example for this projection function. The created points lie at the intersection of a radial line that is normal to the selected axis (e.g., Y in this case) and the surface.



Points Projected to Surfaces Radial to WCF

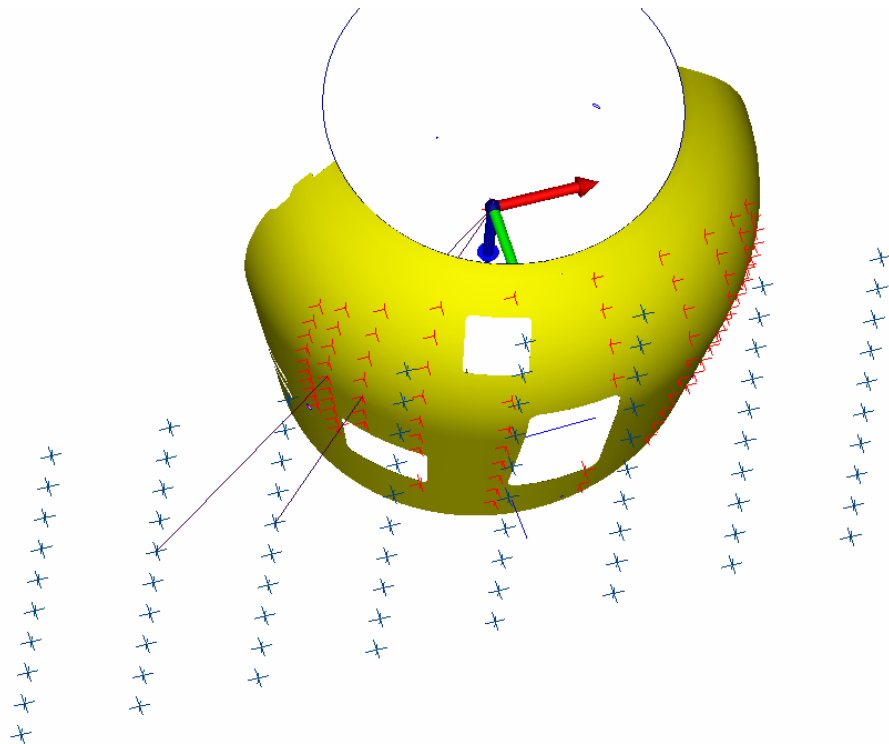
Polar from WCF Origin: A line is computed between each input point and the working frame origin. A point is created at the intersection between the line and surface(s). If the projection intersects the surfaces more than once, the point closest to the input point is created. If the projection does not intersect the surface then a point is not created.

Next a Point Naming dialog is presented to allow the point naming to be controlled. Select or enter a new group name. You can add a point name prefix and/or suffix to each projected point. Note the prefix and suffix are optional.



Point Naming Dialog

The figure below presents an example of the Polar projection function. The projected points lie on a line between the initial point and working frame origin. The point is created at the intersection of line and surface(s).



Points Projected along polar axes onto Surfaces

Projection of Points onto Objects

To create a point that is the projection of another point onto an Object, e.g., Planes, Circle, Lines, Spheres, Cylinders, Cones, etc. use the menu function Construct, Point, Projects Points to Objects. You are prompted to select the desired input points. Now you will be prompted to select the desired objects. A dialog showing the coordinates of the projected point in the Current Working Frame is present to allow you to control the point names. Note that for this projection method, the resulting point may not appear to lie on the object. The result may be off of the bounded graphical representation of the objects.

Intersection Point Construction Methods

The set of functions to create points at the intersection of objects follow a similar process.

Intersection of Three Planes: To create a point at the intersection of three planes select Construct, Point, Intersection, 3 Planes. You are initially prompted to select the three planes. Double click your left mouse button on the desired planes or select using the F2 function. A dialog showing you the coordinates of the point in the Current Working Frame is presented. Enter a point name and group for the computed point.

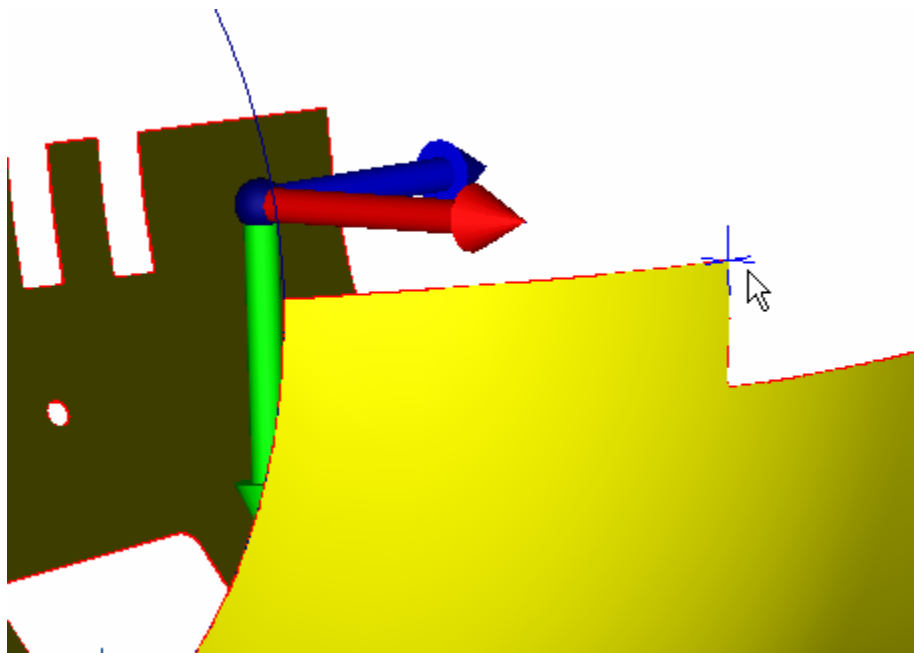
Intersection of Line and Plane: To create a point at the intersection of a line and plane select the menu option for Construct, Point, Intersection, Line and Plane. You are prompted to select the Line and then select the plane. A dialog showing the coordinates

of the intersection point in the Current Working Frame is presented. Enter a point name and group for the new point.

Intersection of 2 Lines (mutual perpendicular midpoint): To compute the point that lies at the intersection of 2 lines is done with the menu function Construct, Point, Intersection, 2 Lines. If the lines are not in the same plane or don't intersect the computed point will lie at the mid-point between the closest intersection of the lines. If the two lines are parallel the function will not create a point.

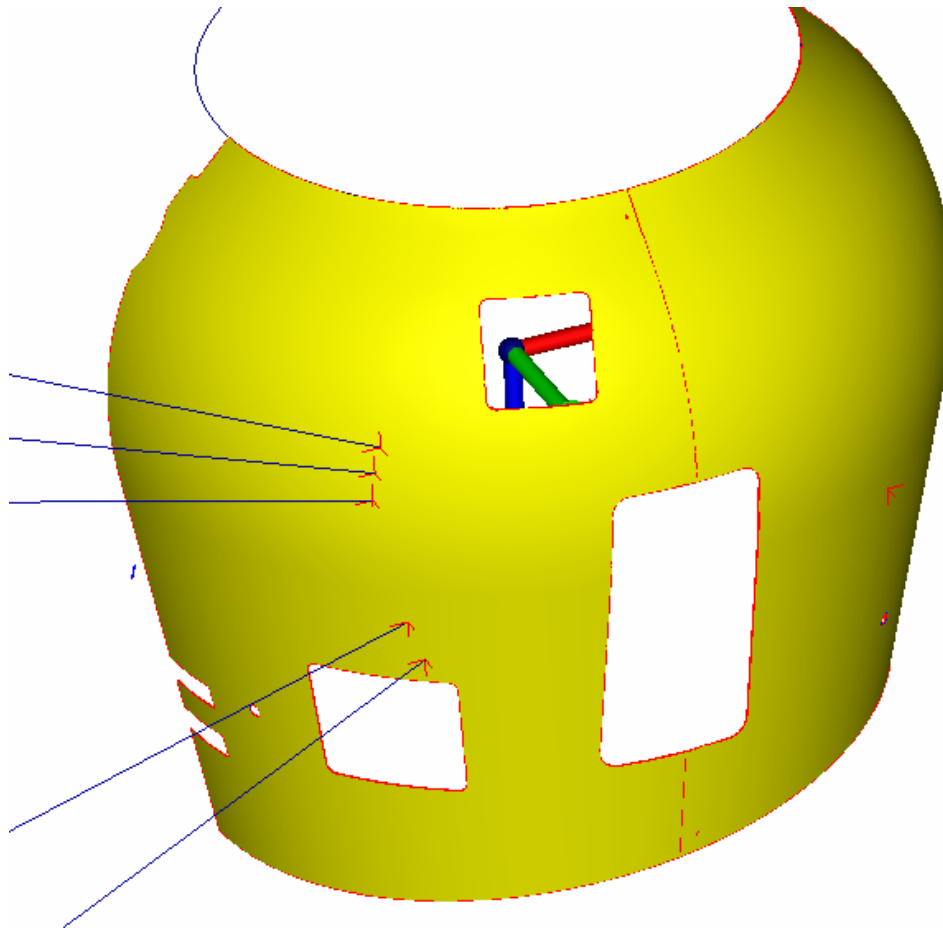
Intersection of 2 B-Splines: To compute the point that lies at the intersection of 2 B-Splines use the menu function Construct, Point, Intersection, 2 B-Splines. If the B-Splines don't intersect the computed point will lie at the mid-point between the two closest points on the respective B-Splines.

The figure below shows an example of a point created at the intersection of 2 B-Splines. In this case the point is at the corner of the part surface.



B-Spline Intersection Point Construction Method

Intersection at the Principle Object Axes Surface: To compute the intersection between objects and surfaces use the menu function Construct, Point, Intersection, Principle Object Axes Surface. All of the intersection points are created and put into a group that you define. An example of this function is shown below. In this example the lines were the principle objects and the points were created at the intersection of each of these objects and the selected surface(s).



Intersection Point at the Principle Object Axes Surface Method

B-Spline and Surfaces: To compute the intersection between B-Splines and surfaces use the menu function Construct, Point, Intersection, B-Splines and Surfaces. All of the intersection points are created and put into a group that you define. An example of this function is shown below. In this example the lines were the principle objects and the points were created at the intersection of each of these objects and the selected surface(s).

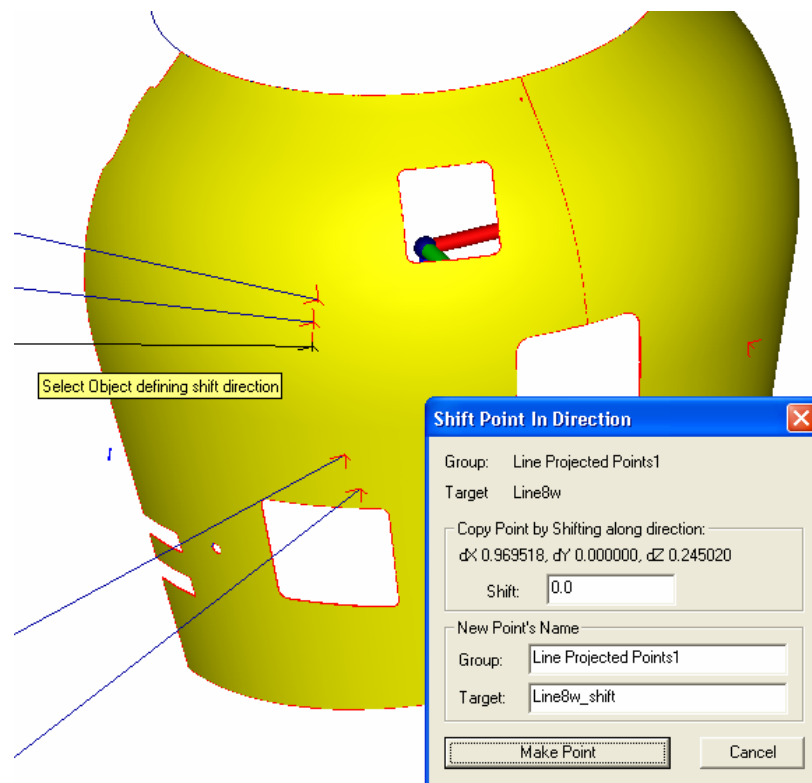
End of a Line

To create a point at the end of a line use the select Construct, Point, End of a Line function in the menu. You are prompted to select the line at close to the end that you want the point. Double click your left mouse button on the desired line or select using the F2 function. A dialog showing the coordinates of the end point in the Current Working Frame is presented to help you name the point and set its group name. Enter a point name for this new point and you're done.

Shift Point in a Direction

To shift a point in a particular direction the Construct >> Points >> Shift Point in a Direction function is helpful. Select the point to shift and then the object that defines the direction. Selecting a line will shift it along the line while a plane will shift it normal to

plane. The example below shows a point being shift down a line. After selecting the object (e.g., line) a dialog with inputs for the distance to shift and the point name and group is presented.



Shift a Point in a Direction Method

Fit to Points

To compute the centroid of a series of points use the Construct, Point, Fit to Points menu options. It prompts you to select the desired points. A dialog showing the computed coordinates of the mean point in the Current Working Frame is shown next. Enter a point name and group name for the computed point.

Hidden Point Bar

You may create a point using a hidden point bar. To do this select Construct >> Points >> Hidden point. You will be prompted to select Point A and Point B of the hidden point bar. You will be prompted to select the desired hidden point rod (in case you have several). After selecting the rod the new point will be added and you will be prompted for a point name.

Patch Normal Shift Construction Methods

The set of functions to create points by shifting along a patch normal follow a similar process. These functions work by computing a surface normal in a local area and then create points at an offset to the normal.

Single Point: This function starts by asking the user to select a point to offset. The next step asks for points that define the normal direction. After selecting these inputs a dialog to control the point name and the offset is presented. The figure below is an example of the input dialog.



Add Shifted Point along a Patch Normal dialog

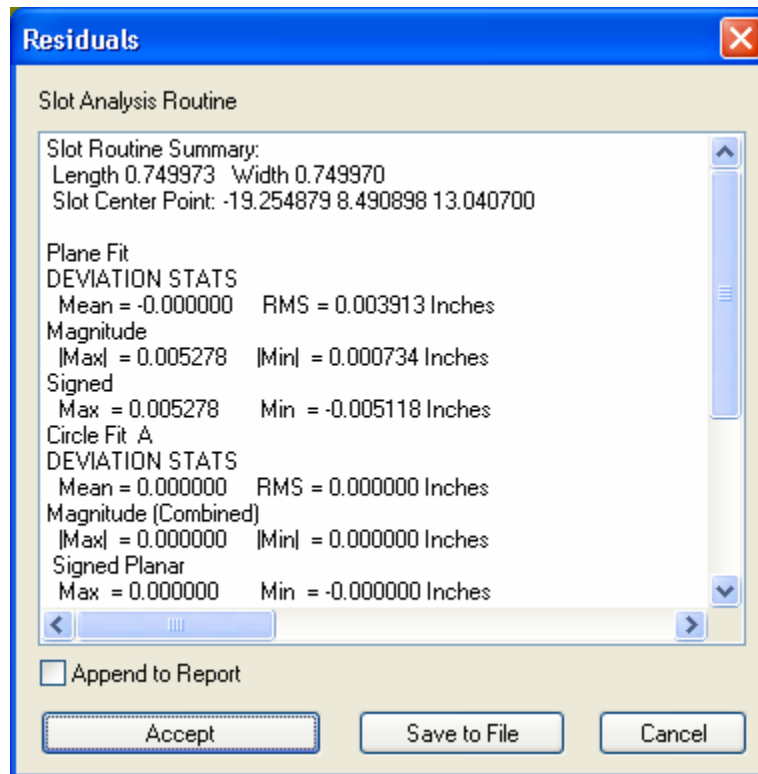
Enter the desired offset in the Target Offset field and any additional offset (e.g., material thickness) into the Additional Offset fields. The Group and Point fields left you name the point that will be created.

Hole Center: This function creates a point at the center of a hole at a specific offset from the surface. The function asks for two sets of points. The first set defines the circle perimeter and the second set defines the patch normal direction. The patch normal set of points can be the same set that defined the circle. The process computes a point at the center of the circle (from the first group of points) and at an offset along the plane normal (defined by the second set). The dialog to set the point and group name along with the offset amount is the same as the one used in the Single Point function (above).

When working with a hole in a surface as your starting point; one method to follow is to create the circle points is with the Construct >> Point >> Layout >> Object Vertices menu function. Select the B-Spline that defines the edge of the hole. If the B-Spline does not exist; create it with the Construct >> Curves >> B-Splines >> from Surfaces function.

Slot: Computing offset points from a slot involve defining the two circles for the slot and then the patch points. The circles are solved for by selecting points at their perimeter. The process asks first for points at the perimeter of CircleA and then for the points for CircleB. The next step asks for the points to define the normal direction. A plane is solved for from these points and then the plane's normal direction is used as the direction to offset the created offset slot points. A dialog allowing the user to set the point name and group along with the offset amount is presented. The created points include the centers of each slot circle offset by the amount defined in the dialog.

The computation presents a Residuals dialog to the user for review before the points are created. The residuals dialog shows a Slot Routine Summary with the slot length, width, and center point. Plane and two circle fit statistics are also posted in the Residuals dialog for review. If the computed characteristics don't match expected results the user can cancel the operation. If the results are within expected bounds then they are accepted with the Accept button.



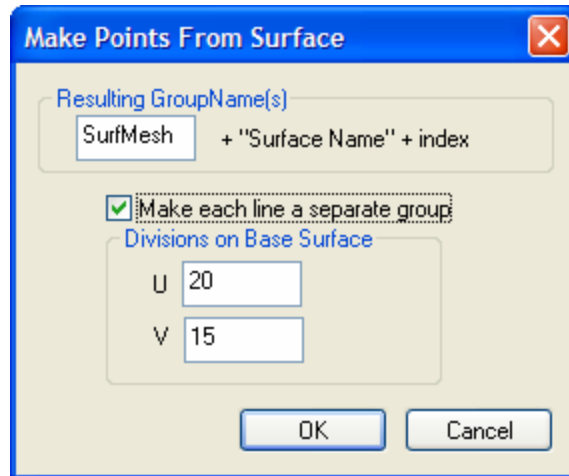
Slot Analysis Residuals and Summary dialog

Plane Section Crossing

Given a group of points and a plane the Plane Section Crossing function first computes a best-fit line from the group of points then solves for the point at the intersection of the line and the plane. The point is automatically named and put into a new group.

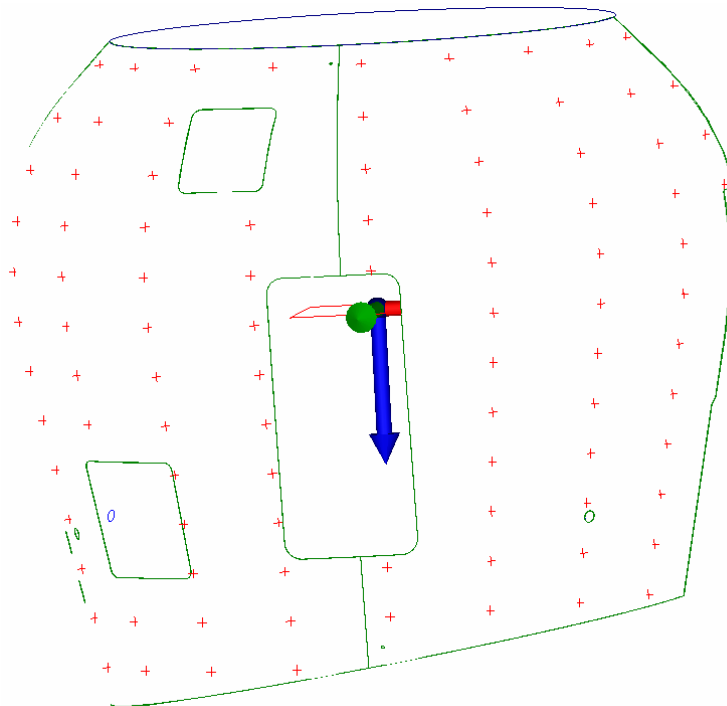
From a Surface Mesh

This option allows you to create points that lie on the vertices of a surface mesh. This function is useful for simulating data collection of a complex part. After selecting the surface(s) to create points on a dialog that defines the group naming and mesh density is presented (shown below).



Surface Mesh Point Creation dialog

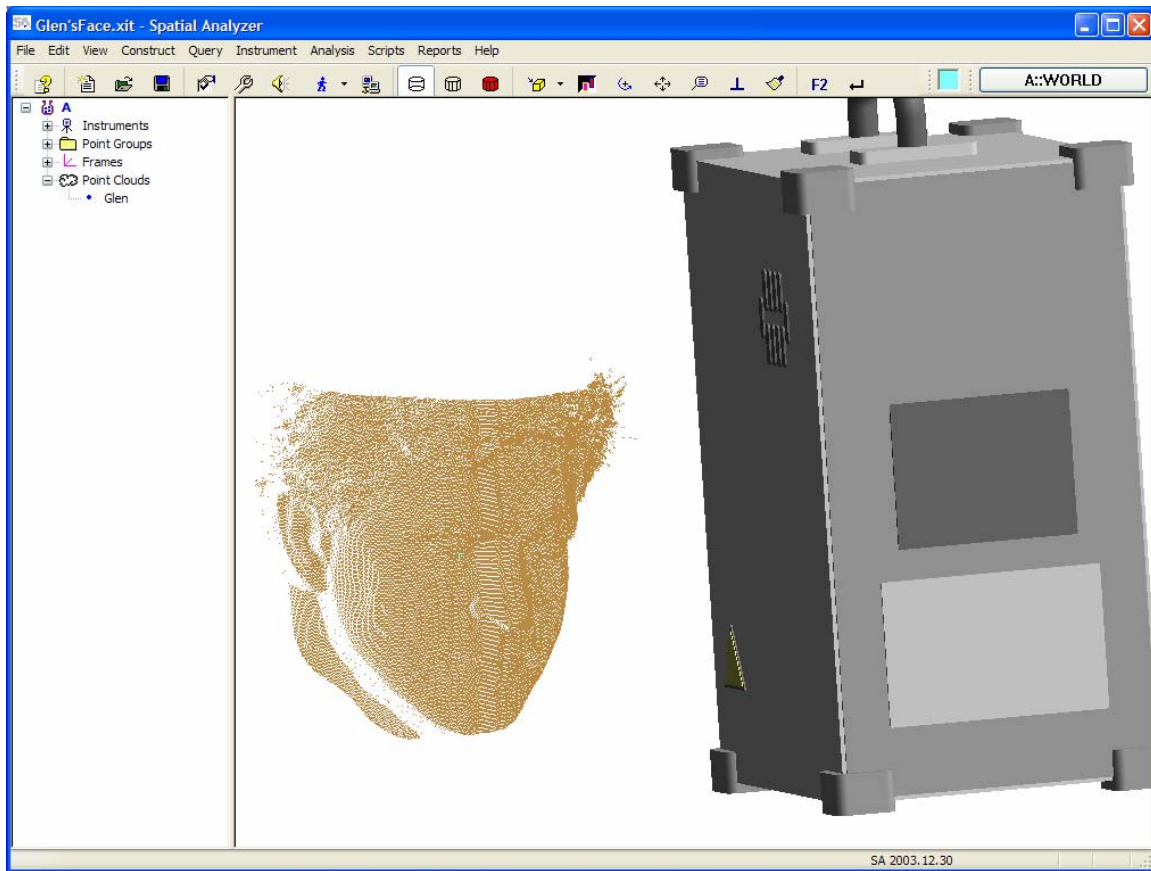
The points are created in a U-V based grid on the surface. By selecting the Make each line a separate group the points will be separated into groups along the U direction. The figure below shows an example of a grid network.



Surface Mesh Points

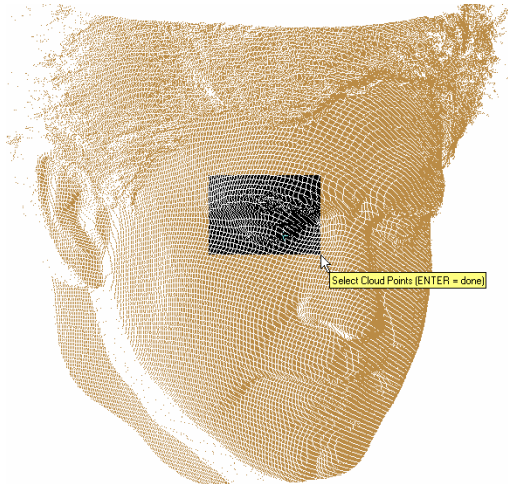
From Cloud Points

In 3D scanning applications there can be millions of points, to help you manage data sets of this type SA uses a special object type called point clouds. These objects make the process of manage scanner clouds of points much easier.

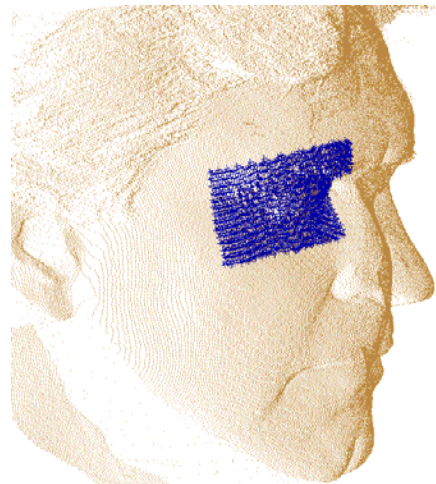


Point Cloud from Scanner in SA

Cloud points need to be converted to points in groups for a number of processes in SA (e.g., fitting to surfaces). The From Cloud Point function is one means of accomplishing that task. The process allows you to select cloud points to convert. The figure below shows a selected area.



**Selected Cloud Points to
convert to Points**



**Points in a group from
Cloud Points**

After selecting the area(s) of interest hit the Enter key. A dialog allowing you to name the point group and the point name prefix is the last step. The figures above show an example of the From Cloud Points function.

From Existing Points

The functions under the From Existing Points submenu are helpful with measurement jobs with a large number of groups. This group of functions lets you operate on points from different groups i.e., without regards to their group boundaries. For example 10 points from GroupA and 6 from GroupB can be moved in one operation without affecting other points in either group.

From Existing Points >> Copy: The copy function supports selection from any/all groups and then puts them into a new common group. When collecting data, sets of points are frequently put into separate groups. Using the copy From Existing Points function is a quick means of re-grouping a series of point sets. Subsequent operations on the points will then only need to be applied to this new grouping of the points.

The copy process has you select points from any of the groups. All of the selection methods are available including graphically double left-clicking points, trapping with the shift button down and dragging with left mouse button, F2 selection tool, and selecting on the treeview by double clicking on the points. After selecting the points to copy; hit the enter key to end the selection process. A dialog allows you to name to “New” group. After naming the group copies of the selected points appear on the treeview under the “New” group.

As discussed previously copied targets do not copy their measurements so the copies of targets are points without measurements.

From Existing Points >> Move: Moving a set of points from a series of groups into a new group can be accomplished with the Move From Existing Points function. The process follows the same one used in the copy function. The results are different because the

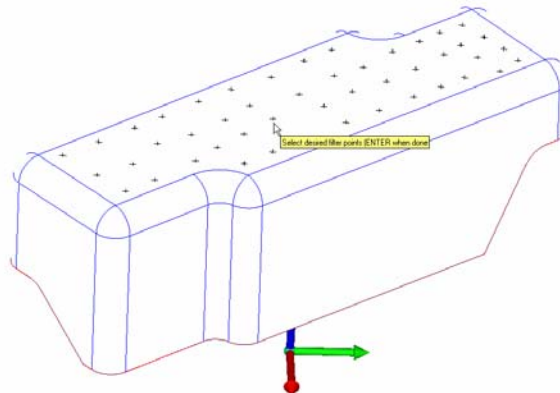
points are removed from the source group(s). This function also does not preserve measurements during the move, i.e., targets are converted to points without measurements in the “New” group.

From Existing Points >> Subset with Greatest Spacing: Sampling a set of points down to those points with the greatest point to point spacing is supported with the Subset with Greatest Spacing function. The first step is to select the points to filter. Hit the enter key when your selection is complete. A dialog then asks you how many points from the selected points are needed in the new group. After computing the filtered group of points a dialog asking you to name the “Spaced Points” is presented. The points in the subset group are copies from source group(s).

Filter Clouds to Points

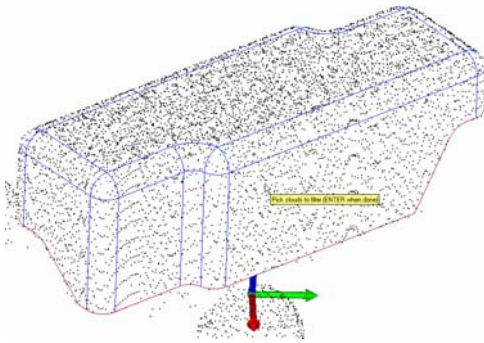
Frequently point cloud data needs to be filtered down to points that relate to specific features (e.g., plane, circle/hole, and surface.) One means of accomplishing feature base point extraction is the Filter Clouds to Points function.

The first step has you select a set of points that will be used as the filter points for the filtering process. For extracting a plane from point clouds select a series of point's through-out on the plane. The figure below shows a series of points on top of the CAD model of the part.



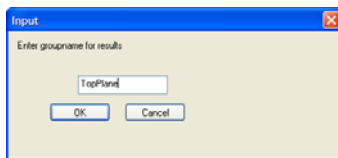
Filter Seed Points to extract Top Plane points from cloud(s) of points

Next select the point cloud(s) to filter points from. The figure below shows a point cloud of 314,000 points. Using the filter points selected in the first step the cloud of 314,000 points will be filtered down to a point group that defines the top plane.

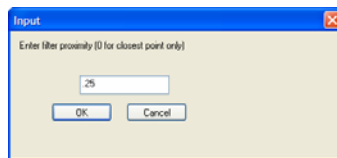


Cloud(s) of points to be filtered

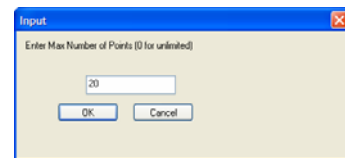
The next three inputs are for the name of the filtered point group, the proximity distance around each filter point, and the maximum number of points to extract around each filter point. The figures below provide an example for these inputs.



**Group name
dialog for
Filtered
Points**

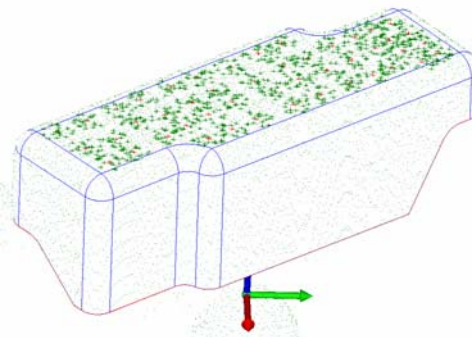


**Proximity
distance
dialog**



**Max number
of points
to filter
from
within
proximity
distance**

Cloud points that are within the radius defined by the proximity distance around each filter point are automatically selected and then converted to regular points. The figure below shows the results.



Filtered points from the cloud

Options:

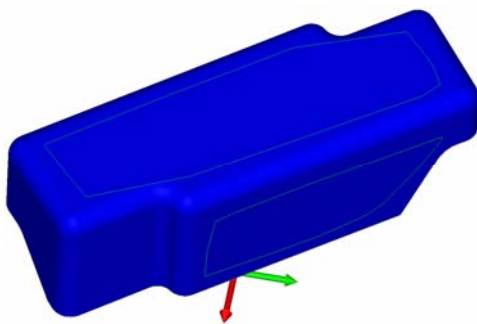
If you only want the closest points around each filter point then set the proximity quantity to zero. The routine will then search the clouds for only the closest points.

If you want the 5 closest points then set the proximity to zero and the Max Number of Points quantity to 5.

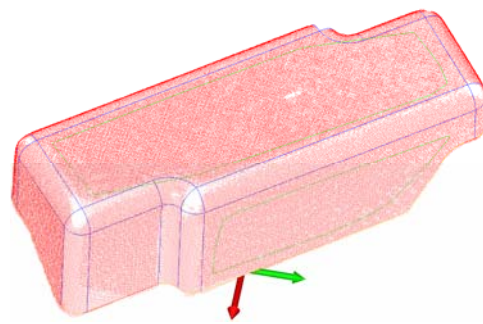
If you want all of the points within a specific radius set the proximity to the radius quantity and leave the Max Number of Points set to 0.

Filter Clouds to Plane

Extracting planar points from a cloud is a common process when working with 3D scanner data. The Filter Clouds to Plane function is one method to extract planar points from a cloud.

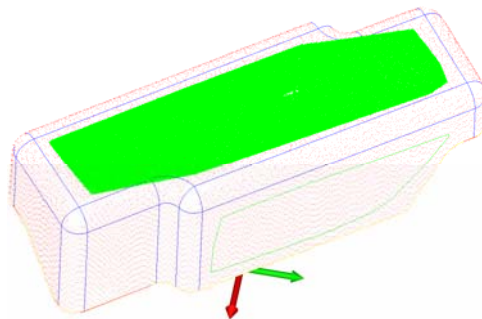


**Model with plane used to
Filter Cloud Points**



**Cloud of (~230,000) points
to filter**

This function asks for a filter plane and then the clouds to extract the planar points from. A proximity quantity is used to specify the maximum distance cloud points can be from the plane.



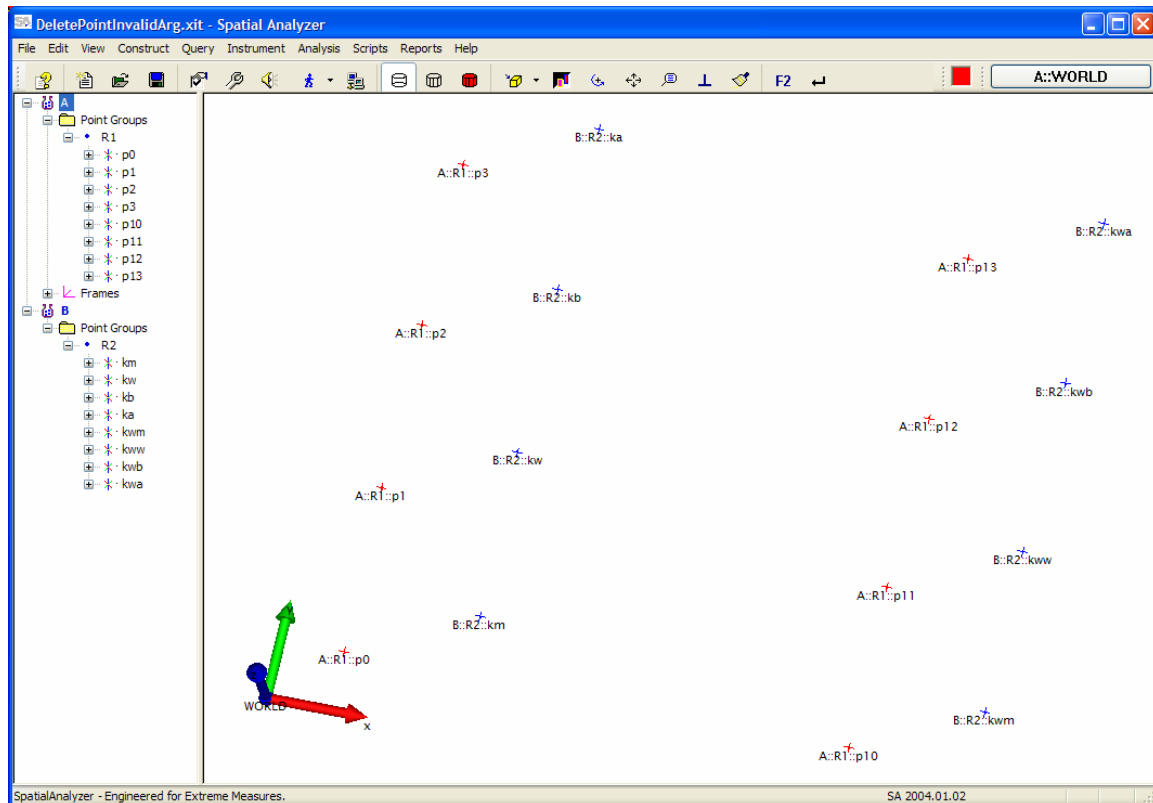
Filtered (~69,000) points to plane boundaries and proximity tolerance

The figure above shows the filter plane and then the cloud of points from which the planar points will be filtered from.

Auto-Correspond Two Groups

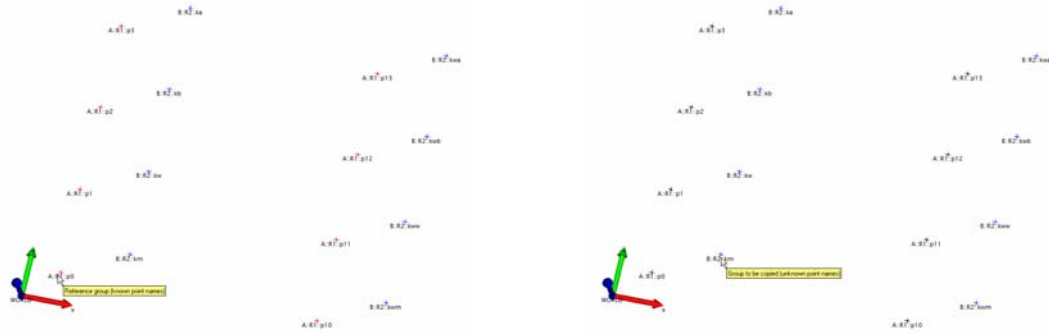
Getting points within two groups to have consistent names is an important step for transforming and reporting tasks. You can use the Auto-Correspond Two Groups function to rename the points within a group to match each respective point in a reference group. The function relies on unique inter-point distances to identify point name correspondence.

The figure below shows two sets of points that clearly correspond but do not have matching point names. The first step in the process is to choose the reference point group. In this case the group R1 in collection A is used as the reference group.



Point groups with names that do not correlate

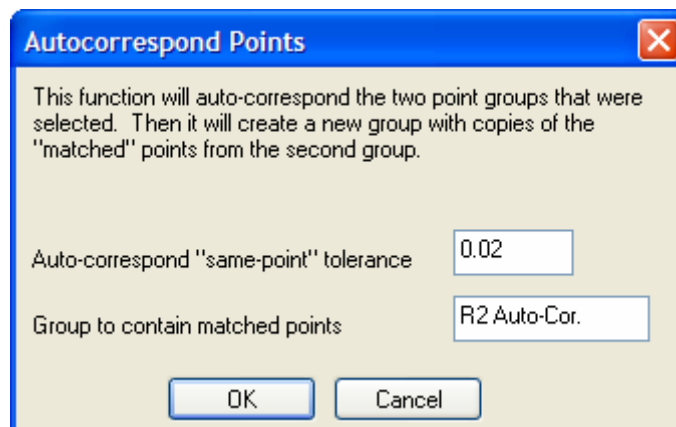
The figures below show the selection process for selecting the reference group and then the group that will be copied and then have its points renamed to match the references.



Select the group that will
serve as set of
Reference point names

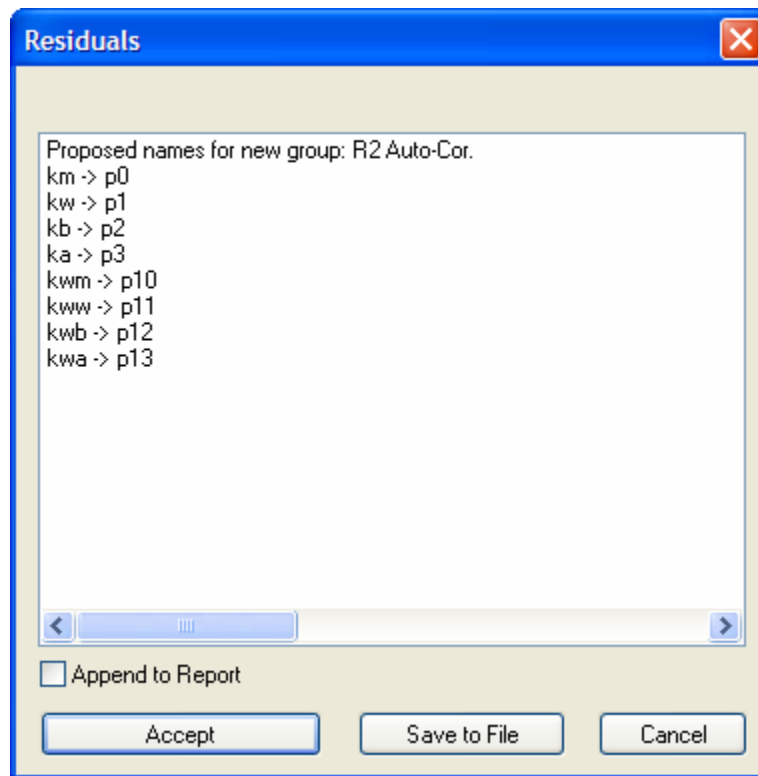
Select the group to be
copied and renamed

After selecting the two groups an input dialog that asks for the Auto-correspond same-point tolerance and the new group name is presented. The tolerance is used to isolate unique inter-point distances. Selection of the smallest tolerance that will uniquely set point to point distances is suggested. For most applications using roughly twice an instruments precision is a good initial estimate for the tolerance.



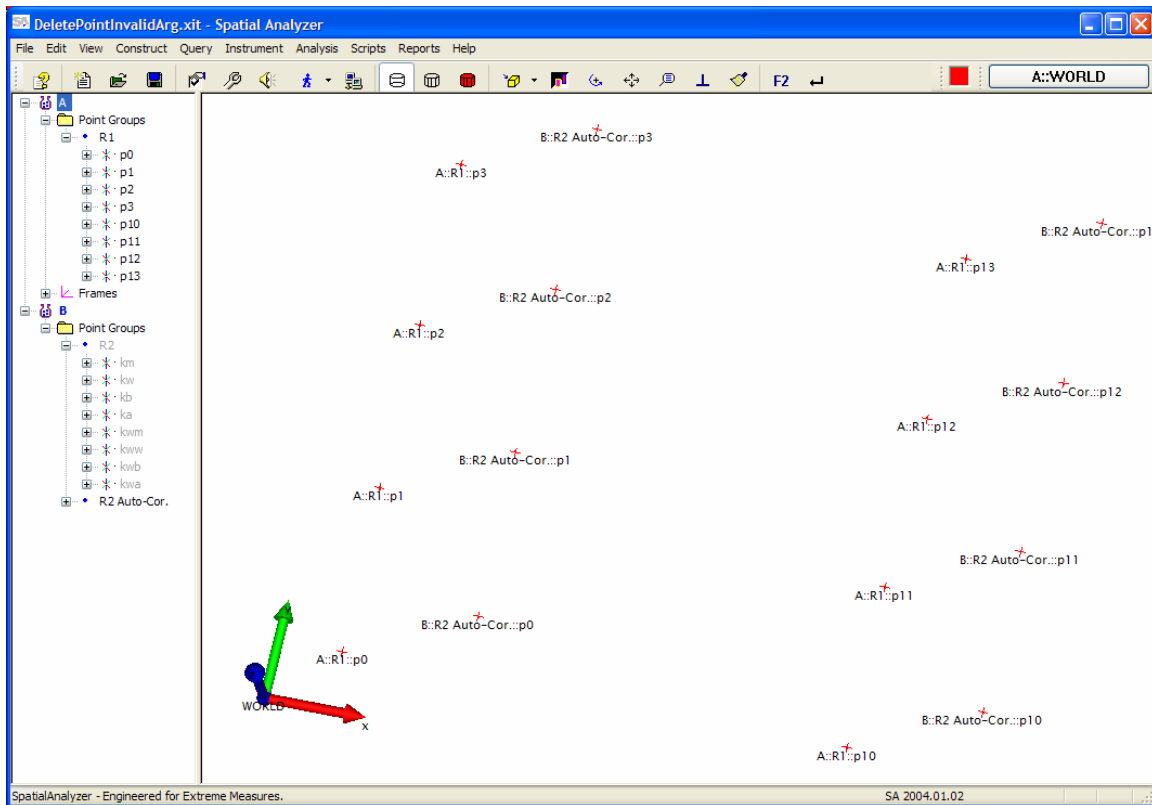
Auto-Correlation Input Parameter Dialog

A residuals dialog with the proposed renaming map shows each point. For those that a match was found a corresponding point is shown. If a match was found the original point name is retained in the new group. The figure below shows the proposed names for the points in the new group.



Auto-Correlation Residuals dialog showing proposed Renaming Map

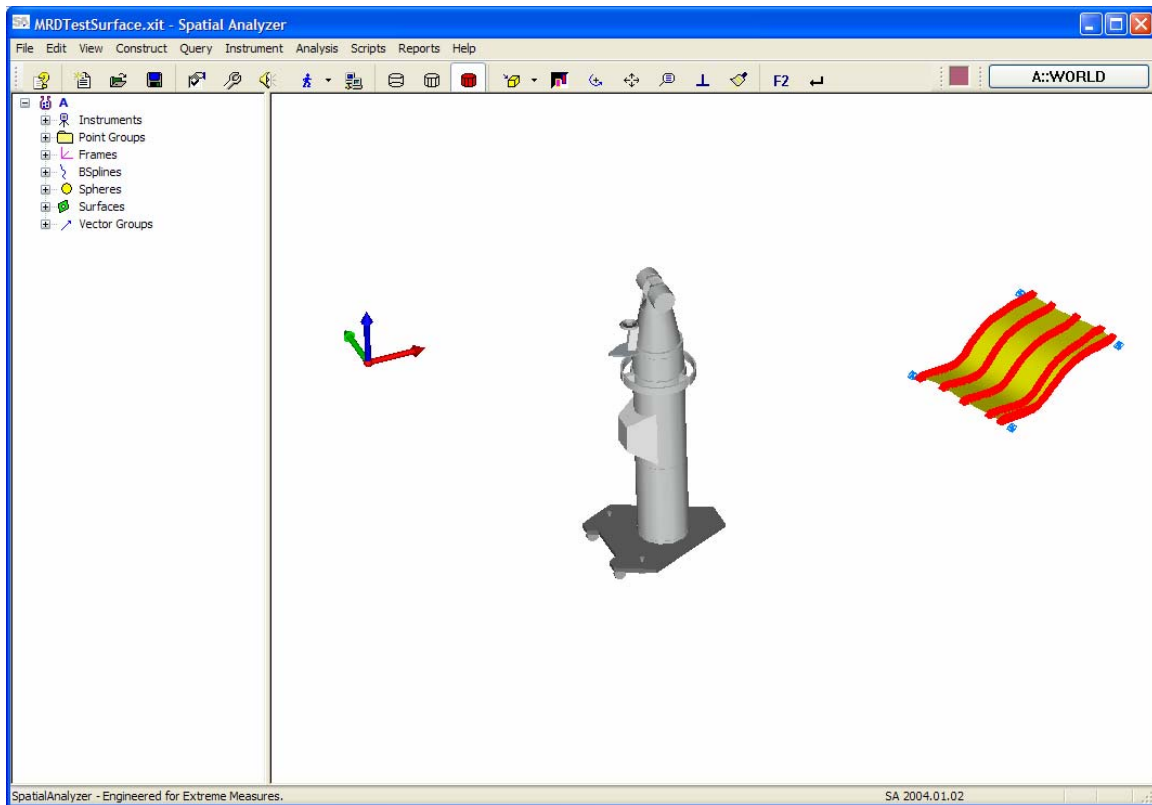
The figure below shows the results after accepting the renamed group. The R2 Auto-Cor group contains copies of the points in the R2 group however the point names now correspond to those in the R1 reference group.



Results from Auto-Correlation of Point Groups

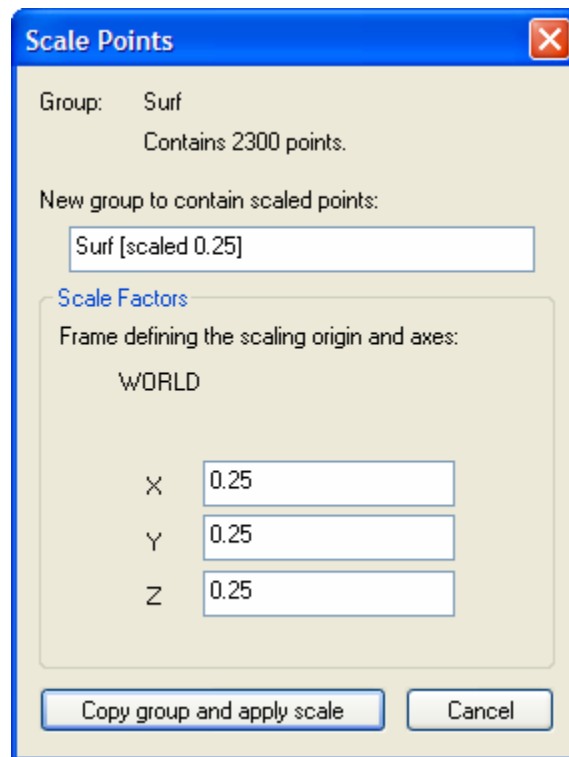
Scale Points

The Scale Points function is able to scale a group of points either consistently in all three axes or differently in each X, Y, and/or Z direction. As an example if the points for the surface show in the figure below need to be scaled down by a factor of 4, you can use the Scale Points function and a scale factor of 0.25.



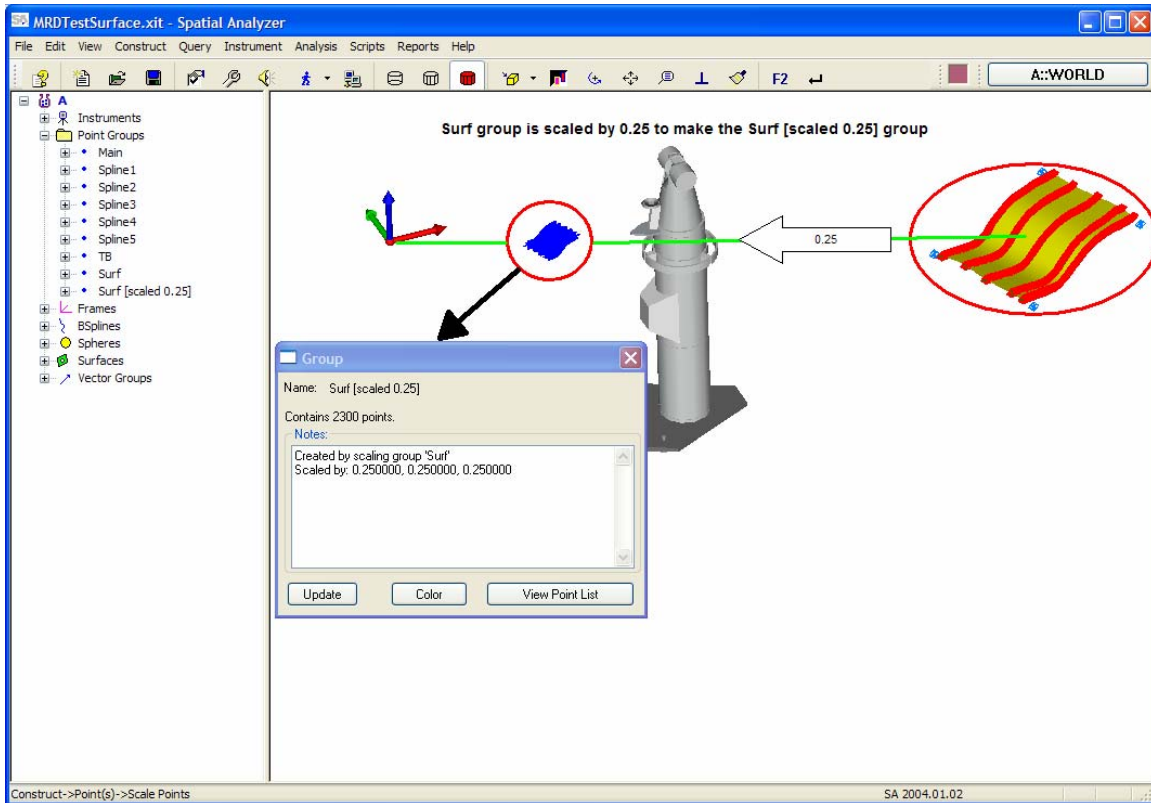
Point group from a model that needs to be scale down by a factor of 4

The first step in Scale Point's process is selecting the frame to scale relative too. The next step is selecting the point group too scale. With the frame and point group known the function now needs to know by how much you need to scale the X, Y, and Z components. A dialog asking for those parameters along with a field for the new group name is presented. For this example a scale factor of 0.25 is used in the X, Y, and Z component directions. Note the component values for each direction can be different. That feature can be particularly useful when compensating for non-isotropic material expansion. The figure below shows the dialog for the example case.



Scale Points Input Dialog

After selecting the 'Copy group and apply scale' button the points are copied and adjusted relative to the selected frame. For the example the results are those shown in the figure below.

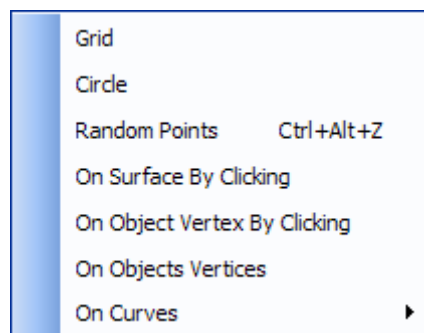


Results from Scale Points Function

Once the group is created the details of the process used to create it is saved in the group's Notes field.

Layout

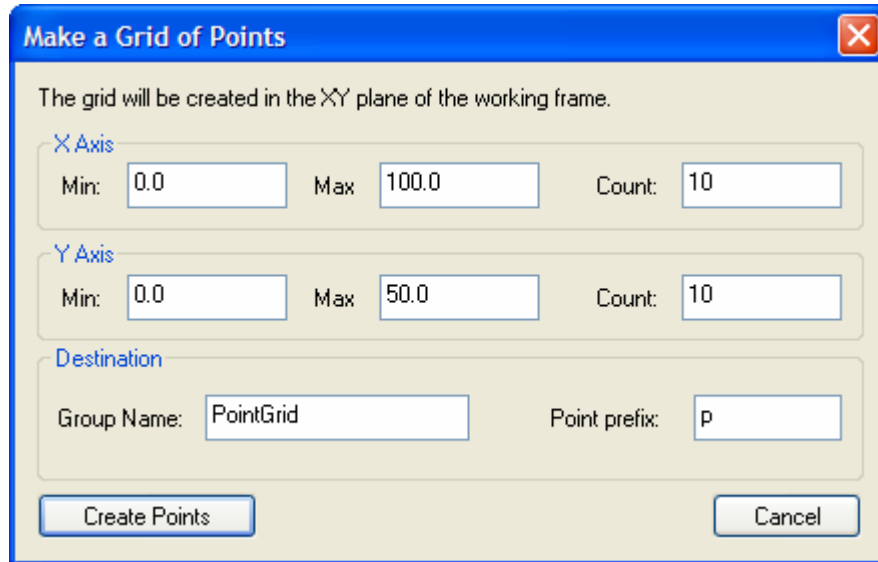
A series of point layout functions are grouped under the Layout submenu. These functions enable you create points in unique ways ... most of these layout processes are the results of special requests from metrology system users.



Layout Point Function Submenu

Grid: This option allows you to create a 2D grid of points in a group. This function is useful for simulating data collected at regularly spaced intervals.

The grid is always created in the XY plane of the working frame. Once created the group of point can be moved with one of the Move Object functions. An alternate technique for positioning the grid in 3D space is to first create a frame with its X-Y plane origin at the location for the initial point of the grid. Then before using the Points >> Layout >> Grid function make the frame the Working frame. The input dialog for the Grid function is shown below.



The dialog box is titled "Make a Grid of Points". It contains the following fields and controls:

- X Axis:**
 - Min: 0.0
 - Max: 100.0
 - Count: 10
- Y Axis:**
 - Min: 0.0
 - Max: 50.0
 - Count: 10
- Destination:**
 - Group Name: PointGrid
 - Point prefix: p

Buttons: "Create Points" and "Cancel".

Make a Grid of Points Dialog

The default grid is created with 100 points, 10 rows in the X and 10 in the Y direction. The axis offsets can be set with the Min and Max fields and then the number of points set in the Count Field. If you want a regular spacing of say 10 inches over a span of 100 inches in the X direction, specify that 11 points (not 10) be created. The distance between points is always distance divided by the number of points minus one.

Circle: Creating a circle or arc of points is a function frequently used when simulating measurement processes. The points on a circle or arc are created in the X-Y plane of the current Working Frame. The default input dialog for creating a circle of points is shown below.

Layout Points on Circle [X]

This function will layout a points in a circle in the XY plane of the working frame.

Radius:

Starting angle (deg., relative to X axis):

Number of points on circle:

☒ Evenly space points around entire circle
☐ Space points by deg.

Point Naming

Group:

First Point:

☐ Create center point

OK Cancel

Default Input Dialog for Laying out Points on a Circle

Inputs for the circle radius, starting angle, and the number of points on the circle are the initial fields to specify. The starting angle is based relative to the X axis. Therefore if you specific starting at an angle of 10 degrees and then space 20 points at an angular interval of 2 degrees the function will create 20 points with the first one at 10 degrees, the second at 12 degrees, then 14 and so on until 38 degrees. The results of this example are shown below.

Layout Points on Circle [X]

This function will layout a points in a circle in the XY plane of the working frame.

Radius:

Starting angle (deg., relative to X axis):

Number of points on circle:

☐ Evenly space points around entire circle
☒ Space points by deg.

Point Naming

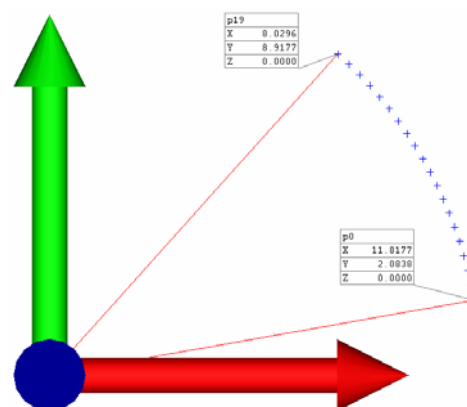
Group:

First Point:

☐ Create center point

OK Cancel

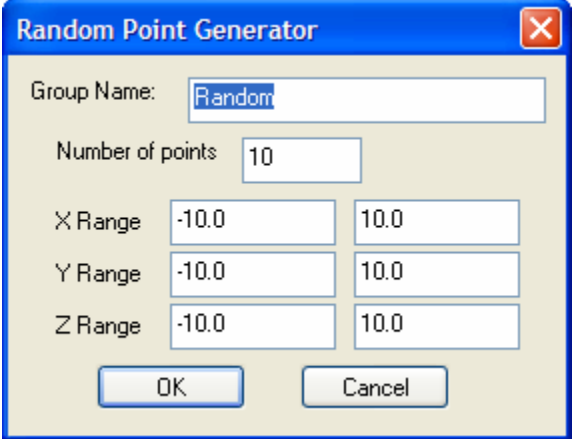
Layout Points on a Circle Dialog



Arc of Points

To specify “point spacing” from a start angle to another angle set the number of points on the circle to the number of spaces plus one. Referring back to the example above; to get the last point at 40 degrees from the start we need to set the number of points on the circle to 21 instead of 20.

Random Points: Creating a set of random 3D points in SA is accomplished with the Random Points function under the Points >> Layout submenu. The 3D space X, Y, and Z ranges are defined relative to the current Working Frame. The default input dialog is shown below.

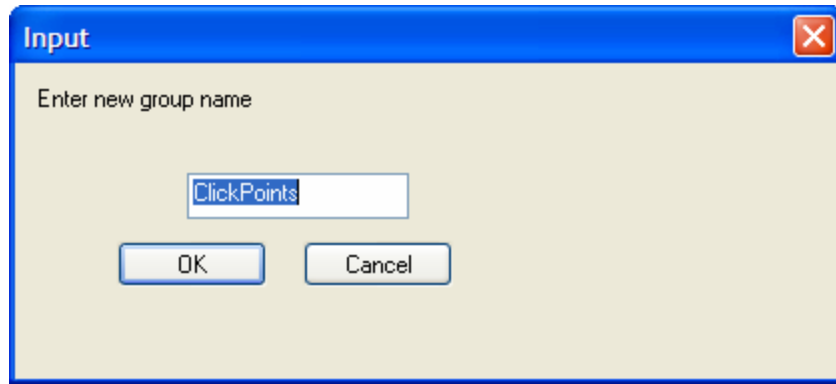
A screenshot of the "Random Point Generator" dialog box. The dialog has a blue title bar with the text "Random Point Generator" and a red close button. The main area is light beige. It contains several input fields: "Group Name:" with a text box containing "Random"; "Number of points" with a text box containing "10"; "X Range" with two text boxes containing "-10.0" and "10.0"; "Y Range" with two text boxes containing "-10.0" and "10.0"; and "Z Range" with two text boxes containing "-10.0" and "10.0". At the bottom are "OK" and "Cancel" buttons.

Create Random 3D Points Dialog

Set the group name and the total number of random 3D points to create. The next set of inputs allows you to specify the spacing over which to randomly select points. The X, Y, and Z component are each randomly chosen so the space does not have to be a cube, or even a 3D volume.

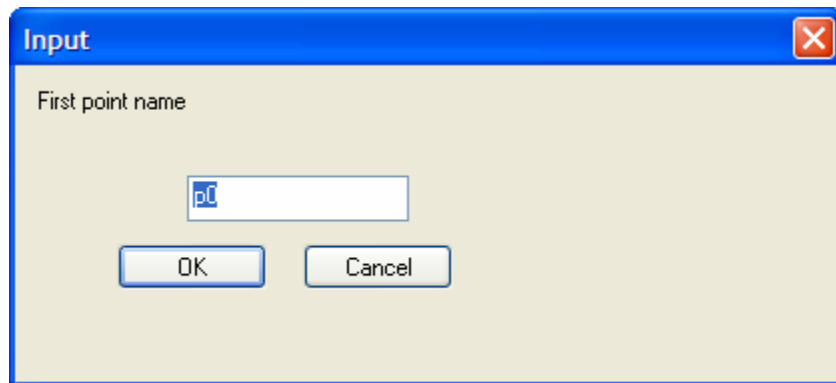
To create random points on a plane set one of the 3 components range values to the same number (e.g., Z Range = -10 and -10.) This results in a plane of randomly spaced points all with Z components of -10. A line of randomly spaced points is created by setting two of the three component ranges to the same number.

On Surface by Clicking: This option allows you click on a surface to create a point. SA projects the clicked location onto the surface and then creates a point. This function is setup by first entering the name of the group to put the points in (shown below.)



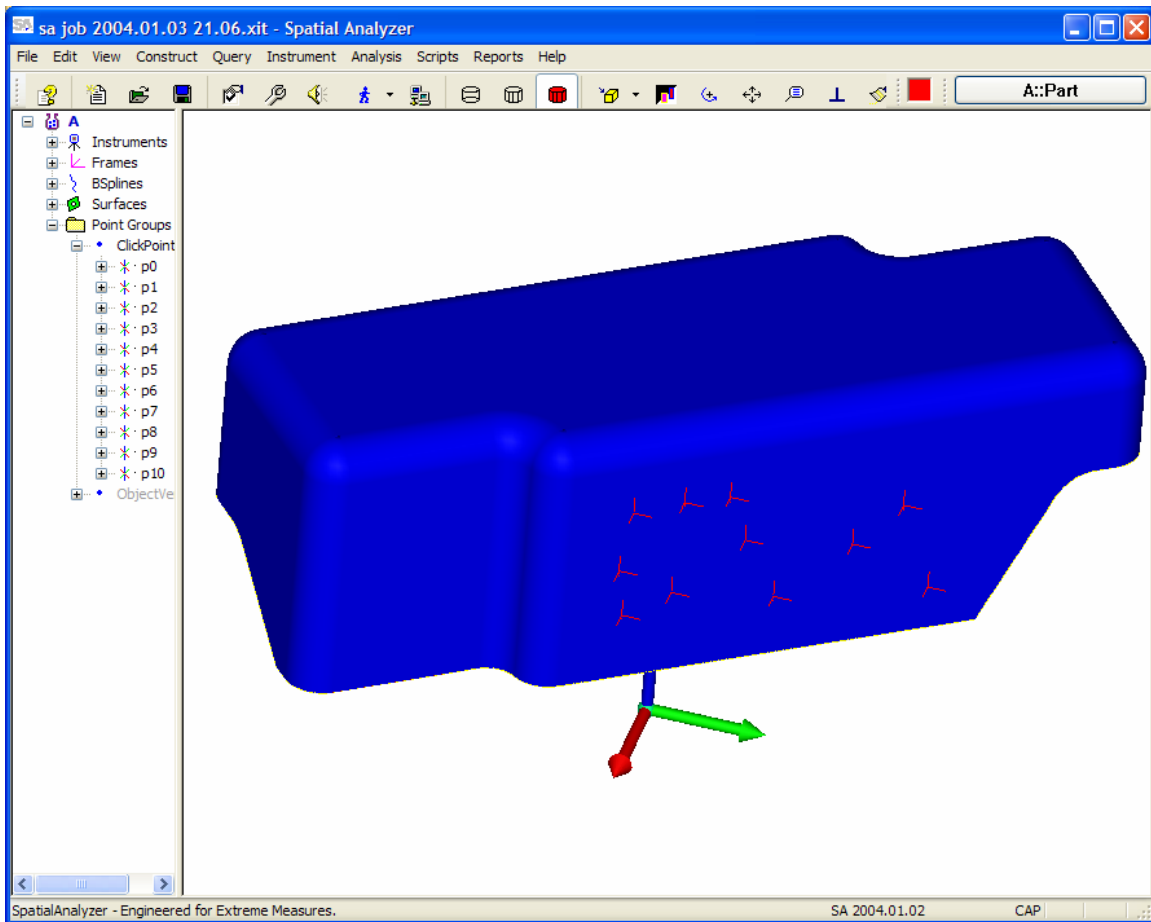
Group Name Dialog

The next step is to identify the name of the first point. Subsequent points are automatically named based on the first point's name. The point name input dialog is shown below.



Point Naming Dialog

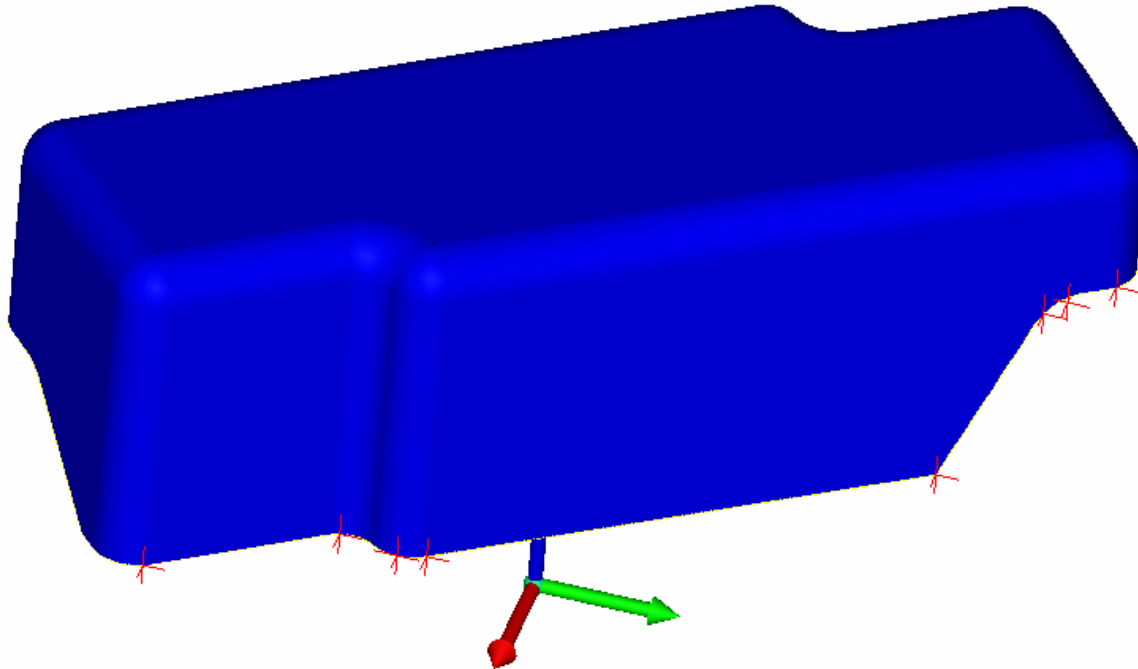
With the group and point name fixed the next step is to click on the surface. Each time you do a point is projected onto the surface and put in the group. The figure below shows an example.



Results from Construct >> Points >> Layout >> On Surface by Clicking

To end the process hit the enter key.

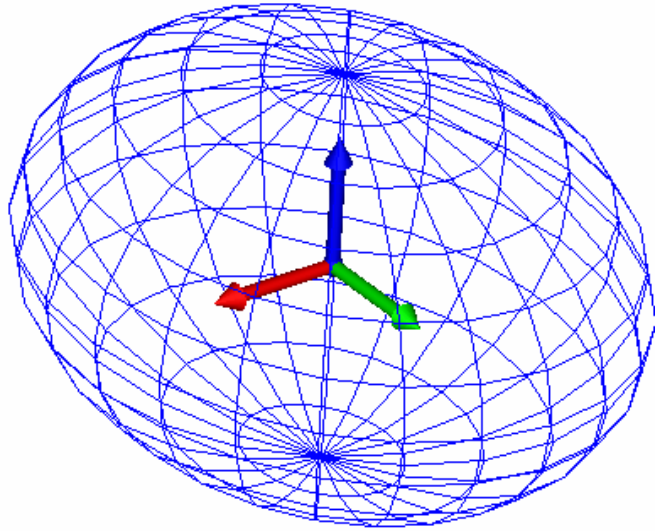
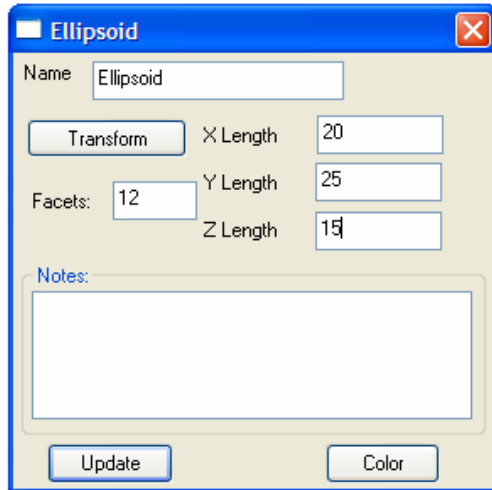
On Object Vertex by Clicking: This option allows you to create a group of points that lie on the vertices of an object. This function is useful for simulating data collection of a complex part. The process follows the same one used for clicking on surfaces. The difference for this function is the projection function finds the closest object vertex and creates a point. An example of the results is shown below.



Results from Construct >> Points >> Layout >> On Object Vertices by Clicking

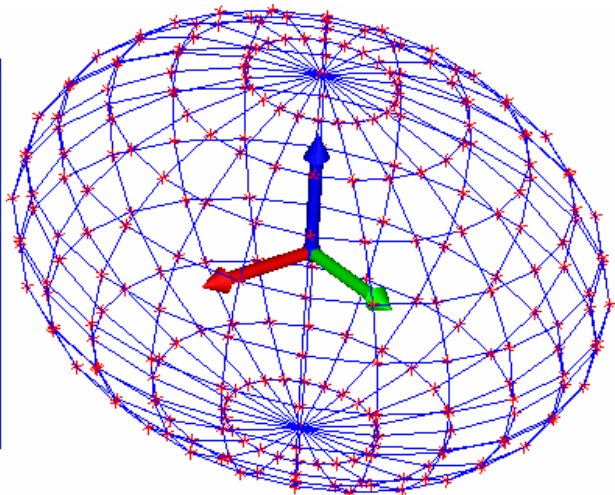
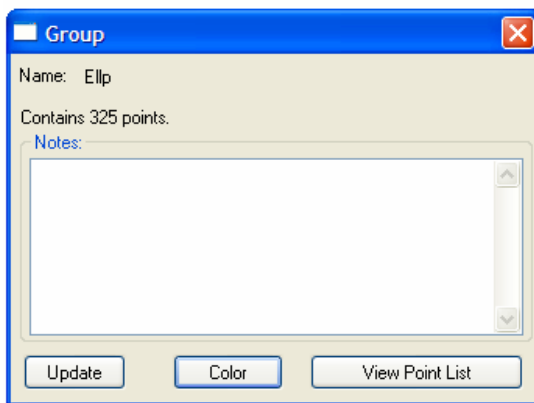
On Object Vertices: SA brings a number of functions to help layout points for a job. The function Construct >> Points >> Layout >> On Object Vertices is able to automatically create a whole series of points in critical areas. These points are then typically used as drive points for measurement processes or in simulations efforts where the measurements of these points are critical to a job's success.

With this function you select an object plus a point group name and points at each vertex are created. Each geometric object in SA supports constructing points on their vertices, making this a general function for creating points to simulate geometry. The example below uses an Ellipsoid object created in SA with the Construct >> Ellipsoid >> Enter function.



Results from Construct >> Points >> Layout >> On Object Vertices by Clicking

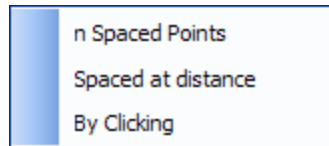
The figure below shows the results of the Construct >> Points >> Layout >> On Object Vertices function. 325 points were created at each object vertex. A change to the facet property will yield a different set of points when you use this function.



Results from Construct >> Points >> Layout >> On Object Vertices by Clicking

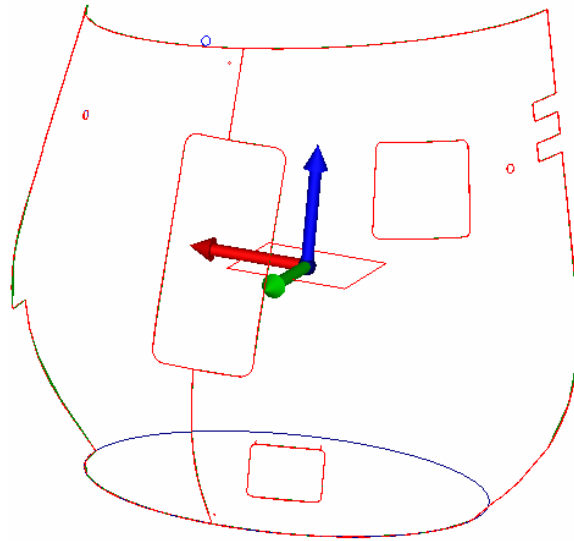
On Curves: This option helps you to create points that lie on curves at locations you choose, at specific intervals along the curve, and will create a certain number of points along a curve. Each function offers a slightly different mechanism to create points on curves and together they bring solutions for your metrology applications needs. This is another function that is useful for simulating data collection of a complex part.

The figure below shows the Layout >> On Curves submenu from Construct >> Points



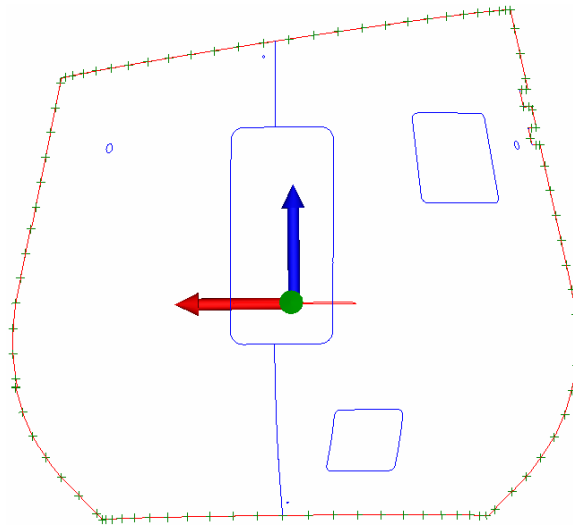
On Curves Point Layout Submenu

For this set of functions the model shown in the figure below helps illustrate the output point group for each function.



On Curves Point Layout Example

On Curves >> n Spaced Points: Putting a specific number of points on a curve at regularly spaced intervals is accomplished with the On Curves >> n Spaced Points function. Inputs for the function include selecting the curve, point group name and the number of points to put on the curve. The figure below shows the results when constructing 100 points on a curve.

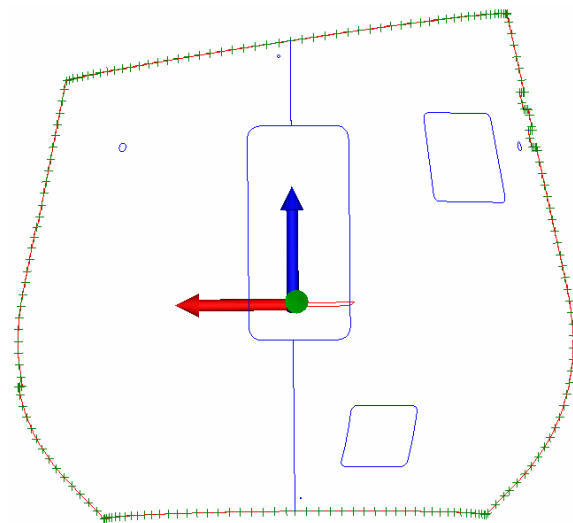


On Curves >> 100 Spaced Points

With this function the distance between points along the curve is equal and evenly divided between the 100 points (e.g., 99 spaces.)

On Curves >> Spaced at a distance: This option allows you to create a group of points where each point is at a set distance along the curve. This function is useful for simulating data collection of a complex part. Inputs include selecting the curve, defining the new group name, and setting the distance along the curve. Job units are used for the distance units along the curve.

The figure below shows the results produced with 1 inch point to point spacing.



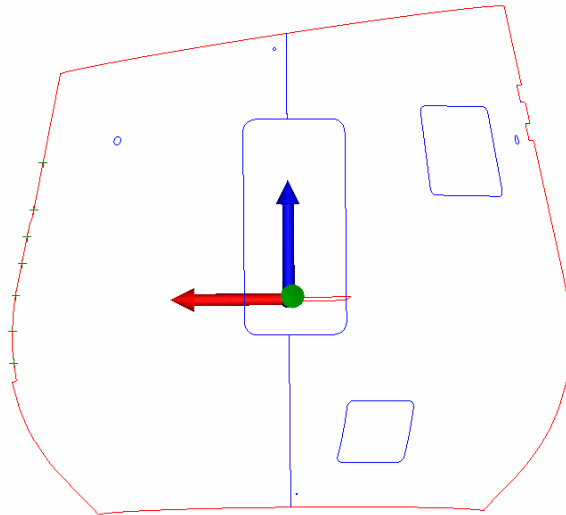
On Curves >> Spaced at 1-inch

This example produced 211 points for the 1 inch spacing. The points are on the curve and the distance between the points is 1 inch when measured along the curve. Using

this point count we can say that the curve is at least 210 inches long but not longer than 211 inches.

On Curves >> By Clicking: Constructing points on a curve by clicking is a function that is helpful. After selecting the curve(s) to put the points on you are asked to name the group and first point. With these inputs know you can click anywhere close to the curves and the closest point to the curves is constructed and put into the group.

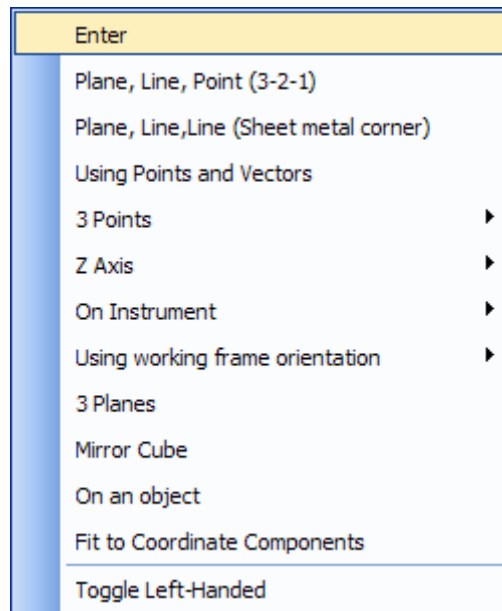
The figure below show the results for several points.



On Curves >> Spaced at 1-inch

Additionally, there are several Point import functions available within the Spatial Analyzer. An IGES file may be imported from another CAD system, or, a point data file, like a GSI V-STARS file, may be imported.

Frames

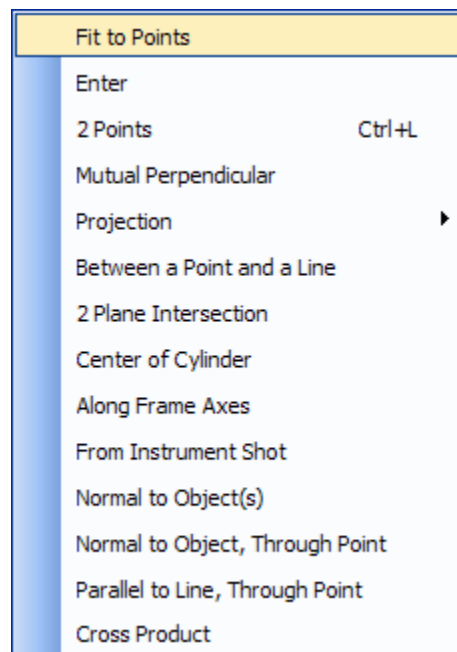


Construct Frames Submenu

The Frame construction options are covered in the titled, "Coordinate Frames and Spatial Transformations" section of the manual.

Line

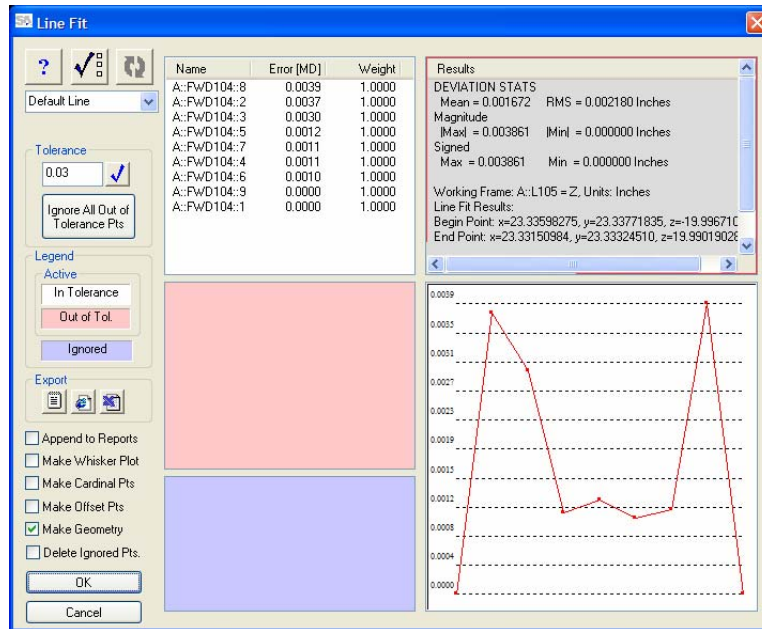
There are several methods for creating lines using **SpatialAnalyzer**. These are listed on the menu shown below.



Line Construction Menu

Fit to Points

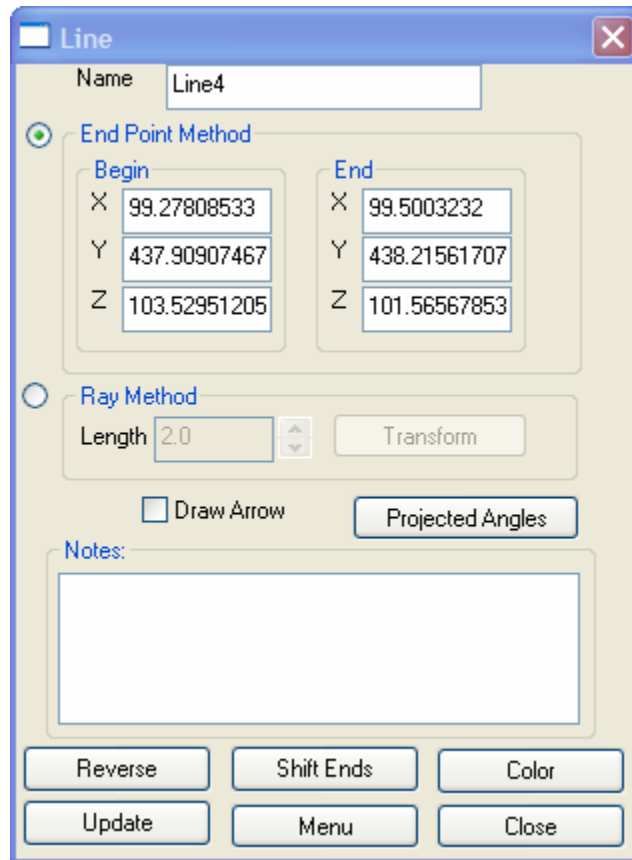
After selecting the points to fit a line to, **SpatialAnalyzer** will determine the line parameters that best-fit the selected points. The results of the fit are displayed, and you will have the chance to modify the results in the Line Fit dialog (shown below).



Line Fit Results Dialog

Enter

This will allow you to enter the end-points of the line or designate a direction and a length. These values are entered using the Line Property dialog shown below.



The image shows a 'Line' property dialog box. At the top, there is a 'Name' field containing 'Line4'. Below this, there are two radio buttons: 'End Point Method' (which is selected) and 'Ray Method'. Under 'End Point Method', there are two columns of input fields: 'Begin' and 'End'. The 'Begin' column has X: 99.27808533, Y: 437.90907467, and Z: 103.52951205. The 'End' column has X: 99.5003232, Y: 438.21561707, and Z: 101.56567853. Under 'Ray Method', there is a 'Length' field with '2.0' and a 'Transform' button. Below these, there is a 'Draw Arrow' checkbox and a 'Projected Angles' button. At the bottom, there is a 'Notes' text area and a row of buttons: 'Reverse', 'Shift Ends', 'Color', 'Update', 'Menu', and 'Close'.

Line Property Dialog

2 Points Ctrl+L

You can construct a line between two points in the model using this option. You will be prompted to select each point and then the line will be created. The property dialog will appear next allowing you to assign a name to the line and change and other properties as needed.

Mutual Perpendicular

This function will prompt you to select two lines. It will then determine the line that is the mutual perpendicular to the selected lines and create it.

Projection >> Line to >> Plane // Circle

After selecting either a plane or a circle as the destination, you will be prompted to select the line you wish to project. A new line will be created on the projection surface.

Between a Point and a Line

This function creates a line between a user-selected point and a line.

2 Plane Intersection

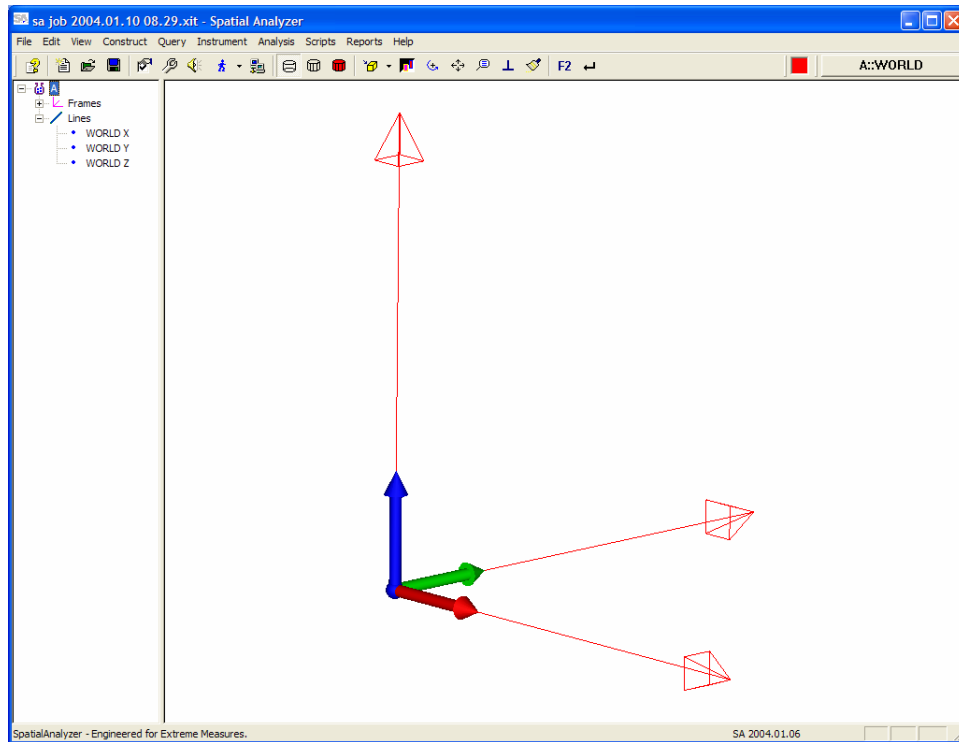
A line is created at the intersection of two planes. If the planes are parallel, the function reports that the line can not be created.

Center of Cylinder

This function is provided to allow you to easily access the axis of a cylinder in the form of a line for other analysis.

Along Frame Axes

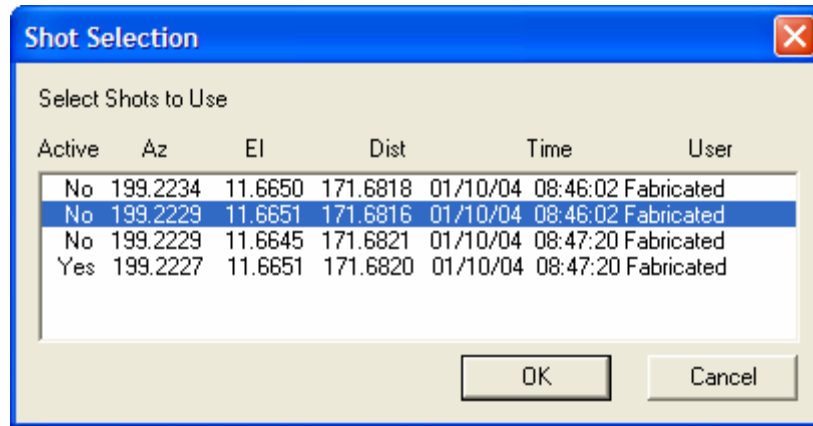
This is another function provided to make it easier to create a line for further analysis. The lines are automatically named with frame name with the axial direction as a suffix (e.g., World X, World Y, and World Z.) The figure below shows an example.



Lines Created along Frame Axes

From Instrument Shot

Creating lines for further analysis from measurements can be a useful tool. This function asks you to first select the instrument and then the target with measurements that you need to make into lines. Since a target could have been measured many times by a station a selection dialog is presented that allows you to select which shot lines to create. An example shot selection dialog is shown below.

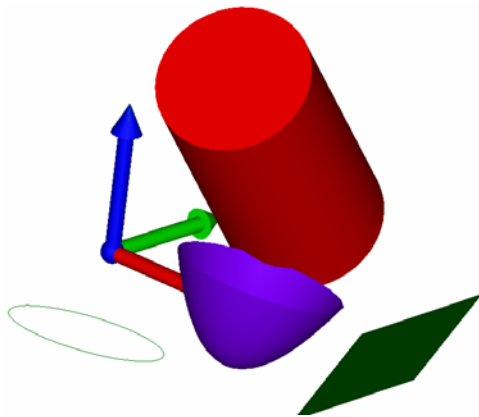


Shot Selection dialog

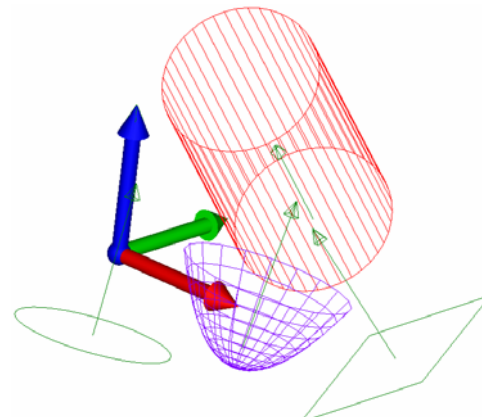
Normal to Object(s)

Creating lines on objects that have inherent normal directions is straightforward with the Normal to Objects function. Objects that have an inherent normal direction include plane, line, circle, cylinder, parabola, cone, and frame (the z-direction). A sphere is an example of an object that does not have an inherent normal direction. It is defined by a point and a radius.

Select the objects that you need to create a line on its normal. Hit the enter key to complete the process. An example of the original objects and the lines created on their normal directions is shown below.



Objects with normal directions



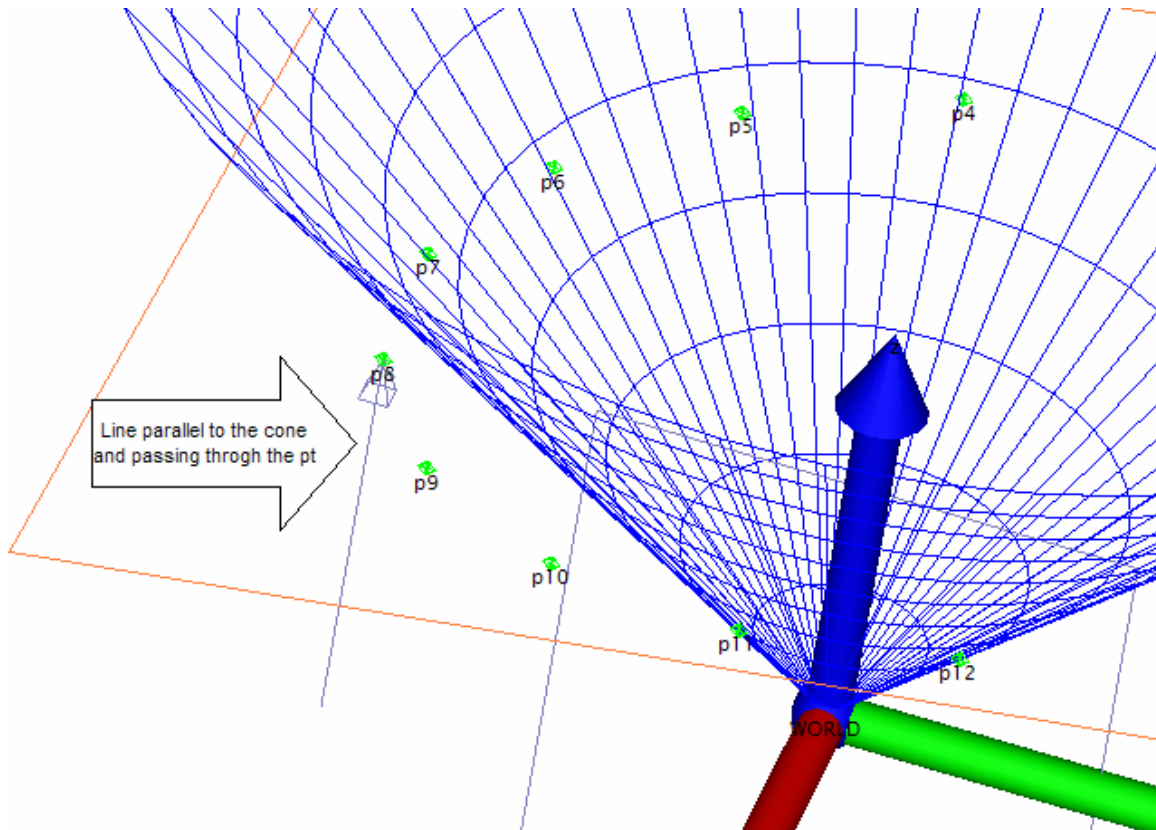
Lines created on object normal directions

Normal to Objects, through Point

Lines that are normal to an object and pass through a point are useful for many analysis tasks. The function Normal to Objects, through a Point creates lines with these properties. The created line will be parallel to the object's normal direction and pass

through a selected point. Objects that have an inherent normal direction include plane, line, circle, cylinder, parabola, cone, and frame (the z-direction).

Selecting a cone and then a point yields a line parallel to the cone's axis that passes through the point. The figure below shows an example.



Create a line parallel to object's normal and through point

Parallel to Line, Through Point

Creating a line that is parallel to an existing line that passes through a point is the output of the Parallel to Line, Through Point function. The inputs for the function are first select a line and then a point. A parallel line passing through the point is created. Graphically the line uses the selected point as one of its ends.

Circles

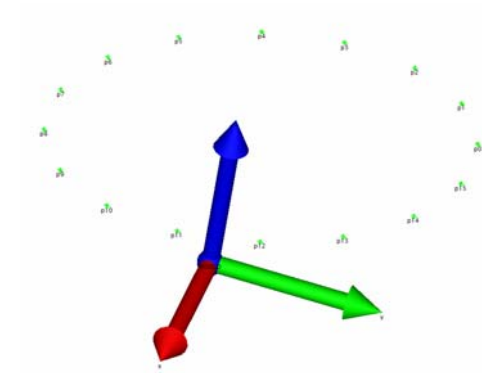
Circles are stored as a center point, a normal vector direction, and a diameter. There is also a parameter for the number of facets used to draw the curved profile. This parameter does not affect the mathematical description of the circle itself, but does affect the visual appearance.

Circles may be constructed with three different methods.

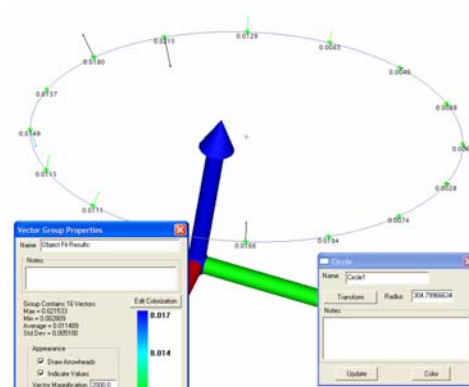
Fit to points

Best-fit circles fitted based on a series of data points. When using this option, you will be prompted to select the points that are to be used in the computation. You may individually double click on all of the points, or you may press F2 to select the points from a list. After all of the points have been selected, press , and the fit algorithm will compute the best-fit circle.

The figures below show an example of a set of points and the circle computed from them.

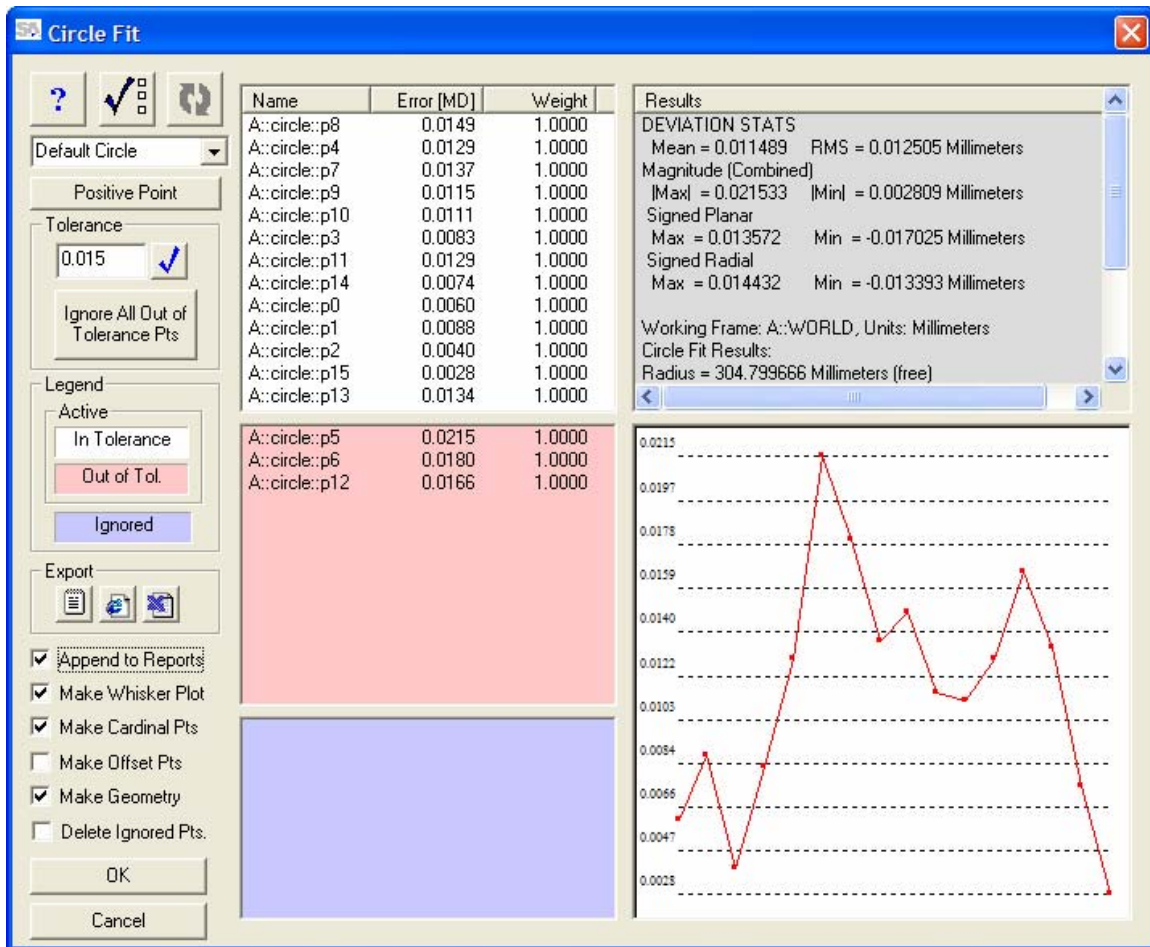


Points to create a best fit circle



Best-fit circle with vector group and center point

After computing the circle a Circle fit dialog showing the residual errors, summary, and graphical views of the residuals is presented. The figure below shows an example for the fit dialog.

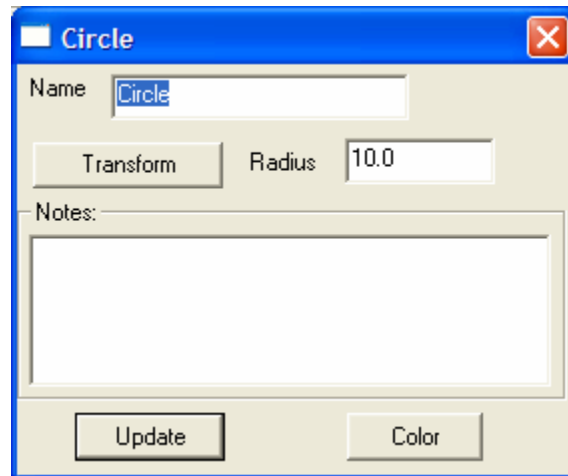


Best-fit circle with vector group and center point

Note also that many of the Plane geometry functions may also be performed on circles. This is because a circle does in fact contain a plane. You may, for example, project a point or line onto a circle in the same manner that you project them onto a plane.

Enter

This is the manual entry method for circle construction. Here the user simply selects the Construct, Circle, Enter menu option and is presented with the circle properties dialog. Using this dialog the operator may enter the circle radius, change the circle's transform, and edit the Color properties of the circle.



Circle Properties Dialog

Maximum Material Condition (from scan patch)

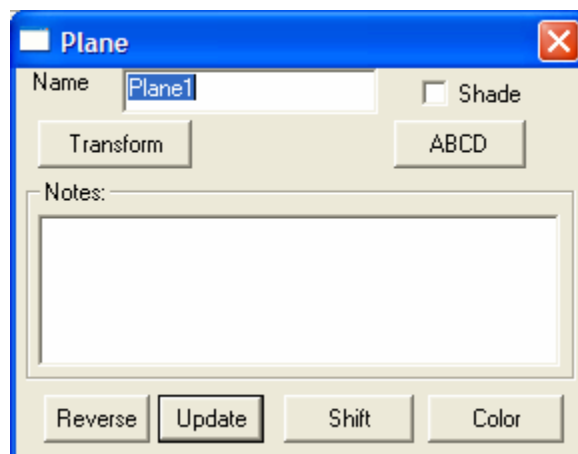
Planes

Planes are stored using the plane coefficients A, B, C, and D. These coefficients store the mathematical representation of the plane given by:

$$Ax + By + Cz + D = 0$$

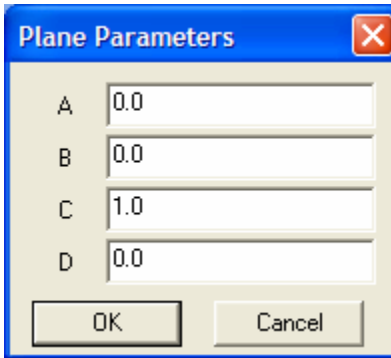
In addition to the mathematical description, there are also physical points created to define the extents of the plane. It is important to note that these extents are merely used to draw a representation of the infinite plane. All projection operations and intersection operations will use the mathematical representation of the infinite plane.

The plane properties dialog (shown below) has controls for naming the plane and setting the shading attribute. Transforming or moving the planes position and orientation is accessible with the transform button.



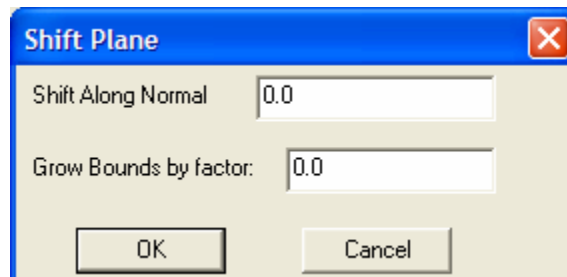
Plane Properties Dialog

Pressing the button marked ABCD produces the plane's parameter dialog. A plane is defined as a direction at a distance from the working frame. The A, B, C parameters are the normalized vector components forming the direction. D is the distance from the origin along the vector. For the example shown below the plane is in the x-y plane and passes through the origin.

A screenshot of a software dialog box titled "Plane Parameters". It has a blue header bar with a red close button (X) on the right. The dialog contains four input fields labeled A, B, C, and D. Field A contains "0.0", field B contains "0.0", field C contains "1.0", and field D contains "0.0". At the bottom are two buttons: "OK" and "Cancel".

Plane Properties Dialog

Shifting a plane along the normal is a common mechanism in a number of metrology applications. The Shift button on the properties dialog makes this task straight-forward. Enter the distance to shift the plane into the Shift Along Normal field. The sine of the value determines the sense of the along the normal.

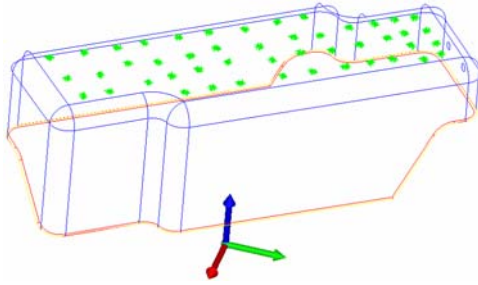
A screenshot of a software dialog box titled "Shift Plane". It has a blue header bar with a red close button (X) on the right. The dialog contains two input fields. The first is labeled "Shift Along Normal" and contains "0.0". The second is labeled "Grow Bounds by factor:" and contains "0.0". At the bottom are two buttons: "OK" and "Cancel".

Plane Properties Dialog

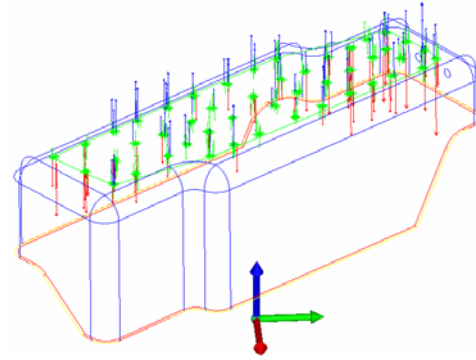
The Grow Bounds by factor allows you to grow or shrink the graphical presentation of the plane. Enter a value that is less than one to shrink the graphical re-presentation. Entering a value greater than one for the Grow Bounds by factor field grows the graphics presentation of the plane.

Fit to Points

Planes are most often created by determining the coefficients A, B, C, and D that best-fit a series of points. To do this, select the plane-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press , and the best-fit parameters will be determined and the errors will be displayed in a Fit Dialog. The two figures below show the points on a planar surface and then after the plane is solve; a vector group showing the residual errors against the computed best-fit plane.

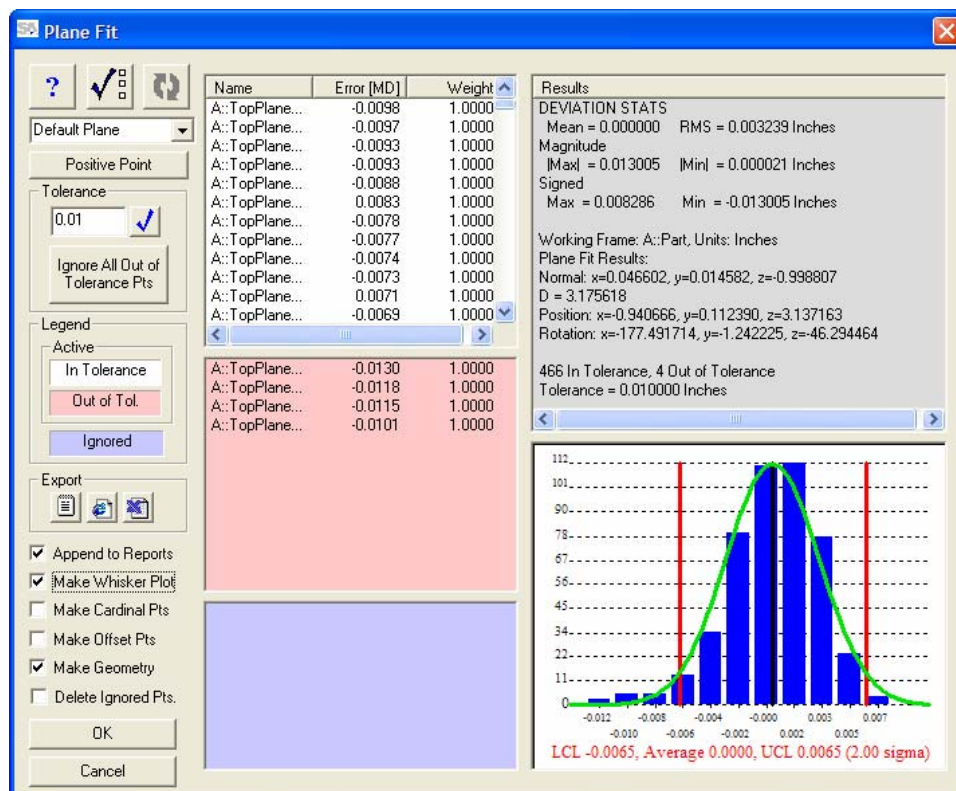


Points to fit Plane



Plane + Vector Group

The plane fit dialog provides a number of analysis views of the fit and the details of computed plane. You are able sort and modify which points are included in the fit ... resolve and then define what output analysis and reports you want created from this dialog. The figure below is an example showing the fit for the plane points. The graphic is a histogram of the residual errors. In this case the errors have a nice Gaussian profile.



Plane Fit Dialog

Enter

This is the manual entry method for plane construction. Here the user simply selects the Construct, Plane, Enter menu option and is presented with the plane properties dialog. Using this dialog the operator may enter the plane coefficients, change the plane's transform, edit the Color properties of the plane, and if desired represent the plane as a shaded object.

Normal to Object, through Point

A plane is created that is normal to a selected object and passes through a selected line. Objects that have an inherent normal direction include plane, line, circle, cylinder, parabola, cone, and frame (the z-direction).

Normal to Line, through end

A plane is created that is normal to the line and passes through a selected end of line.

Mirror from 2 measured points

The mirror plane function is used to determine the location of a mirror by measuring a point both directly and through the mirror. SA also includes uncertainty analysis for the location and orientation of the mirror.

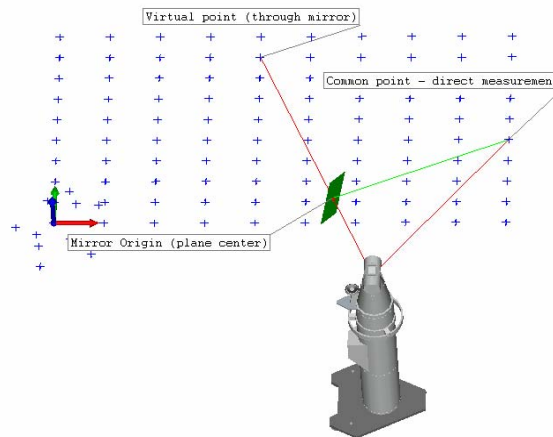
The process follows this outline:

Pick the instrument.

Pick the point measured directly (not through mirror)

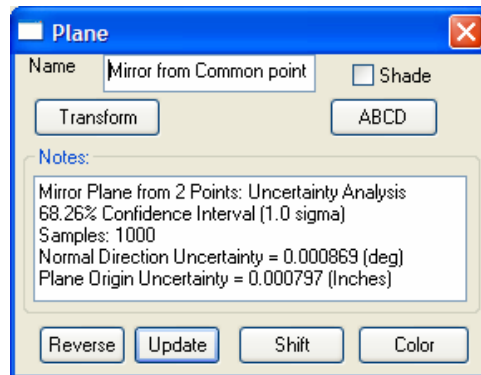
Pick the virtual point that is really the same point, but measured through the mirror.

SA creates a plane representing the mirror location. You can make a frame on this plane if needed.



Layout for Mirror from 2 Measured Points

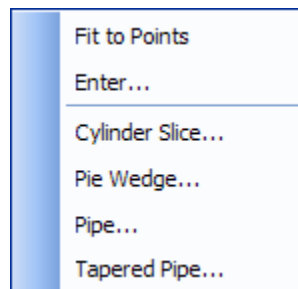
If the points used in the construction have uncertainty fields (Analysis>>Coordinate Uncertainty>>Create Point Uncertainty Fields), SA will automatically compute the uncertainty of the mirror plane and add the results to the notes field for the constructed plane:



Plane Properties Results Mirror from 2 Measured Points function

Cylinders

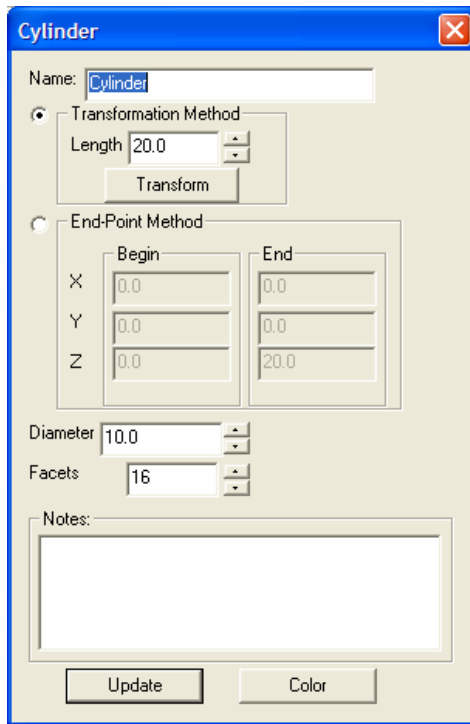
A cylinder is defined using two points and a diameter. There is also a facets parameter, but as with circles, this parameter only affects the graphical representation of the geometry.



Cylinder Construction Options

Enter

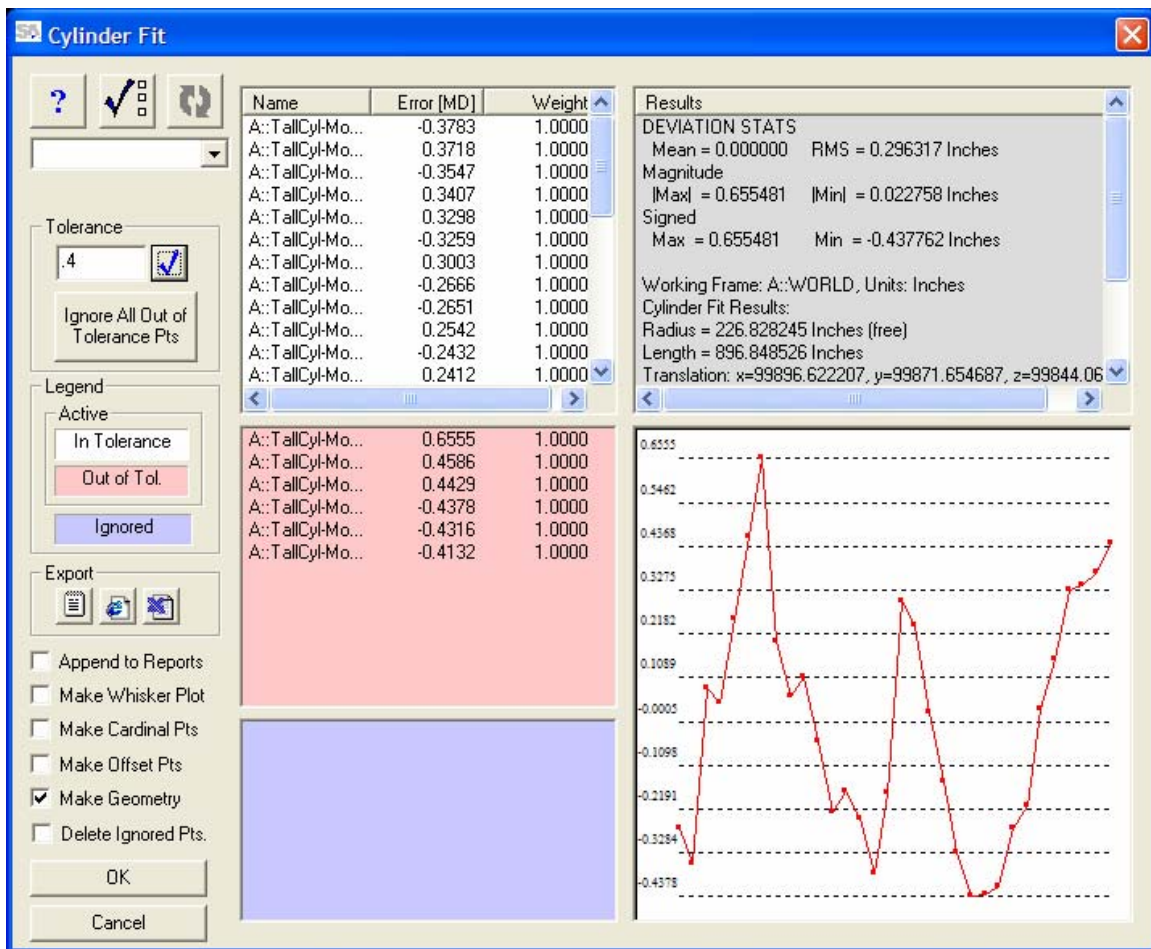
In the Spatial Analyzer, the cylinder properties window allows the user to enter and view the parameters in two ways. One method displays the A and B coordinates as well as the diameter. The other method uses a transformation matrix to place the base of the cylinder, a length along the Z axis of that transformation, and a diameter. The interface will allow you to toggle between these two description options.



Cylinder Properties Dialog

Fit to Points

One may also create Cylinders by using a best-fit algorithm. This algorithm will take a series of points and determine the cylinder that best-fits the given point field. The interface for this option is similar to that for circles and planes. Once the points are selected, you will be prompted for an initial guess at the cylinder's diameter. This helps the algorithm to find the appropriate solution. Next, the fit is performed, the errors are displayed, and the cylinder will appear.



Cylinder Fit Dialog

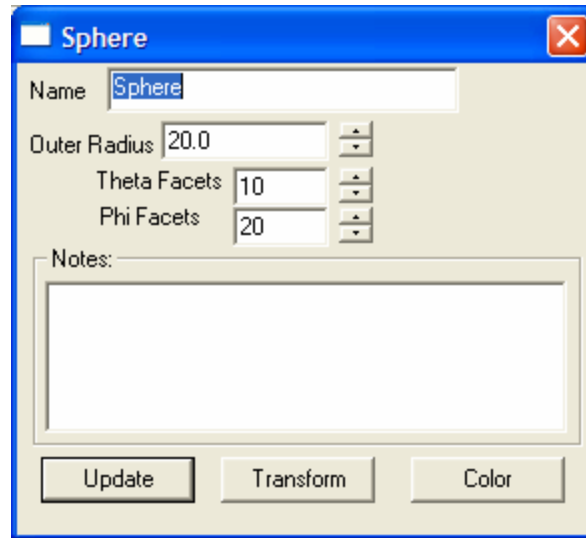
There is also a "re-fit to data" option on the Construct menu (Construct >> Re-Fit Geometry >> Use existing geometry as a starting point). This option allows you to perform the fit operation again using the current cylinder as the starting guess. This is to help the algorithm find the appropriate solution when dealing with point data that represents "squat" cylinder. When you press the re-fit button, you will be prompted to select the points for the fit, and the operation will be performed.

Sphere

Spheres are stored as a center point and a radius. There is also a facets parameter, but as with circles, this parameter only affects the graphical representation of the geometry.

Enter

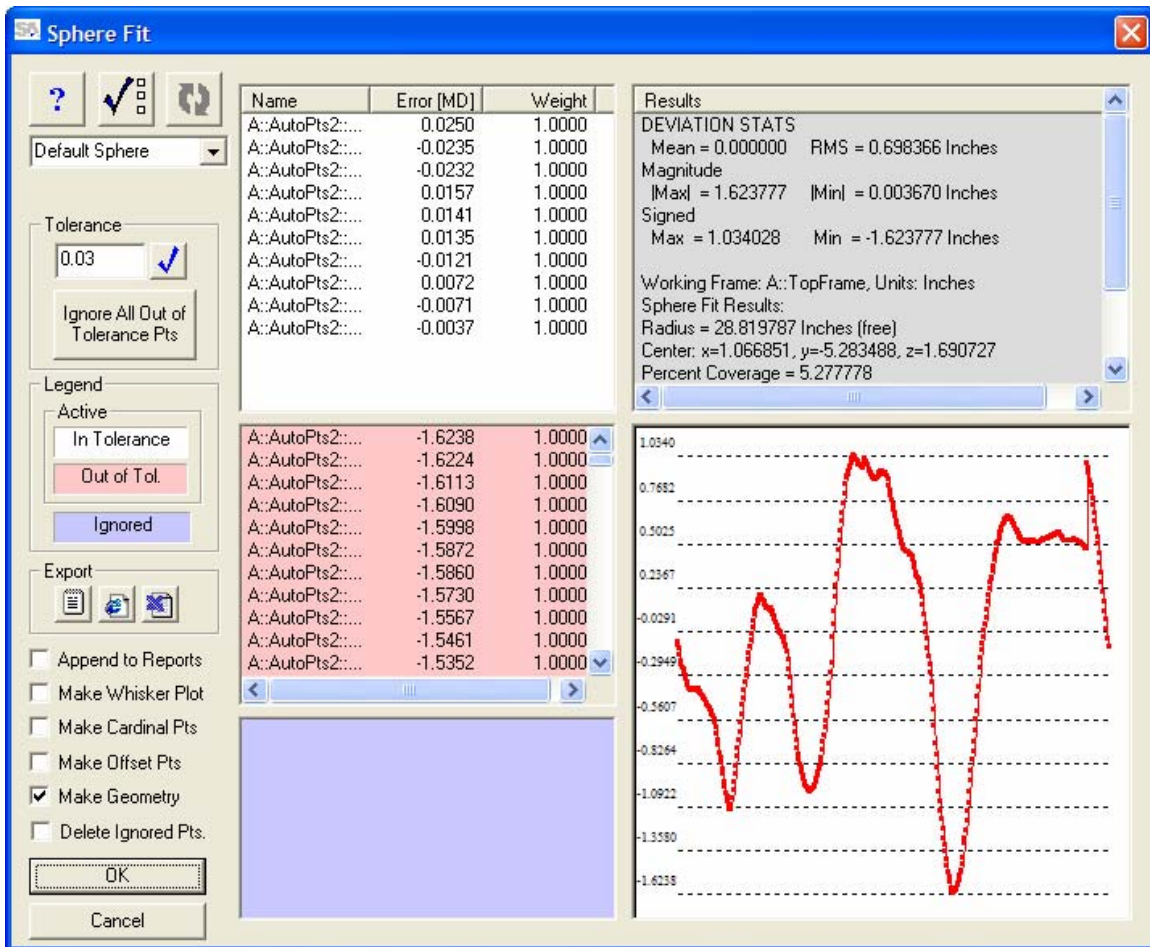
This is the manual entry method for Sphere construction. Here the user simply selects the Construct, Sphere, Enter menu option and is presented with the sphere properties dialog. Using this dialog the operator may enter the sphere center coordinates, change the sphere's radius, and edit the Color properties of the sphere.



Sphere Properties Dialog

Fit to Points

Spheres are most often created by determining the center point and radius that best-fit a series of points. To do this, select the sphere-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press enter and the best-fit parameters will be determined and the errors will be displayed in the fit dialog. An example Sphere fit dialog is shown in the figure below.



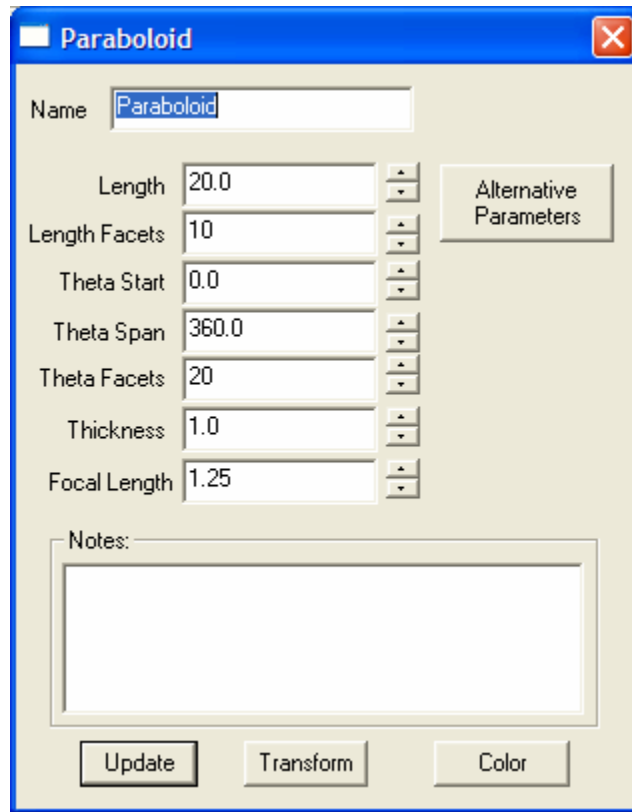
Sphere Fit Dialog

Paraboloid

Paraboloids are stored as a transform, a physical length, and an focal length. There are also several other presentation parameters, but these parameters only affect the graphical representation of the geometry.

Enter

This is the manual entry method for cone construction. Here the user simply selects the Construct, Paraboloid, Enter menu option and is presented with the paraboloid properties dialog. Using this dialog the operator may enter the paraboloid transform, change the paraboloid's physical length and focal length, and edit the Color properties of the paraboloid.



The image shows a software dialog box titled "Paraboloid". It contains several input fields for defining a paraboloid's properties. The "Name" field is set to "Paraboloid". The "Length" field is 20.0, "Length Facets" is 10, "Theta Start" is 0.0, "Theta Span" is 360.0, "Theta Facets" is 20, "Thickness" is 1.0, and "Focal Length" is 1.25. Each of these fields has a small up/down arrow icon to its right. To the right of these fields is a button labeled "Alternative Parameters". Below the input fields is a "Notes:" label followed by a large empty text area. At the bottom of the dialog are three buttons: "Update", "Transform", and "Color".

Paraboloid Property Dialog

Fit to Points

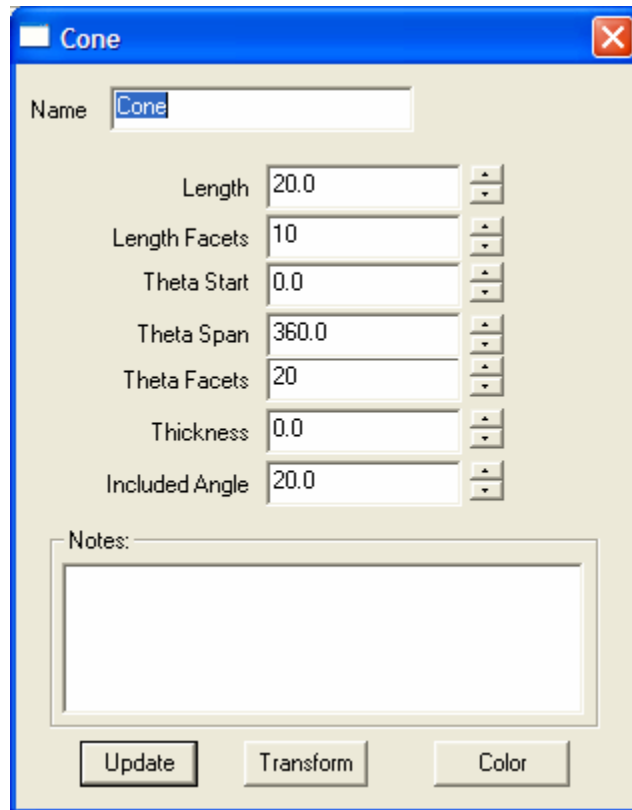
Paraboloids are most often created by determining the transform and focal length that best-fit a series of points. To do this, select the paraboloid-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press enter and the best-fit parameters will be determined and the errors will be displayed in the fit dialog.

Cone

Cones are stored as a transform, a length, and an included angle. There are also several other presentation parameters, but these parameters only affect the graphical representation of the geometry.

Enter

This is the manual entry method for cone construction. Here the user simply selects the Construct, Cone, Enter menu option and is presented with the cone properties dialog. Using this dialog the operator may enter the cone transform, change the cone's length and included angle, and edit the Color properties of the cone.



The image shows a software dialog box titled "Cone". It has a blue title bar with a close button (X) in the top right corner. The main area is light beige. At the top, there is a "Name" label followed by a text box containing the word "Cone". Below this, there are seven rows of numerical input fields, each with a label to its left and a small up/down arrow button to its right. The fields contain the following values: Length (20.0), Length Facets (10), Theta Start (0.0), Theta Span (360.0), Theta Facets (20), Thickness (0.0), and Included Angle (20.0). At the bottom of the dialog, there is a "Notes:" label followed by a large, empty text area. Below the notes area are three buttons: "Update", "Transform", and "Color".

Cone Property Dialog

Fit to Points

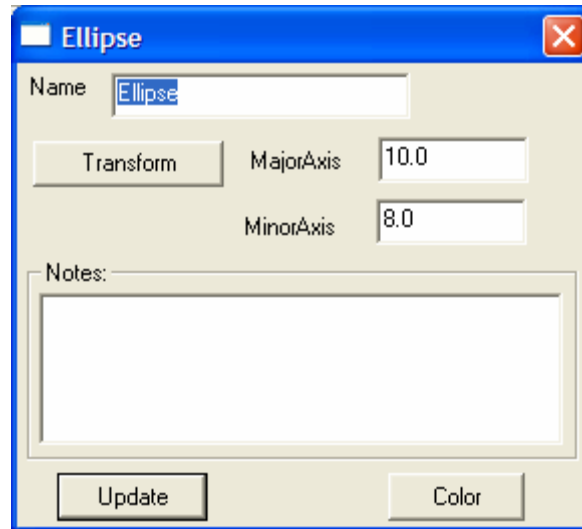
Cones are most often created by determining the transform, length, and included angle that best-fit a series of points. To do this, select the cone-fit option then pick the points. As with all other point selection operations, you may either click on the points, or press F2 and select them from the list. Once you are finished, press enter and the best-fit parameters will be determined and the errors will be displayed.

Ellipse

Ellipses are stored as a transform, a major axis, and a minor axis. There is also a facets parameter, but as with circles, this parameter only affects the graphical representation of the geometry.

Enter

This is the manual entry method for ellipse construction. Here the user simply selects the Construct, Ellipse, Enter menu option and is presented with the ellipse properties dialog in the figure below. Using this dialog the operator may enter the ellipse transform, change the ellipse's major and minor axis values, and edit the Color properties of the ellipse.



Ellipse Properties Dialog

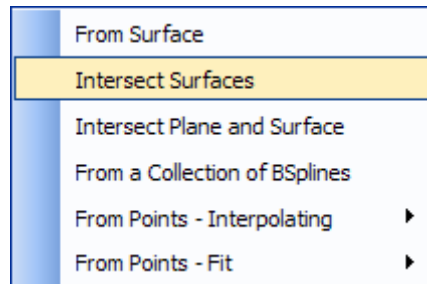
Re-Fit Geometry, Use Existing Geometry as Starting Point

When using numerical methods to solve for geometries from data sets a number of interesting out-comes can be produced. SA provides an alternate method for solving for geometric objects from points. The 'Re-Fit Geometry, Use Existing Geometry as Starting Point' function can use an object as an initial guess and then associate a dataset of points to use in a numeric optimization of the object.

The function asks you first to select the object that will serve as the initial guess and then the points to use in the solution. The results are then displayed in a fit dialog.

Curves

You can use these options to construct curves using either a linear or a cubic model. You will be asked to select the points for the curve, and the curve will be constructed.



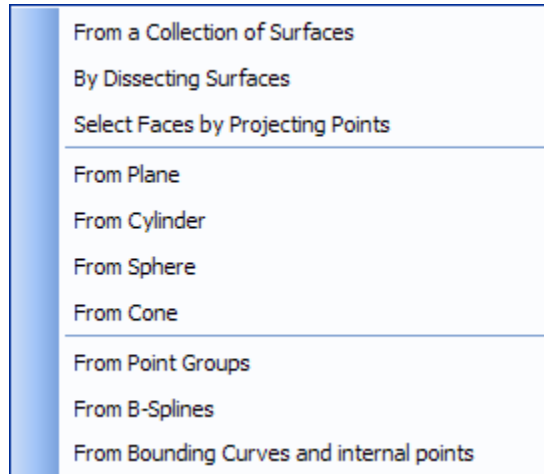
Curve Submenu

From Surfaces

Curves are created on the bounding edges of the selected surface. Each curve is segmented into the constituent of the surfaces.

Surfaces

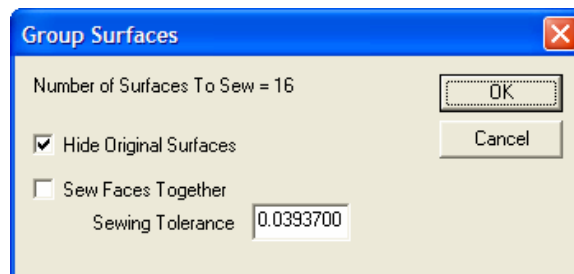
SpatialAnalyzer contains surfacing functionality. You can import nominal surface information from IGES files, compare it to measurements, and produce colorized error plots. In addition, you can best-fit a series of points to a nominal surface and perform a transformation.



Surface Submenu

From a Collection of Surfaces

Surfaces can be merged into a single object for use in analysis functions with this function. Select the surfaces to collect into the common object and press enter. The next step asks for input on whether the original surfaces should be hidden and if the surface faces should be sewn together. An example of the dialog is shown below.



Group Surfaces dialog

After accepting the inputs in the dialog the surfaces are merged together and put on the treeview as a single object. The original surfaces will be hidden if the default option is selected.

By Dissecting Surfaces

If several surfaces have been merged together they can be divided into individual surfaces. Select the surfaces to dissect and hit the enter key. The component surfaces are then created from the parent and put on the treeview.

From Plane

Creating a planar surface is supported with this function. Select a plane and a surface object is created from it and put on the treeview.

From Point Groups and From B-Splines

You can construct a Non-Uniform Rational B-Spline Surface using groups of measured points. In this case, the coordinates must be grouped in scan-line order. In addition, each successive group of points must proceed in the same direction (i.e. left to right for example).

The process of fitting a NURB surface to measured points is quite complex. You must select the order of the surface, and the number of control points you wish to use in both the U and V parametric axes.

The algorithm will then construct a bi-parametric cubic spline surface using the selected points. Using this surface, the initial values for the control points will be guessed by evenly sampling the cubic surface in the U and V direction.

This option allows you to build the surface using only point data. This data must be ordered in terms of scan lines. Within this option, there are several methods for making the point selection:

Enter Group Names – This option requires you to type in the names of the groups that you wish to fit the surface to.

Select a Point from Each Group – You will be allowed to select a single point from each group, and the program will use the entire group for the scan line. Remember to pick the groups in the proper order.

Pick Window Corners – In the case of a laser scanning device, the scan lines can automatically be imported in properly ordered groups. An example is the RAS file format for the CLR measurement device. In this case, the groups are also sequential in the database. As a result, you can pick two points (one in each corner), and the program can determine both the groups to use and the subset of points within the group to use for the scan lines.

From B-Splines

Bi-parametric cubic spline surfaces are composed of a series of parametric cubic curves running in the u direction. By sampling these curves with a uniform parametric sampling, it is possible to construct another set of curves traversing the v direction. These curve definitions produce a surface definition.

If you are to construct a bi-parametric cubic spline surface, you must sample all of the contour lines in the same direction with no overlap. In addition, place each individual line scan in a separate group.

In this case, you select a series of parametric curves in the proper order, and the program will sample them and construct the bi-parametric surface. This requires

creating the curves before creating the surface. To do this, use the Curve construction option discussed previously.

Boxes

This collection of parametric primitives is intended only for use as a graphical modeling toolkit. You may use these geometries to illustrate the measurement process to help a user visualize the task at hand.

This topic is covered in 6 in the Boxes section.

Vector Group

By Clicking Surfaces

From Multiple Vector Groups

User Defined

This option allows you to define a primitive based on an ASCII file containing solid modeling information.

Extrusion...

The extrusion object allows the user to define general solids of extrusion. This is done using a shape file (.SHA). This file contains a list of 2-D points defining the end facet of the extrusion. The extrusion object will load these points, and then extrude the solid as specified. The file format is as follows:

```
5 Points in list
1.6      -6.48      x,y of the point
1.6      1.50
-1.0     1.50
-4.4     -2.00
-4.4     -6.48
```

Rotation...

The rotation object allows the user to define general solids of rotation. This is done using a shape file (.SHA). This file contains a list of 2-D points defining the end facet of the rotation. The rotation object will load these points, and then rotate the solid as specified. The file format is as follows:

```
5 Points in list
1.6      -6.48      x,y of the point
1.6      1.50
-1.0     1.50
-4.4     -2.00
-4.4     -6.48
```

Once a shape has been loaded, the axis of rotation as well as the span of the rotation may be specified.

Custom...

The Custom object may be used to add any general user defined object to the model. This object is loaded from a text file (.USR) using the following format:

```

3      Number of points
7      3 5 6
10     10 -4 8
15     4 10 2
3      Number of Lines
7 10
10 15
15 7
1      Number of Facets
3      7 10 15

```

The points segment includes a list of user-defined points. Each point is preceded by an ID number. This number is used in the line and facet section to identify the point. These numbers do not have to be in order and may skip numbers.

The lines segment contains a list of the endpoints of the lines. 7 10 will connect point 7 to point 10.

The Facets segment contains a list of facet points. These points are preceded by a count indicating the number of points in the facet. The points should proceed around the facet in a right-hand sense. In other words, the right hand rule should yield an outward normal for the facet. The first three points will be used to compute the normal vector for the facet.

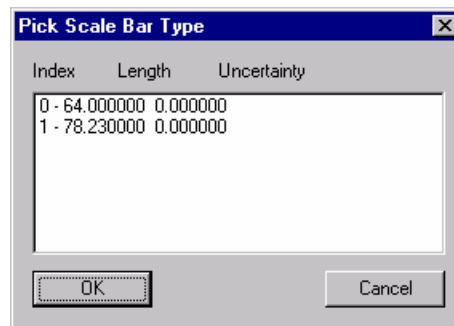
Pyramid...

This is another graphical modeling-only type primitive.

Scale-Bar

This will allow you to add a scale-bar to the database by double clicking on the appropriate end-point in the model. This causes the same effect as going to the Scale-Bar Database tab in the User Options area and entering in the group and point names.

After selecting the points, you will be asked to select the scale-bar type using the dialog shown below. To use this option effectively, you must pre-define the scale-bar types using the property page dialog in user options.

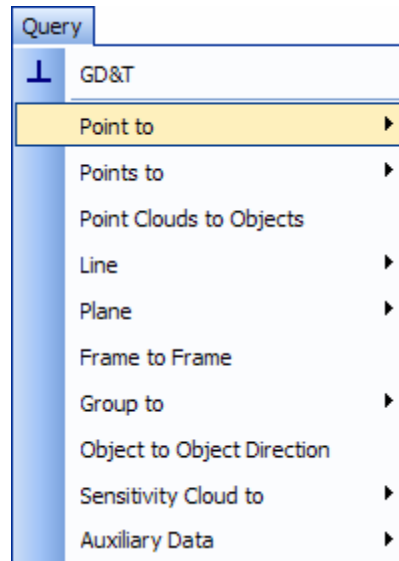


Scale-Bar Selection Dialog

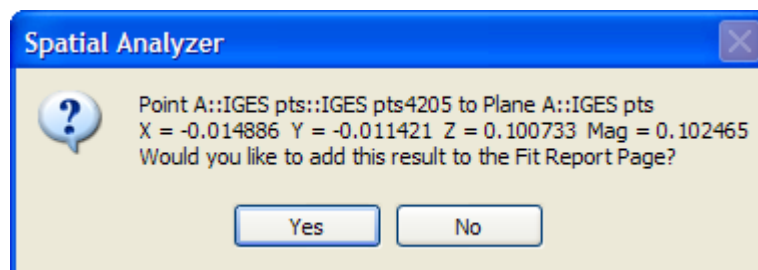
Pick 2 Points

Query

There are several classes of queries shown below. These queries display their results in message-box type dialogs as well as posting the results to the status bar. Example results of a point to plane query are shown in the figure below. If you choose from this dialog, the results of the query will be added to the Fit Reports Page. Also, as will all other coordinate output functions in **SpatialAnalyzer**, all values will be given relative to the current working frame.



Query Menu

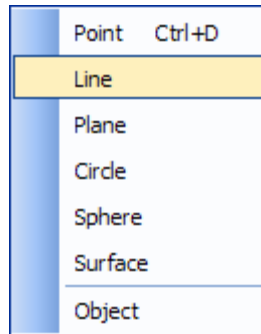


Example Point to Plane Query Results

GD&T

Point to

A single point may be compared to a variety of geometries. These are shown in the Point Query menu shown below

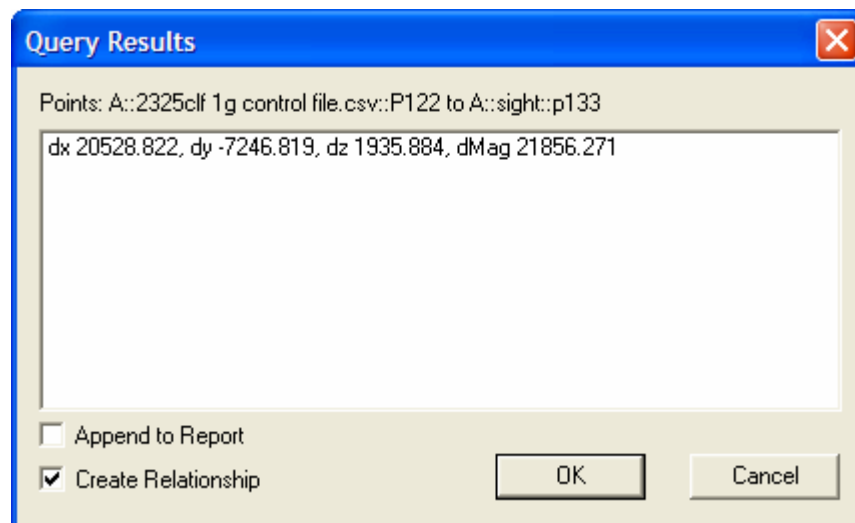


Objects for Point Queries

Point (Relationship)

Ctrl+D

The distance between points and the components relative to the working frame are presented on a dialog.

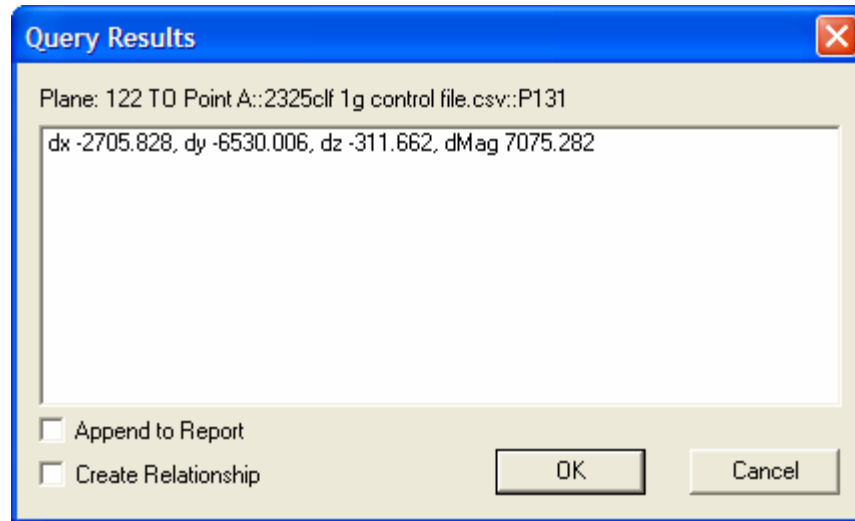


Point to Point Query results

The point to point query also allows you to compare one point to any other point. Check the Append to Report option to have these results posted to the Fit/Query Report. Check the Create Relationship option to track the differences between the points as a dynamically updated relationship. The relationship is automatically added to the Relationship branch on the treeview.

Point to Object Query

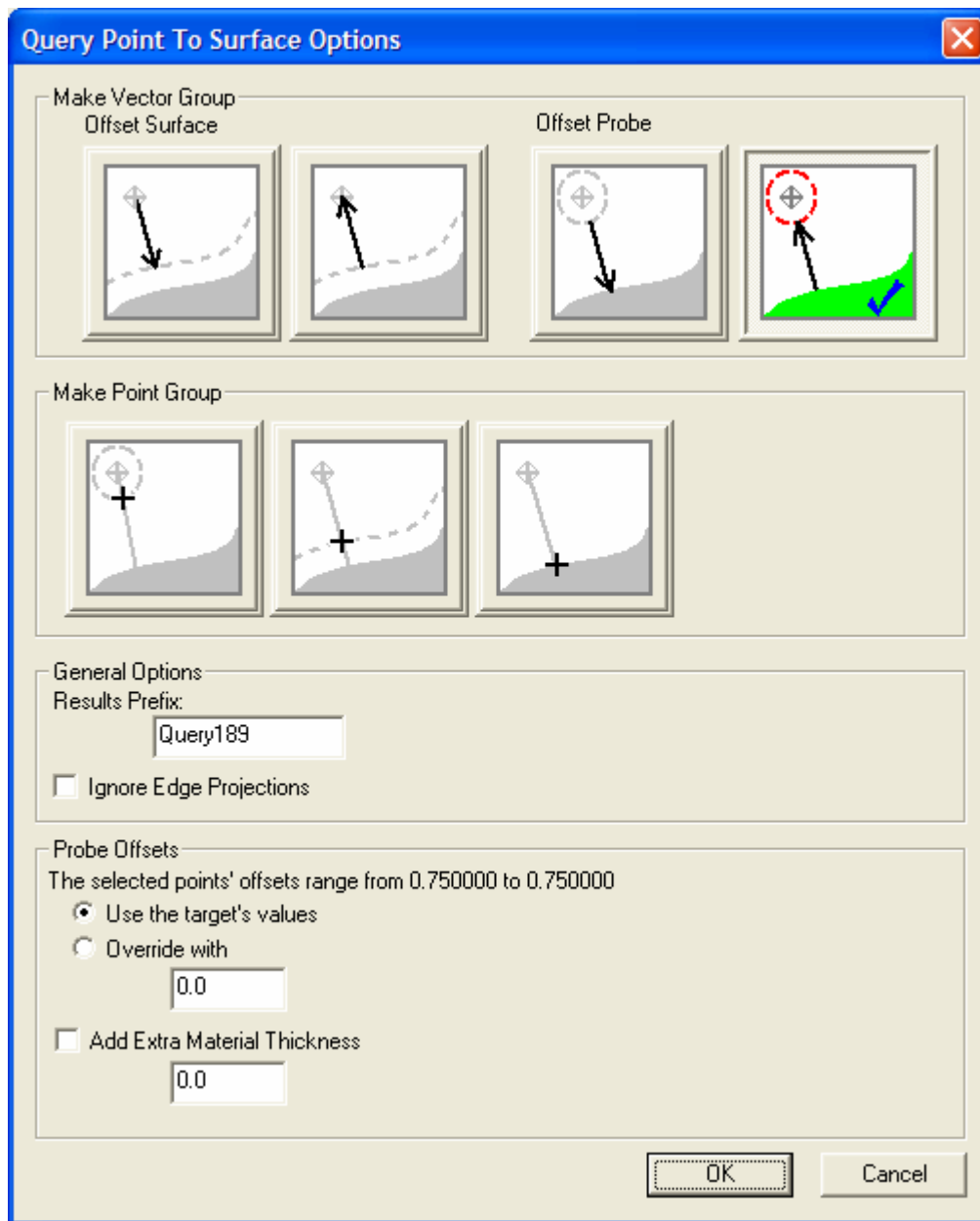
The results from a point to object query is presented in a Query Results dialog. Check the Append to Report option to have the results posted to the Fit/Query Results report. The Create Relationship option establishes a dynamically updated relationship on the treeview.



Point to Object Query results

Points to Object

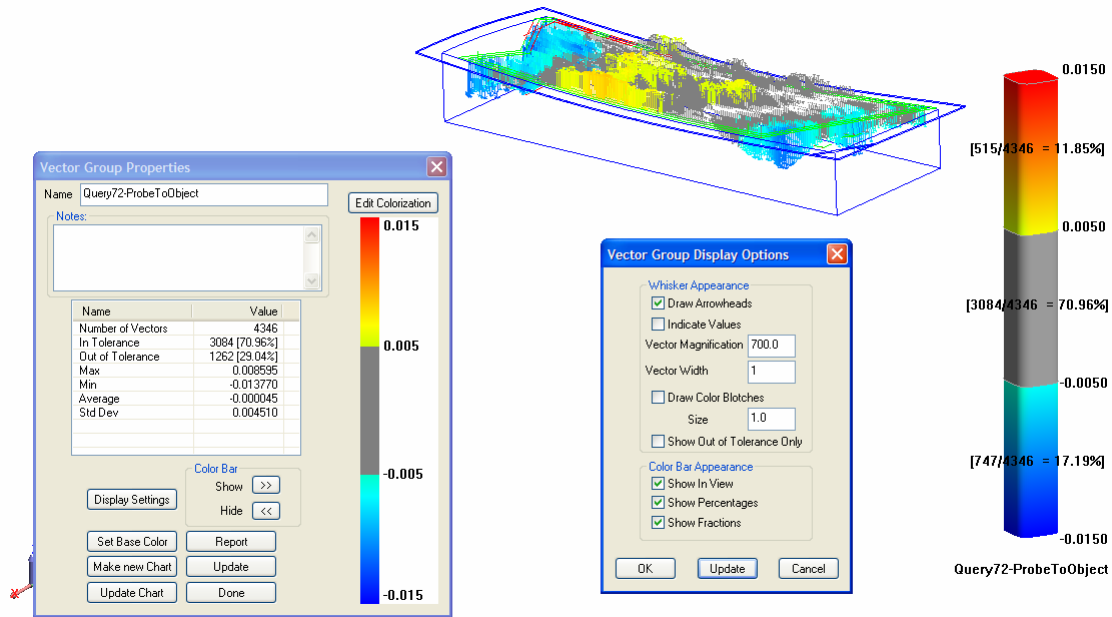
This option allows you to compare many points to geometry. The first step is to define the projection direction and offset values. An addition option on the Query Point to Object Options dialog allows for creating point by projecting them onto the object. An example of the projection options dialog is shown below.



Query Point to Surface Options

Select one or more of the projection options from the first row. Points can be constructed at specific offsets with Make Point Group option in the second row. Select one or more of these options if you need to construct points with these properties.

The results are displayed in a vector group properties window. An example of a multi-point query is given in the figure below.



Vector Group Properties dialog

Points to Points

Points to points query results in a distance map between each combination of points. An example inter-point distance map is shown below.

Inter-Point Distance

Print Excel Done

Precision: 4 WCF = A::WORLD

	A::2325clf 1g control file.csv::P121	A::2325clf 1g control file.csv::P122	A::2325clf 1g control file.csv::P129	A::2325clf 1g control file.csv::P130	A::2325clf 1g control file.csv::P131	A::2325clf 1g control file.csv::P132	A::2325clf 1g control file.csv::P141G
A::2325clf 1g control file.csv::P121		6639.9727	6639.2833	982.9746	5300.5303	578.4959	23699.6131
A::2325clf 1g control file.csv::P122	6639.9727		1227.2200	6833.4339	1470.7836	6153.4863	27311.3115
A::2325clf 1g control file.csv::P129	6639.2833	1227.2200		6654.3411	2150.7080	6208.2065	27439.9671
A::2325clf 1g control file.csv::P130	982.9746	6833.4339	6654.3411		5603.0517	1379.4620	23768.6284
A::2325clf 1g control file.csv::P131	5300.5303	1470.7836	2150.7080	5603.0517		4787.4461	26524.9771
A::2325clf 1g control file.csv::P132	578.4959	6153.4863	6208.2065	1379.4620	4787.4461		24052.2384
A::2325clf 1g control file.csv::P141G	23699.6131	27311.3115	27439.9671	23768.6284	26524.9771	24052.2384	

Points to Points Query results in Inter-Point Distance Map

Points to Object

After selecting both the points and the object (e.g., plane), the results will be displayed in the Vector Group Properties window shown above.

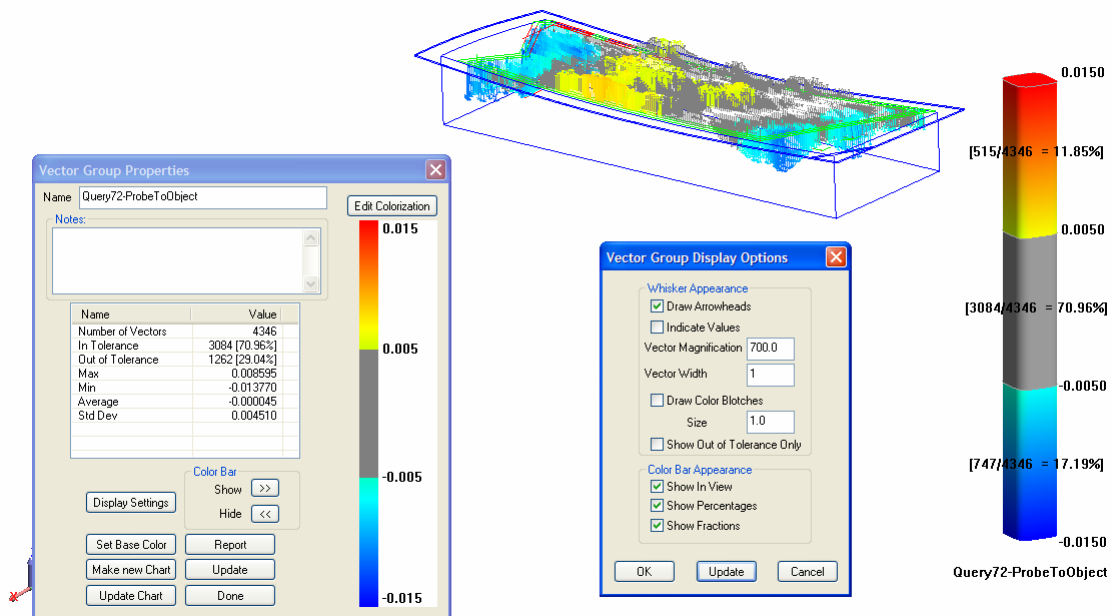
Points to Surfaces

When querying points to a surface, you have two options. Both options project the points to the analytical model of the surface and return the offset. The major difference lies in how the results are displayed in the graphical view.

In both cases, the graphical view is colorized according to the errors. A dialog, shown in the figure below provides a legend for the colors. There are several methods for colorizing, zero referenced colors and the full color range.

Whisker Plot

This option will produce whisker lines in the graphics to represent the errors at each point. This option will also allow you to select several surfaces to compare the points to. In the case of multiple surfaces, the



Vector Group Properties and Colorization Dialog

Line

There are several query options for lines. They will be discussed below.

Length of Line

This option will allow you to select a line. The program will respond with its length. As with all other queries, you will have the option of adding the results to the Fit/Query page.

Line to Line Angle

After selecting two lines, this option will compute the angle between the lines. This angle is determined based on the axis defined by the cross product of the two vectors. The angle returned is the amount you must rotate one of the lines about the axis to point it in the direction of the other line.

Line to Plane Angle

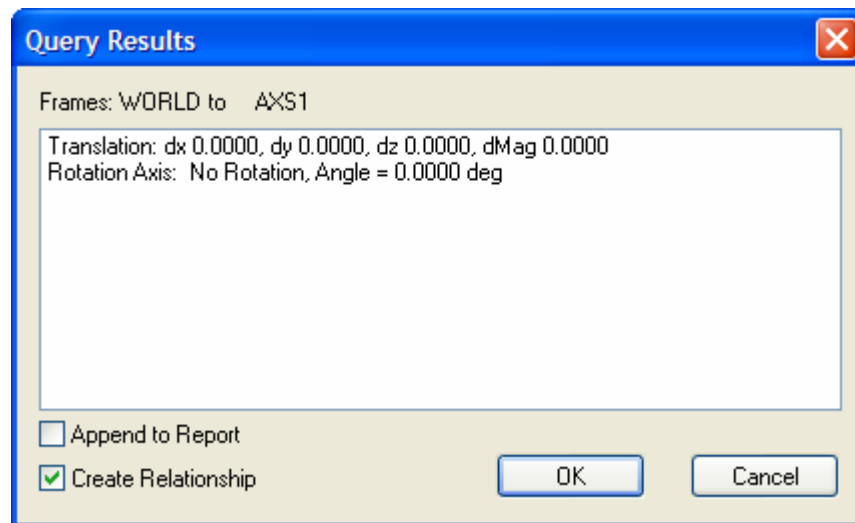
After selecting two lines, this option will compute the angle between the lines. This angle is determined based on the axis defined by the cross product of the two vectors. The angle returned is the amount you must rotate one of the lines about the axis to point it in the direction of the other line.

Plane to Plane Angle

Returns the angle between two planes.

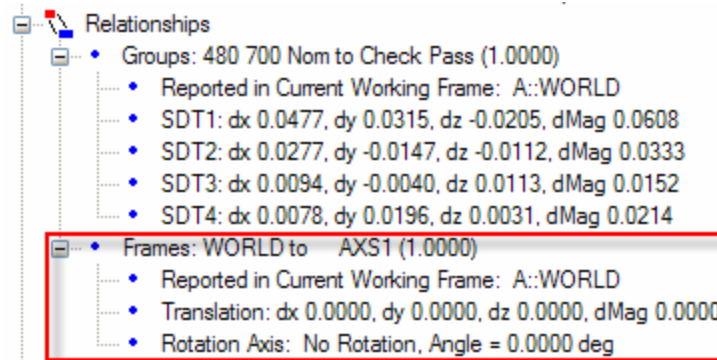
Frame to Frame (Relationship)

Frame to Frame relationship show the difference between two frame objects. The relationships is created with an option on the Query dialog. Select the Create Relationship option on the dialog.



Frame to Frame Query Results Dialog

The relationship is saved on the treeview. Open the branch on the tree to see the current state.

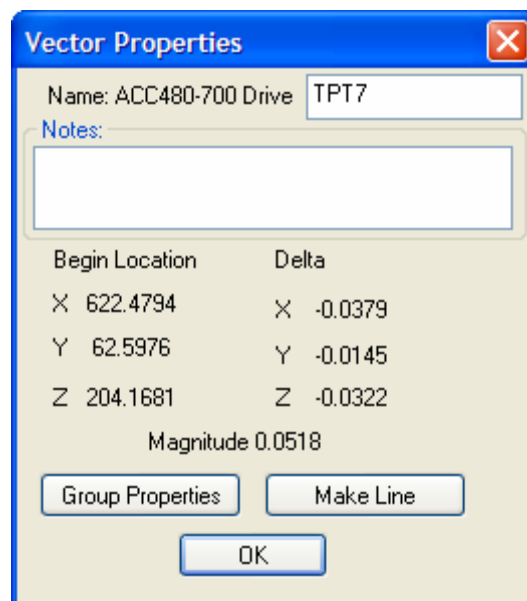


Frame to Frame Relationship

Relationship reports are available by right-clicking on the relationship and selecting the Quick Report option.

Group to Point

A series of vectors is created between each point in the group and the other point. Double clicking on the vector yields the option to convert the vector to a line or review the Vector Group Properties for all of the vectors created with the query.



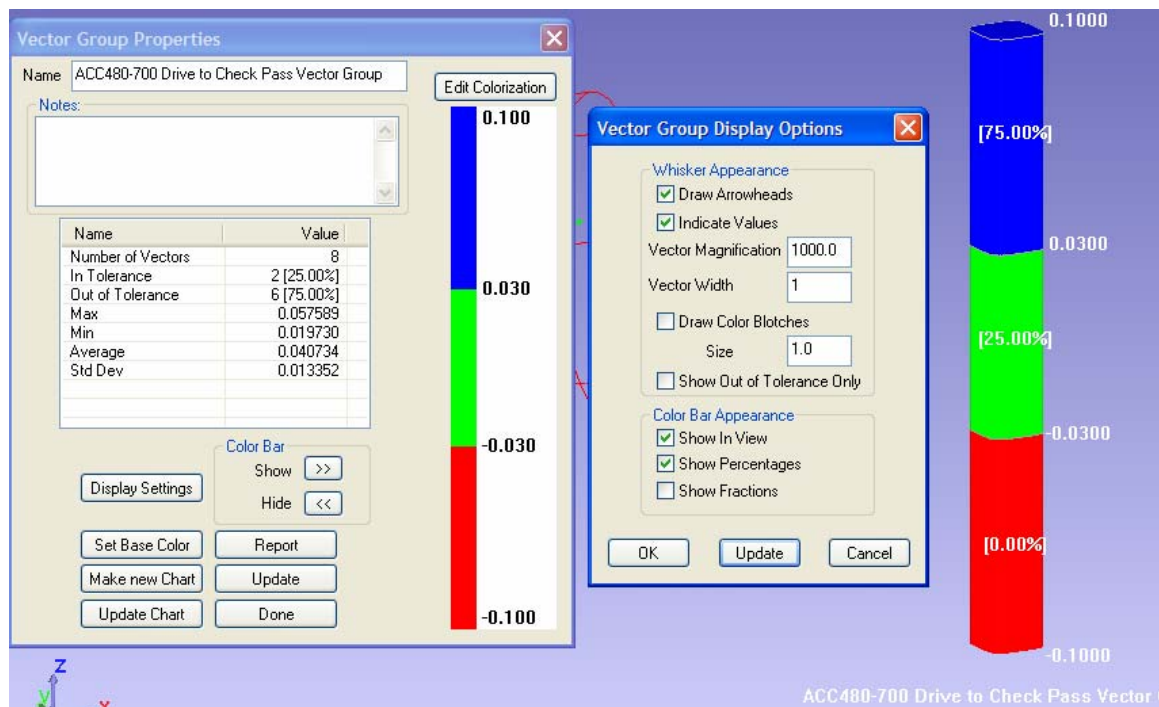
Vector Properties Dialog

Group to Group

There are options for the type of analysis output you need when comparing Groups to Groups.

Make Vector Group (whiskers)

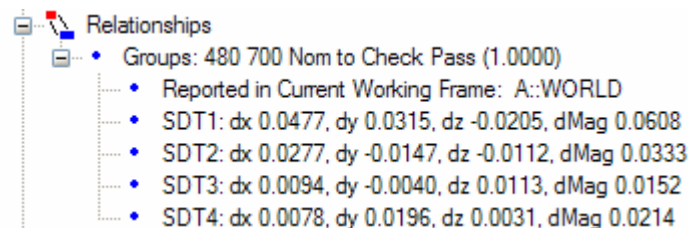
When making vector groups between point groups the output is the typical Vector Group properties dialog. An example is shown below.



Vector Properties Group to Group Dialog

Add a Group to Group Relationship

This option creates a group to group relationship. Group Relationships are traditionally used for outputting structured point to point comparisons. Relationship reports are done by right-clicking on the relationship and selecting the Quick Report option.

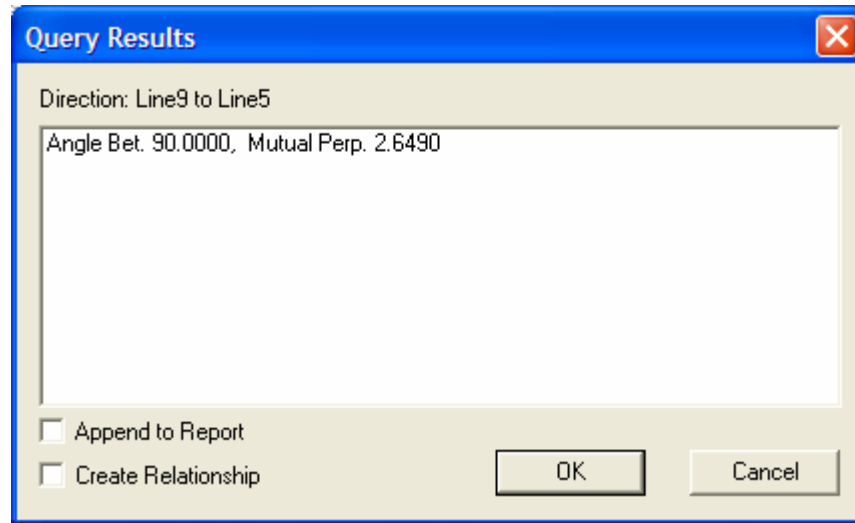


Group to Group Relationship on Treeview

Object to Object Direction (Relationship)

Defining the relationship between two objects can be a powerful alignment and analysis tool. This relationship will dynamically monitor the differences in direction between two

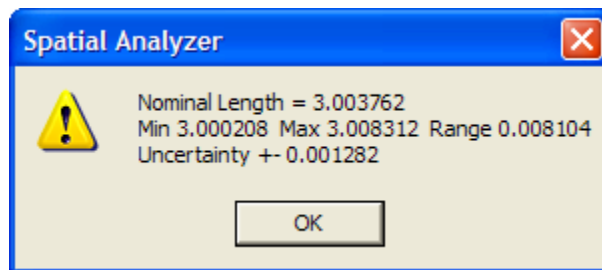
lines within an assembly. An example analysis between two lines is shown below in the dialog.



Object to Object Direction (Relationship)

Sensitivity Cloud to Sensitivity Cloud

Using the **SpatialAnalyzer** coordinate uncertainty computation capabilities, you can create a cloud of points around a measured point. These points represent the “cloud of uncertainty” of the coordinate. If you have clouds created for at least two points, you can determine the uncertainty of the line between them. The results are presented in the format shown the figure below.

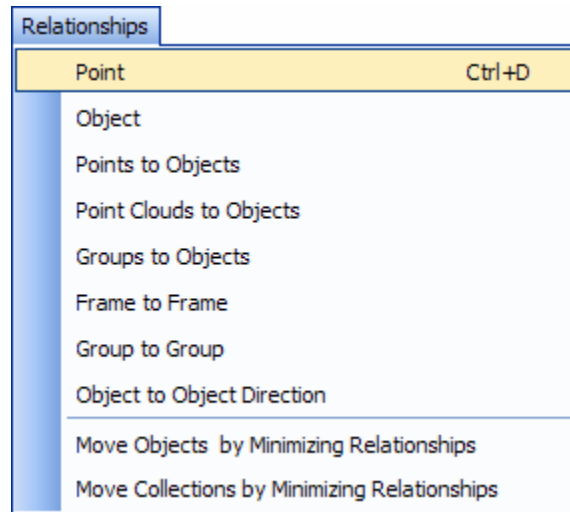


Point to Point Distance Uncertainty

Relationships

Relationships can be used for automatically updating dynamic changes between objects, make creating vector groups for analysis faster, provide a quick overview of the status of build operations and enable you to optimize your measurement directly against design constraints. By optimizing relationships they can provide a special type of spatial transformation capability. By simultaneously optimizing (i.e., best-fit) relationships i.e., the distance between (measured) points and known geometric shapes (e.g., points, lines, planes, and surfaces) the process fits measurements directly against there nominal features. The goal of the optimization is to find the spatial orientation of the

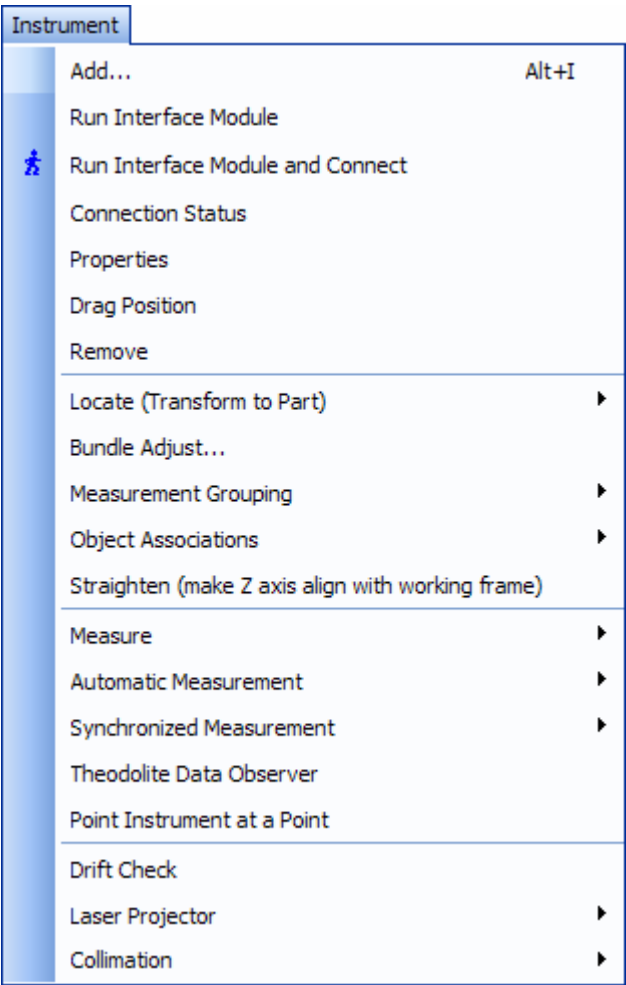
measured points such that the relationship or group of relationships (i.e., distance between the points and their geometric shape) is minimized. The relationships of that can be used for fitting include those listed in the figure below.



Relationships menu

Relationships and relationship optimization is covered in a preceding section specifically on Relationships.

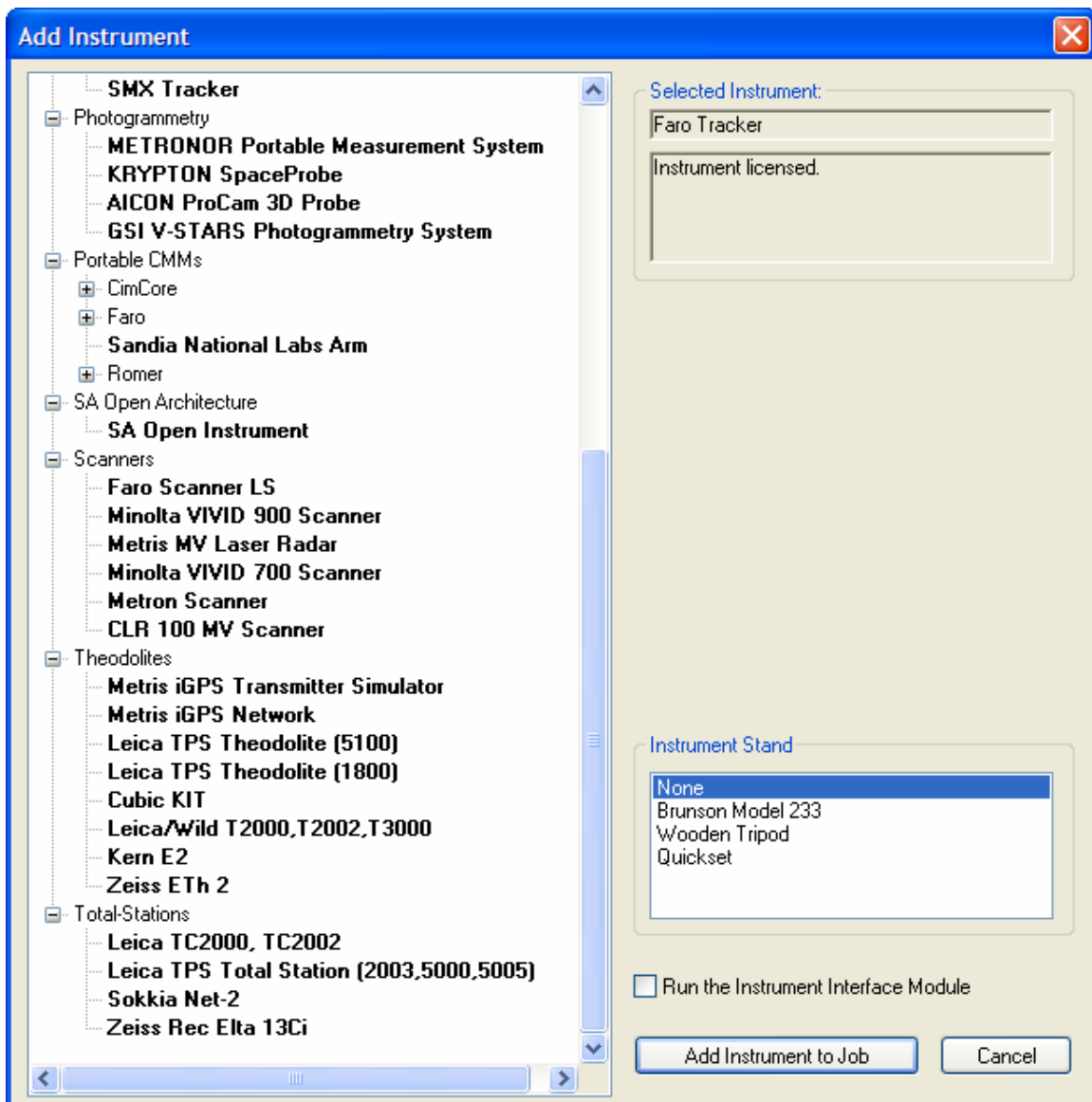
Instrument



Instrument Menu

Add... Alt+I

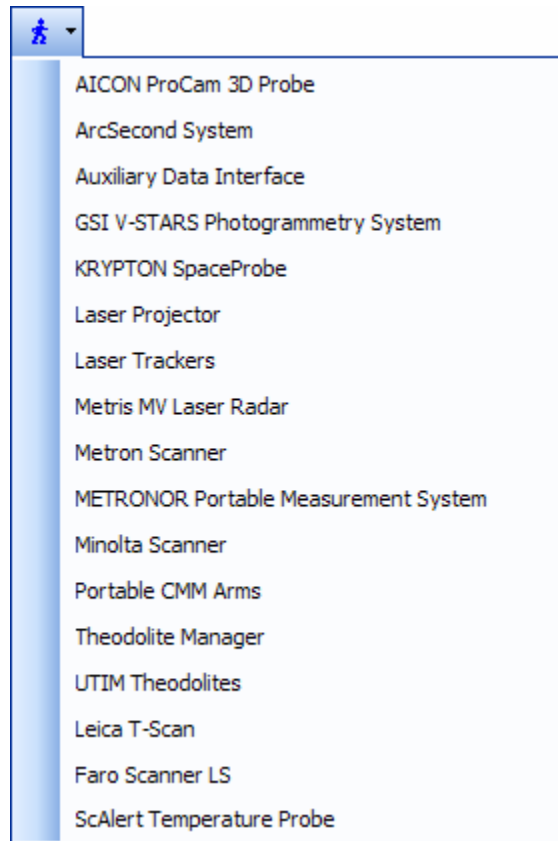
This option allows you to add an instrument to the model. This step is required in order to accept measurements from the device. After selecting this option, you will be able to select the instrument using the dialog shown below. After selecting the instrument and stand, it will be placed in the model 50 units past the previous instrument along the X-axis of the World coordinate system.



Add Instrument Dialog

Run Interface Module

This function drops a list of interface models to run. Select the appropriate option from the list. An example of the dropdown list of instrument interfaces is shown in the figure below.



Run Interface Model dropdown list of Instrument Interfaces

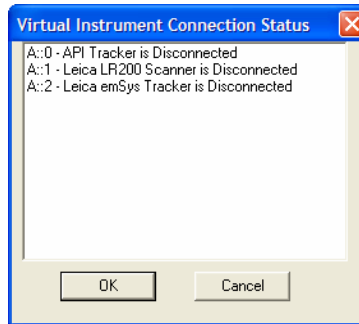
After selecting the interface to run a logon dialog is displayed. This option allow you to select which instrument interface to connect to and several of the optional connection parameters.

Run Interface Module and Connect

This function drops a list of interface models to run. Select the appropriate option from the list. This function attempts to automatically connect to the instrument with the settings from the last successful connection.

Connection Status

This option will display the Connection Status dialog. It indicates which of the instruments in the model are currently connected to their instrument interfaces through the TCP/IP connection. This is not to be confused with the instrument driver's connection to the physical device. If you turn the labels on in the user interface, you will see the word "live" along with the instrument number in the graphical view.



Instrument Connection Status


Properties


After selecting the instrument, you will see the following dialog. This dialog will allow you to move the sensors on the instrument, change the transformation, and adjust the uncertainty characteristics of the device.

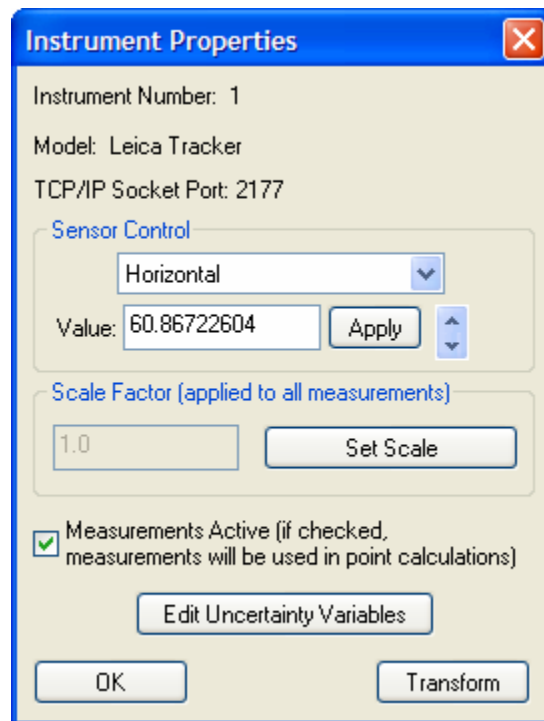
The Sensors area allows you to select each axis of the instrument and control its current position.

The scale factor field allows you to scale all measurements from this device during the process of point computation. For non-distance measuring theodolites, this value has no effect on the computed points.

The measurements active check box allows you to eliminate all the measurements from a particular instrument from the point computation.

The  button displays the transformation dialog and allows you to move the instrument relative to the current working frame.

The  button allows you to change the uncertainty parameters for the axes of the instrument. This dialog is shown below. These variables are used in combination with Gaussian distributed random numbers to compute many points for a measurement. These points are then processed statistically to determine the coordinate uncertainty values for each axis.



Instrument Properties

Instrument Number: 1
 Model: Leica Tracker
 TCP/IP Socket Port: 2177

Sensor Control

Horizontal

Value: 60.86722604 Apply

Scale Factor (applied to all measurements)

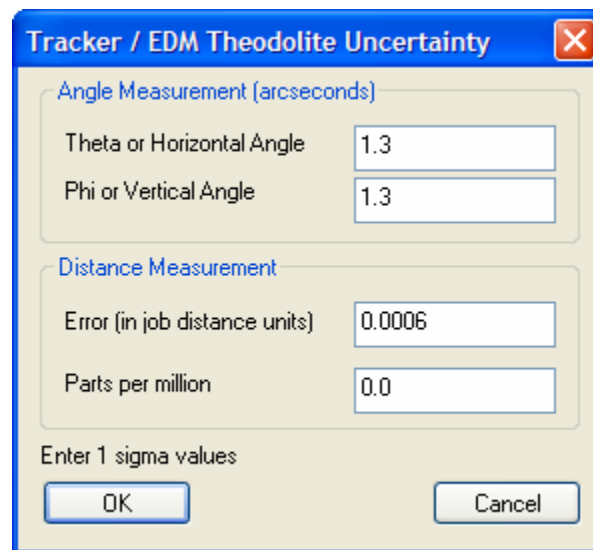
1.0 Set Scale

☒ Measurements Active (if checked, measurements will be used in point calculations)

Edit Uncertainty Variables

OK Transform

Instrument Properties Dialog



Tracker / EDM Theodolite Uncertainty

Angle Measurement (arcseconds)

Theta or Horizontal Angle: 1.3
 Phi or Vertical Angle: 1.3

Distance Measurement

Error (in job distance units): 0.0006
 Parts per million: 0.0

Enter 1 sigma values

OK Cancel

Instrument Uncertainty Settings Dialog

Drag Position

This option allows you to change the position and orientation of the instrument by dragging the mouse in the graphical view. This is often used to roughly position theodolites before performing a bundle adjustment operation. This positioning step is

required since the bundle operation is heavily dependent on the starting guess for the instrument transformations. One of the easiest ways to orient the theodolites properly is to point them at one another and query their angles. Next, drag the instruments until the measurement rays point at the other instrument.

After selecting this menu option, you will have to select the instrument you wish to move. Once you have selected it, the cursor prompt will instruct you on how to drag the instrument. Hold down the left button to drag the instrument in the plane of the current view. Hold down the right button and drag around the instrument to rotate it about the axis normal to the view. When you get the instrument to the proper location, press ENTER to end the drag mode.

Remove

Use this option to remove an instrument from the model. Since the measurements from the instrument require its existence, they will be deleted along with the device.

Locate (Transform to Part)

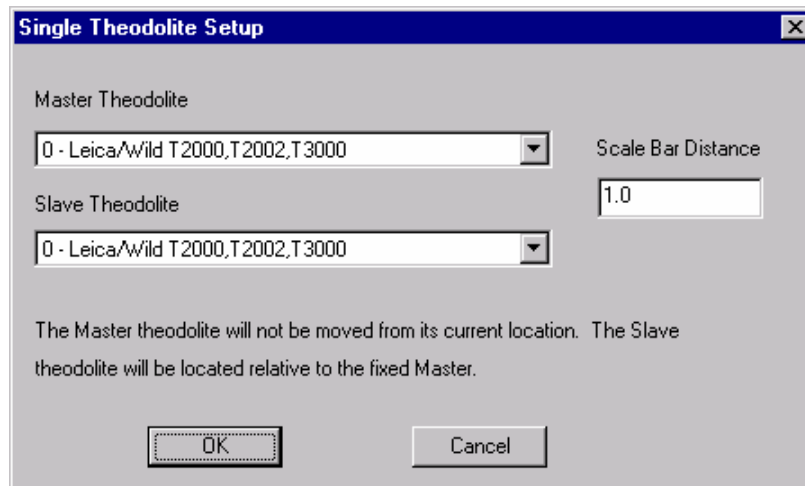
The Locate Instrument routine is covered in detail in the Least Square Transformation section.

Bundle Adjust...

This option will initiate a bundle adjustment. This process is described in detail in the of this manual.

Single Theodolite Setup

In some cases, you may wish to setup a single theodolite relative to another by collimating the theodolites instead of bundle adjusting the system. Though it is recommended that you drag the instruments to their proper position and bundle the system, this option is provided for cases where a bundle is not desired. After selecting network setup, the dialog shown below appears. You then select a fixed and a moving instrument. It will then send the appropriate instructions to each instrument interface. Remember that the single theodolite setup procedure is only necessary if you are not planning on bundle adjusting the system. You may still bundle the theodolites on common targets at anytime after using this procedure.



Single Theodolite Setup Dialog

Synchronized Measurement

Master - Slave

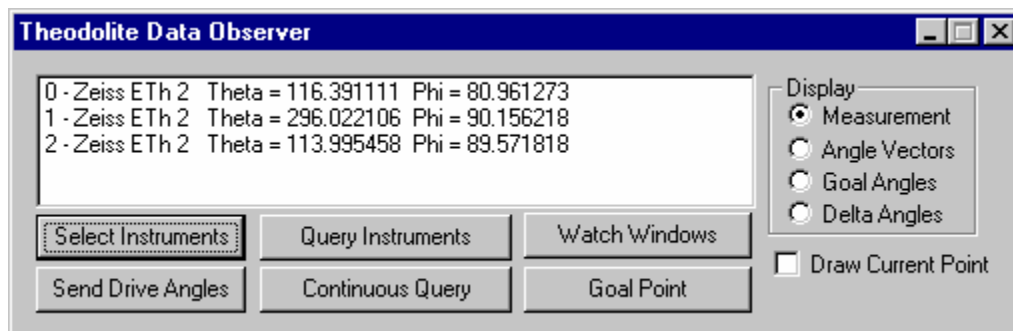
Polling

Theodolite Data Observer

In the case of theodolites, multiple devices must be used to create a single coordinate value. The TDO performs this function. It allows you to then create watch windows using the current point computed based on all instruments being updated in the TDO.

The TDO will update all data and associated watch windows whenever an instrument it is monitoring submits a measurement Update (or Query response). The instrument interfaces may be instructed to query the instrument, or you can perform this operation from the TDO interface.

To run the Theodolite Data Observer, go to the Instruments menu and select Theodolite Data Observer. It will appear as shown below.



Theodolite Data Observer Dialog

Only instruments that are currently active (or connected) will be listed in the data window. To change the list of selected instruments, press the Select Instruments button.

Auto Measure Mode

Auto Measure Mode is used to automatically measure a group of points by first transforming into the coordinate system using known anchor points. To run this mode, select the instrument you wish to use and then the group of points to measure. The points will be placed in the dialog shown below.

surf1::	Stat	Norm. X	Norm. Y	Norm. Z	Act. X	Act. Y	Act. Z	Delta
p0		17.2421	10.7634	16.2345				
p1		17.4415	10.4225	16.2637				
p2		17.6024	10.0617	16.2406				
p3		17.7036	9.6704	16.2834				
p4		17.7750	9.2635	16.1719				
p5		17.7829	8.8211	16.1310				
p6		17.6874	8.3225	16.0049				
p7		17.5083	7.8323	15.9204				
p8		17.2724	7.4574	15.7358				
p9		16.9975	7.1531	15.5742				
p10		16.6884	6.9159	15.3777				
p11		16.4560	6.8552	14.9717				

Auto Measure Mode Dialog

Auto Measure Mode (streamlined)

The mode provides a minimal interface when measuring points in a sequence.

Auto Measure a Vector Group (multi-measure)

Measure Batch of Points

This option allows you to select points and transmit their group and name to the instrument interface. The idea is that you can automatically name points and group names while you measure simply by selecting reference points from an SA template or CAD file.

AutoCorrespond with Proximity Trigger >> Points ... Vectors

This mode is similar to the laser radar SVI mode. It provides a method to have named drive points and or vectors on a part when measuring with instrument with target offsets, e.g., Trackers, CMM Arms, and Probe Targets. When the operator sweeps the SMR over the surface the interface will automatically grab a measurement when they get within a specified tolerance.

To use it start the Tracker interface and go to the Instrument then start this measurement mode for either point or vectors. Select a point or vector group and enter trigger tolerances. Points/vectors are measured when the probe is within the tolerance and they are automatically named to match the closest point. The measurement are recorded into a measurement group.

Point trigger targets are evaluated radially around each point in the reference point group. When measuring against a vector group both the point and vector trigger tolerances are used in the evaluation. The probe must be within the point trigger and vector trigger tolerances before a point is recorded. The point trigger is evaluated radially from vector's begin point. This ensures the measured points get tied to the right vectors.

A point is recorded when the target gets inside the trigger tolerance to any of the points in the point group. The measurement mode keeps recording points as the probe continues to get closer to the target points. Only the measurements that are closest to each target point are saved.

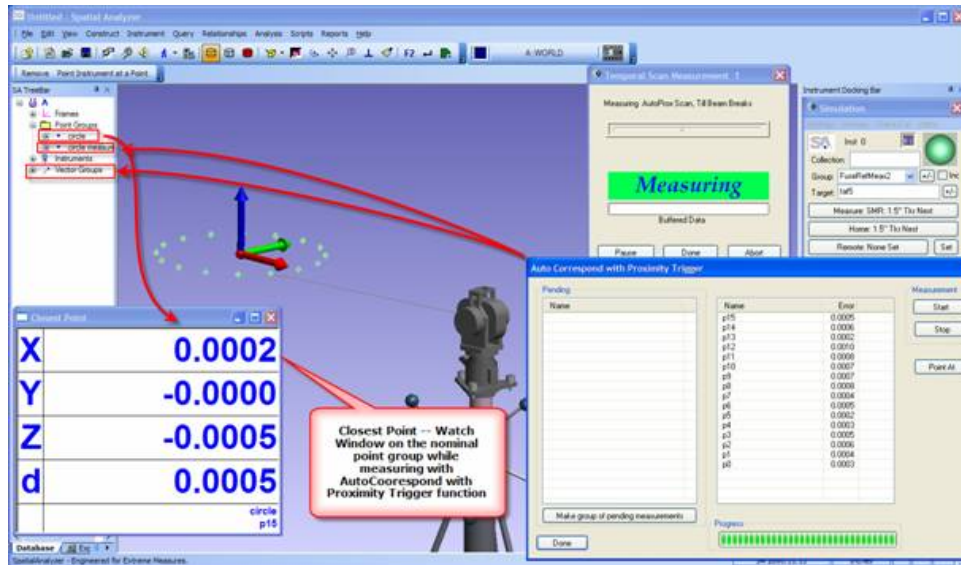
This mode is used with a tracker or PCMM Arm to collect repeatable data that is automatically named. It enables the solution to have a naming convention and pre-configures the points/vectors needed for analysis. Additionally it can limit the amount of points that are collected to the important ones. Time savings on filtering and wondering whether enough data was collected are minimized. Using a drive file means data collected by different operators and companies is consistent.

The interface in SA shows two lists ... points/vectors "to measure" and ones that have been "measured." The operator's job is to sweep the reflector on the surface collecting points which moves them from the "to measure" list to the "measured" list. A weasel mode (PC speaker) lets the operator know how close the reflector is to points or vectors.

One of the options when setting up the Auto-Correspond with Proximity Trigger is to have it create a vector group as it measures. So while the user measures it records the deltas automatically.

Watch Windows with Auto-Correspond with Proximity Triggers

Use the Auto-Correspond with Proximity Trigger and a watch window up at the same time. The measurement mode sends SA Watch Updates as it measures so watch windows are updated continuously. The figure below shows the process.



Consider making the proximity zone a wider than the application may require. The function automatically uses the point closest too the nominal drive point.

Create Vectors while Measuring with Auto-Coorespond with Proximity Trigger modes

With this option the user does not have to make the query. It is automatically done for them as the targets are measured. Exporting the deltas to Excel or other applications is straight-forward with a Vector Group.

Auto Measure Proximity Trigger Properties

Group to contain measurements

Vector Group to hold results while measuring
☒ Make Vectors while measuring
 Vector Group:

Options:
☐ Project results to nominal vector

Trigger Parameters
 Point distance threshold:
 Vector axis threshold:

Drive tone (warbler)
 Tone ramp start zone:

OK Cancel

**Make vectors of
the dx, dy, and dz
component as
each point is
measured**

Point Instrument at a Point

Analysis

Analysis

- Re-Compute Targets From Shots
- Re-Compute Hidden Points
- Activate All Measurements
- Best-Fit Transformation ▶
- Undo Last Best-Fit Transform
- Temperature Compensation
- Measurement Simulation ▶
- Coordinate Uncertainty ▶
- Pipe Relationships Module
- Robot Calibration and Uncertainty Analysis

Analysis Menu

Re-Compute Targets From Shots

Whenever a measurement comes into **SpatialAnalyzer** from an instrument interface, an attempt is made to compute the location of the point. If there are other measurements at the point, they will be included in the computation. If not, and the measurement does not contain complete coordinate information (theodolite shot for example), the point will not be computed until another measurement is taken.

Since the computation occurs when the measurements come in, all of the targets are computed based on the most current information. If, however, you move an instrument, the targets are no longer properly referenced to the device. You must use this menu option to re-compute all the point locations based on the measurements from the instruments.

The effect will be seen as follows: Suppose you have 10 points measured with a laser tracker. If you move the laser tracker, the points will remain fixed. The measurement rays from the tracker will, however, move with the device. If you select this menu option, the points will move to their new location based on the new device position and the measurement rays.

Re-Compute Hidden Points

Hidden Point Bar computation is discussed in more detail in the User Options page section. Basically, a hidden point is computed using two measured points and extending a vector to the third, or hidden, point. When points are measured using a special naming convention for hidden points, the third point is automatically computed when the two points have been measured. If the points used to compute the hidden point are moved, this option will allow you to automatically update all automatic hidden point computations.

Operations such as a Bundle Adjust will move the points used for hidden point computation. For this reason, it is suggested that you run this command after a bundle adjustment if you have hidden points in the model.

Activate All Measurements

Measurements may be tagged as either active or inactive. If active, they are used in the process of point computation. If inactive, they remain in the database, but are not used for point computation.

You can inactivate measurements using the tree-view, or by double clicking on a point and selecting Details. This option activates all measurements in the database.

Best-Fit Transformation

This option allows you to perform best-fit transformations for a group of points. There are two options, Group of Points or Surfaces. The best-fit transformation process is described in more detail in the Best-Fit Transformations of this manual.

Undo Last Best-Fit Transform

This option allows you to undo the most recent best-fit transformation for this document session. If you close the job file and re-open it, you will not be able to use this option. It

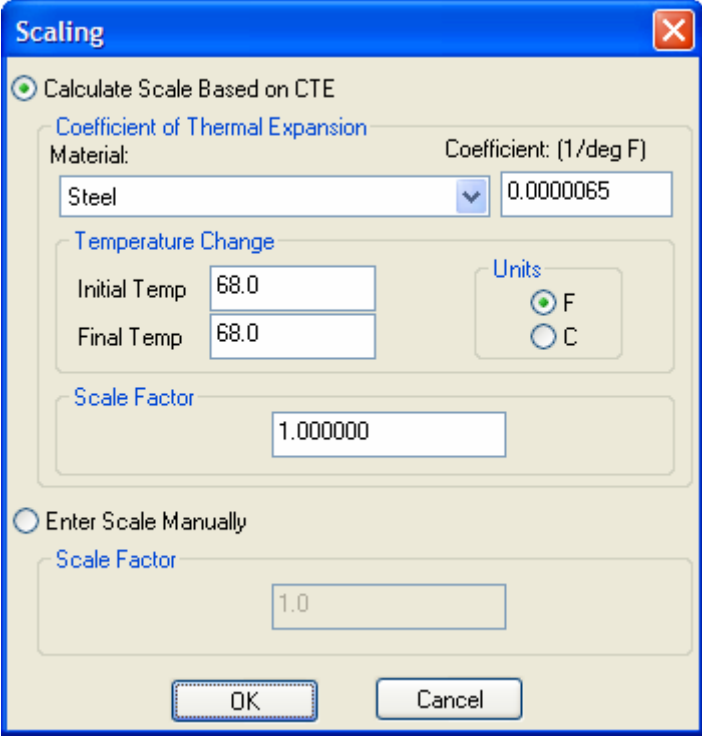
is intended more for the case where a best fit is performed and immediately thereafter, you need to undo it.

In addition to this method of undo, remember that **SpatialAnalyzer** backup files are written throughout the job. Before a best-fit transformation is performed, a backup file is written to the backup directory. You can revert to this file if necessary.

Temperature Compensation

Often, it is necessary to scale a series of points based on changes in temperature. This scaling is, of course, performed equally in all axes although this does not exactly match behavior in real-life. This approximation is commonly used in metrology applications so it is supported in **SpatialAnalyzer**.

After selecting this option, you will select the group of points you wish to compensate. After making your selection, the dialog shown below will appear.



The image shows a software dialog box titled "Scaling" with a close button (X) in the top right corner. The dialog has two main sections. The first section, "Calculate Scale Based on CTE", is selected with a radio button. It contains a "Coefficient of Thermal Expansion" section with a "Material" dropdown menu set to "Steel" and a "Coefficient: (1/deg F)" text box containing "0.0000065". Below this is a "Temperature Change" section with "Initial Temp" and "Final Temp" text boxes, both containing "68.0". To the right of these is a "Units" section with radio buttons for "F" (selected) and "C". Below the temperature section is a "Scale Factor" text box containing "1.000000". The second section, "Enter Scale Manually", is unselected with a radio button. It contains a "Scale Factor" text box containing "1.0". At the bottom of the dialog are "OK" and "Cancel" buttons.

Temperature Compensation Dialog

Coefficient of Thermal Expansion

In this area of the dialog, you will select the Color and temperature change you wish to apply. Notice that when you select a Color, the coefficient will appear in the coefficient field. If, instead of selecting the coefficient, you wish to enter the coefficient, you may do so.

The information in this area of the dialog will be used to determine the scale factor that will be applied to the points.

Coordinate Frame Anchor Point

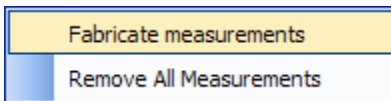
In order to properly scale the points, you must select the origin point about which they will be scaled. To do this, select the appropriate frame.

Group Information

When the points are compensated, you have the option of making a copy of the points so the originals are not tampered with. If you select this option, the default name of the new group will be set to the original group name plus the temperature differential.

Measurement Simulation

This menu contains options relating to the simulation of the measurement process. There are several options that are shown below.



Analysis Simulation Options

Fabricate measurements

Using this option, you can create measurements from an instrument to a point. This may be used to simulate a measurement session and observe the coordinate uncertainty that will result from the measurement geometry. One of the questions you will be asked after selecting the instruments and points is if you wish to inject random noise based on the instrument uncertainty values.

Remove All Measurements

This option will go through all of the instruments and remove all the measurements.

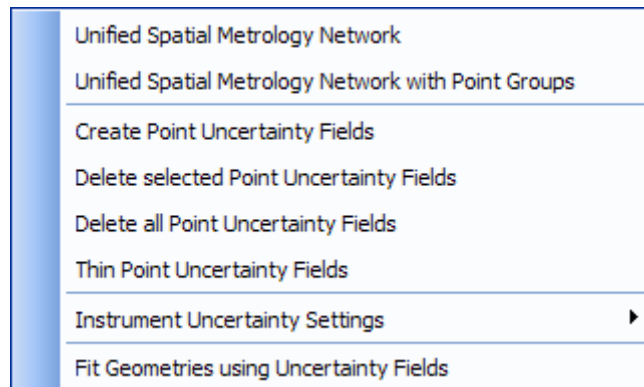
Coordinate Uncertainty

Coordinate Uncertainty options are covered in detail in several preceding s. A brief introduction follows however for detailed information on this topic please refer to the earlier s.

Large-scale manufacturing operations are increasing their reliance on portable coordinate measurement devices. These devices include electronic theodolite networks, total station systems, video photogrammetry systems, laser tracking systems, laser scanning systems, and portable coordinate measuring machines. This increase in portable metrology is driven by cost and efficiency. Commercial airframe construction is a good example of this trend. Instead of relying on elaborate holding fixtures and precise tooling, manufacturers are using the component parts as the tooling, and verifying design conformance with portable coordinate measurement devices¹¹. This design paradigm shift eliminates the need to manufacture and maintain complex, costly check-fixtures. It does, however, introduce portable metrology into the production

¹¹ Muske et. al., 1999

process and therefore requires a rigorous accounting of the uncertainty in the design conformance measurements.



Coordinate Uncertainty Submenu

Create Point Uncertainty Fields

After picking this option, you will be asked to select the points you are interested in. In addition, you can enter the number of points you wish to use in each cloud. Once you enter this information, the target sensitivity clouds will appear in the graphical interface.

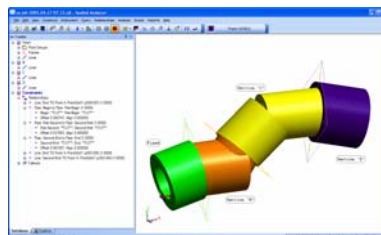
For more information on the scaling and display of these clouds, see the User Options , specifically the Analysis options.

Delete Point Uncertainty Fields

If you create uncertainty clouds using the Analysis menu or other methods, this menu option will allow you to clear them.

Pipe Relationships Module

Pipe Fitting Relationships is an optional SA module. It solves difficult pipe fitting, cut and alignment jobs. Optimally setting reference rings on individual components ensures cuts, milling and welding operations are secure and deliver joints within process capabilities. It provides a consistent solution for optimally cutting and orienting sections together. Getting fittings optimally aligned and cut correctly saves critical time and project resources.



Pipe Relationships take measurements of the components and the system constraints and return the optimal orientation for each section, coupling, fitting, and reference structure. SA's core measurement and analysis functionality enables teams to build/validate each component into place.

Nominal component locations can be imported from CAD or formed directly from measurements of each component. Each system component is setup independently within an SA Collection. Organizing the system within collections helps keep the process of setting and solving pipe relationships easier to manage.

Getting components aligned within process requirements is easier to accomplish with SA's graphics environment. Setup a Pipe to Pipe relationship between measured pipe center lines. Each pipe has a beginning and end along with an OD and ID. Cut plane regions for each pipe can be specified with a nominal frame or SA will propose cut planes for each component interface after the system couplings are optimized. Configure each Pipe to Pipe Relationship weights before optimizing. Pipe to Pipe Relationship weights include cut locations (e.g., axis offset, axis alignment and center pull – off nominal) along with Out of Material Condition weights (i.e., no metal penalties.)

Pipe Fitting Relationships use a Procedural Constraint-Reduction through Cut Process. Initially, all cut planes are free to move to optimal locations. As the process progresses (first is cut, second it cut, etc), gradually reduces the degrees of freedom and properly constrains the solution. The result is the measured pipe orientation to is fit to the model pipe system (e.g., measured points, sections, Relationships, couplings, reducers, expanders.) The user is able weight each Pipe Relationships individually relative to its importance to the system requirements. The process is relatively simple to implement and at the same time easy to visualize in SA.

Benefits of Pipe Fitting Relationships

- Pipe Relationships Optimization improves results over traditional processes
- Preparation, Setup, and Flow time are minimized
- Reduces measurement tolerance guess work
- Integrate assembly configuration for design/tolerance improvements
- Pipe Relationships between features are optimized directly → Build / Inspect
- Manage and Build custom configurations directly on the job
- Using the CAD Geometry in Pipe Relationships improves system conformance
- Pipe Relationship reporting improves conformance analysis

SA Robot Calibration Module

This section summarizes calibration of an industrial robot using the using the SA Robot Calibration module. SA Robot Cal refines the robot's kinematic model by analyzing the differences between commanded end-effector positions (joint poses) and surveyed actual end-effector positions.

SA's calibration module is capable of calibrating any serial manipulator. The basic steps are as follows:

- Describe the nominal kinematics of the robot
 - Create its D&H parameter table
 - Determine how to map the joint angles from the controller to the D&H angle convention.
- Measure a series of tool points using SA and any instrument, and store the robot's joint angles corresponding to each pose.

- Load the joint data into the SA calibration module.
- Run the calibration.
- Apply the modified D&H kinematic values to the robot controller's kinematic model.

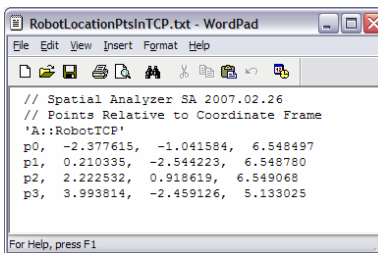
To illustrate SA's calibration process, this section will walk through a calibration on an ABB model IRB6400 M98.



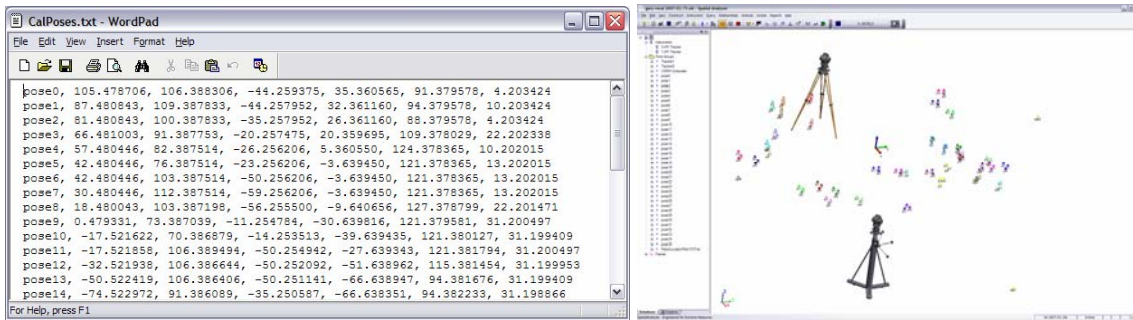
The survey is performed with either 3-D or 6-D measurements. During the survey, the robot's joint angles are recorded and paired with the survey measurements. Once the survey data collection is completed, the optimization is run. It takes as input the Denavit-Hartenberg (DH) kinematic model, the joint angles and survey results. Optimization outputs include the refined DH model and the robot's end-effector uncertainty.

Survey Data Collection

In our example, a tracker is used to collect 3 points on a calibration artifact attached to the end-effector. Our simple artifact features three holes that nicely accommodate repeatably nesting a 1.5 inch Spherically Mounted Reflector (SMR). The 3 points are characterized relative to the robot's Tool Center Point (TCP) in the attached file RobotLocationPtsInTCP.txt

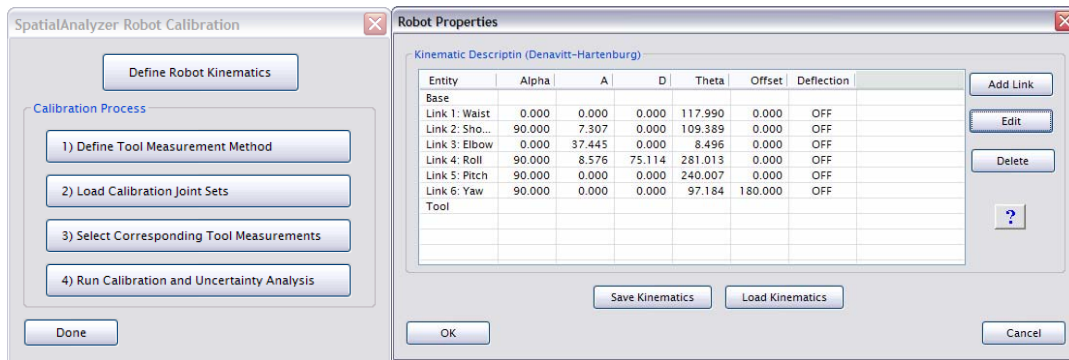


Calibration data is collected by commanding the robot to multiple positions, and for each position, collecting joint angles and survey data. These positions are recorded in the attached file CalPoses.txt



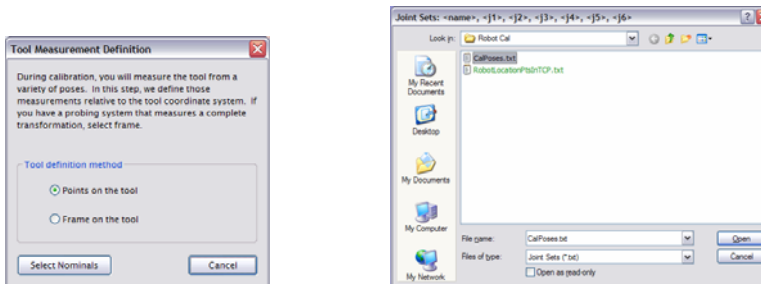
Optimization

Begin the optimization by selecting **Analysis>>Robot Calibration and Uncertainty Analysis** from the SA menu. The Spatial Analyzer Robot Calibration dialog lists the steps to be performed. First, the robot's kinematic model is defined (or imported), then measurement method is chosen, followed by loading the joint sets, selecting the survey data and finally calculating residuals.



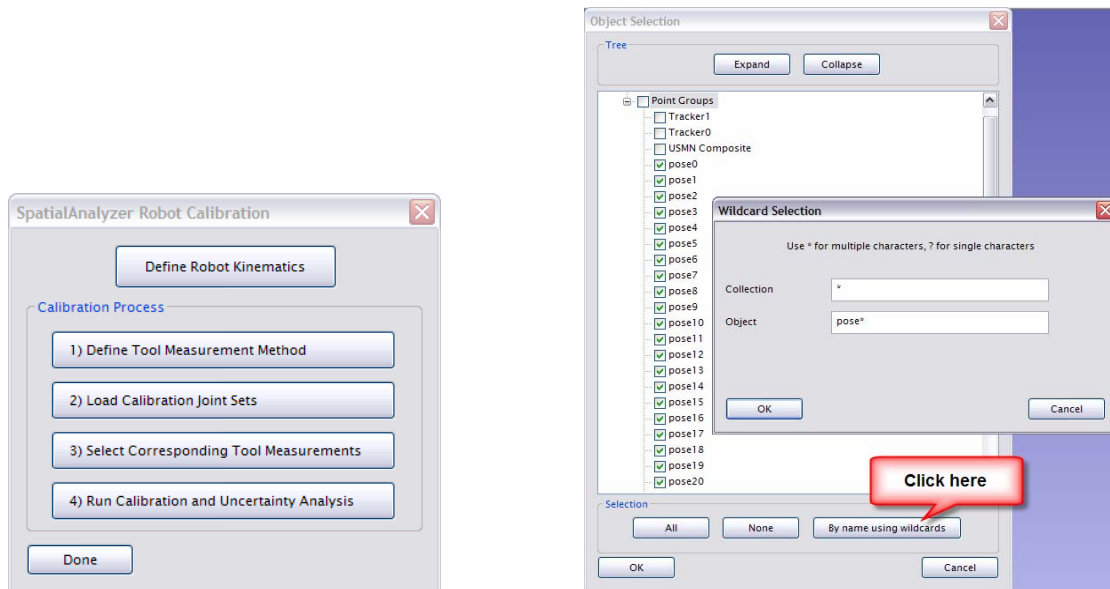
Beginning with defining robot kinematics, click the Load Kinematics button and select a SARobKin file. If the DH model were not available, the Add Link and Edit buttons can be used to manually define the DH model. (Note: this sounds much worse than it really is. See the subsequent DH Notation section and get a tape measure, if necessary. The input model doesn't have to be exact.)

Next, click Define Tool Measurement Method, select the Points on the Tool radio button and click Select Nominals. Select the frame Tool Frame for Nominals and the point group RobotLocationPtsInTCP.

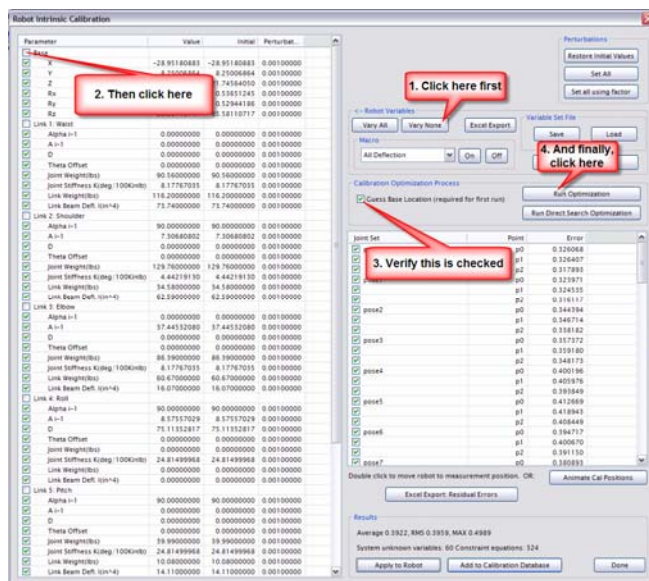


Next, click Load Calibration Joint Sets and open a file of poses. The dataset format is shown in the title bar. The name field corresponds with the surveyed point group names (or frame names if the Tool Measurement Definition had been selected as Frame on the Tool).

Next, click Select Corresponding Tool Measurements and press F2. Using the wildcard selection button, select all point groups matching the object string “pose*”. Don’t forget to press enter to complete the group selection after clicking “OK” on the Wildcard Selection and Object Selection dialogs.

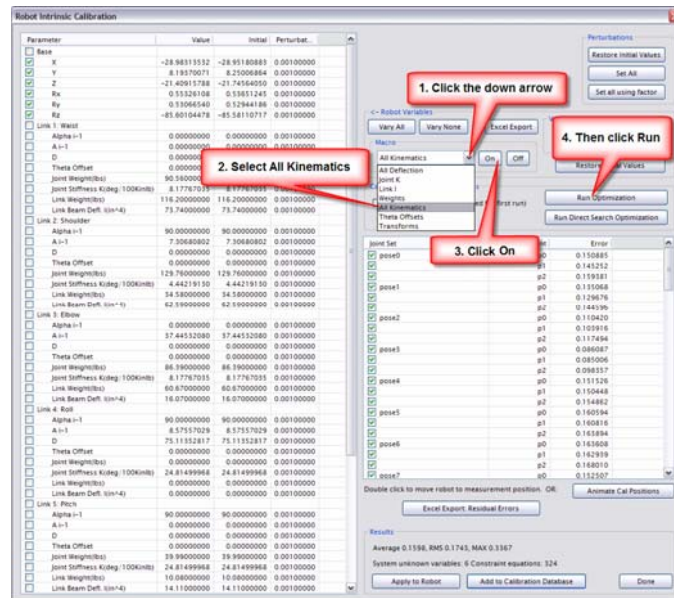


Now, Click Run Calibration and Uncertainty Analysis, and the following dialog appears.



First, we'll align to the robot's base, then we'll solve for the kinematic model. Solve for base alignment by clicking the four places shown in the following illustration. First, clicking Vary None removes all check marks in the Parameter column on the left. Then, clicking the first Parameter box, labeled Base, frees motion in 6 degrees of freedom for the robot base. The Guess Base Location configures the optimization for the first iteration, and finally clicking the Run Optimization button begins the calculations.

Once complete, the Guess Base Location checkmark has been cleared, and it's time to solve for the kinematic model. Enable All kinematics and run the optimization as shown. Results, on the bottom right, show the fit improvement from about a half of an inch to around twenty thousandths. Run it again, this time selecting Transforms, and the fit improves to about seven thousandths of an inch.



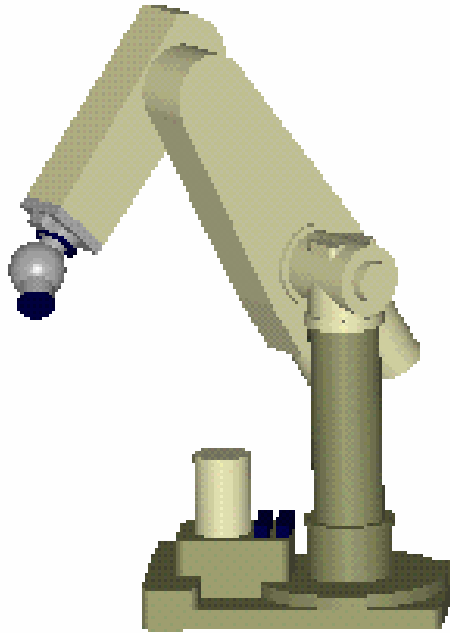
At this point, you could remove outliers if any exist, then resolve the optimization. When finished, click Apply to Robot, then close the dialog. Note that while in the dialog, you can double click on the poses to see the robot move to that pose in the SA graphics. You can also hit the Animate Cal Poses button to see the whole workspace quickly.

When calibration is complete, return to the main SA Robot Cal interface window, and hit the "Define Robot Kinematics" button. Then, you will have the D&H table with the values that should be applied to the robot controller.

Robot Kinematics

The first step in calibrating a manipulator is to understand its nominal kinematic parameters, and the controller's ability to accept new, calibrated values for these parameters. This section will describe how to represent a robot's kinematics. Most manipulators are serial devices, meaning they are composed of a series of links and joints connected in an open chain. Denavit-Hartenberg notation is a standard means for

specifying the kinematics of a manipulator. See the example table below, then a brief description of the meaning of the parameters.



D&H Table Example: Unimation Puma 260 Robot

Link	α	a	d	θ	Type
1	0.0	0.0	0.0	0	Revolute
2	-90.0	0.0	0.0	0	Revolute
3	0.0	8.0	-4.8	0	Revolute
4	-90.0	0.0	8.0	0	Revolute
5	90.0	0.0	0.0	0	Revolute
6	-90.0	0.0	0.0	0	Revolute

Links are rigid members; joints allow relative motion between the links. The chain starts at the base, or ground link, and ends at the working tool. Except for the tool link and the base, each individual link in the chain begins and ends with a joint. The tool may be a gripper, paint sprayer, welding tip, electro-magnet or any of a wide variety of devices. Common manipulators use only revolute (turning) or prismatic (sliding) joints to connect links. More complicated forms of joints can always be modeled as a combination of these simple elements. The function of a link is merely to fix the relationship between the joints at either end of the link. As long as the two joints are held at the same relative

position and orientation with respect to one another, the link serves only as a rigid connector. The actual physical shape of the link can be changed without altering the kinematic relationship between the two interconnected joints.

Link Parameters: A kinematic description of the manipulator is generated by SpatialAnalyzer using the standard Denavit-Hartenberg notation. This remarkably simple representation uses only two values to describe each link, and two additional values to describe the joint connection between adjacent links. Links are described in terms of their length (a) and the twist they maintain between adjacent joints (α), as explained below.

Link Length (a): the distance between the two joint axes of a link. This distance must be measured along a line that is normal (i.e. perpendicular) to both joint axes. This line is referred to as the common normal between the joint axes. Links have zero length when the joint axes intersect. This is a common arrangement in industrial manipulators.

Twist Angle (α): the angle between the first and second joint axes on a link. This angle is measured in a right-hand sense when viewed directly along the common normal.

The joint connection between adjacent links is described in terms of the offset (d) and the joint angle (θ), as explained below.

Offset (d): The distance between the common normal of one link and the common normal of the next, measured along the axis of the joint connecting the two links.

Joint Angle (θ): The angle between the common normal of one link and the common normal of the next, measured about the axis of the joint connecting the two links.

The four parameters a, α, d , and θ , when taken for each link and joint in the chain, provide a complete kinematic description of the manipulator. When taken together with a description of the physical shape of each link and limits on joint motion, an accurate simulation of robot motion and operating workspace can be developed.

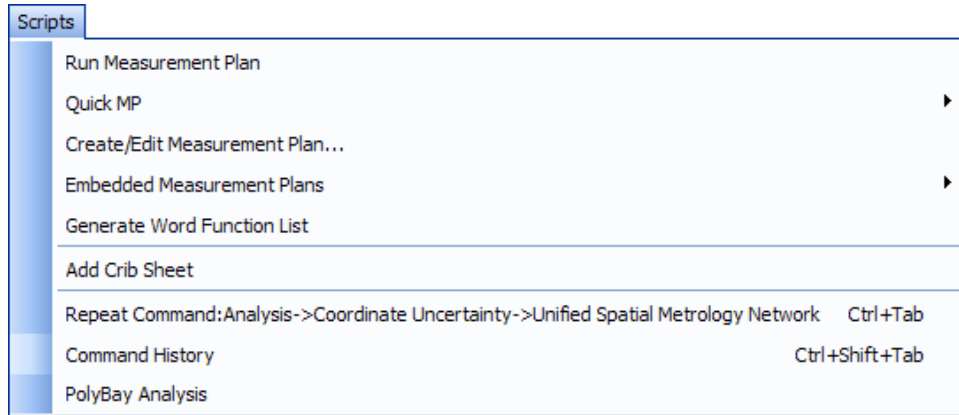
In addition, there is usually a Joint Offset value computed during the calibration. This value represents the difference between the “kinematic zero” of a joint, and the mechanical zero based on how the encoder was set in the joint mechanism.

For a more detailed description of Denavit-Hartenberg notation see:

Craig, J. J., 1989, *Introduction to Robotics: Mechanisms and Control*, Addison Wesley, New York.

Scripts

The Scripts menu provides access to create, edit, and run the programming elements of SpatialAnalyzer. The top two menu selections allow you run and create Measurement Plans. The last menu items allow you Repeat Commands and to view a command history.



Scripts Menu

Run Measurement Plan

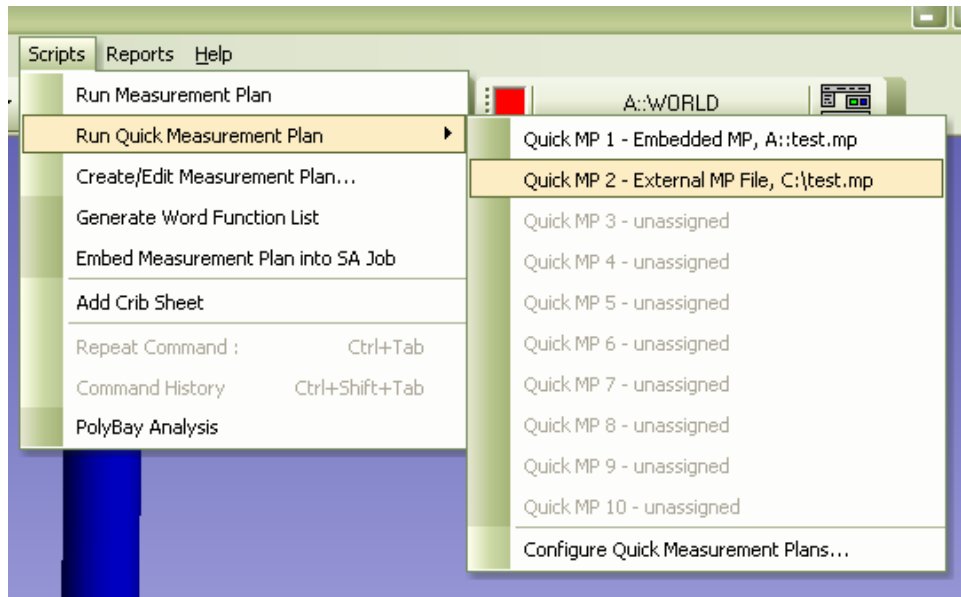
SpatialAnalyzer's Measurement Plans (MP's), operators can script repetitive or automated metrology tasks. These plans can vary from simple utilities constructed in just a few minutes to comprehensive inspection plans that provide go/nogo type automation and drive other machinery such as robots or CNC machines. This menu selection allows you to run a previously created MP. MP functionality is covered in detail in the Measurement Plan Programming section of this manual.

Quick MP...

This feature provides 10 Quick MP slots per job (.xit) file. Each Quick MP slot can be assigned to an embedded MP or to an external MP file. Once assigned, the associated MP can be run by clicking on its entry in the menu. Using the recently added User Interface Customization feature, the Quick MPs can also be assigned to toolbar buttons or toolbar drop-down menus for easier access.

Accessing Quick MPs

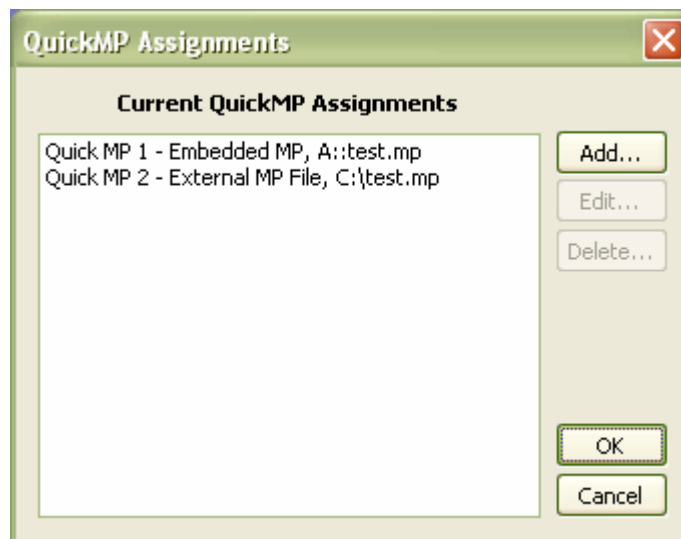
The Quick MP feature is accessed using the "Scripts>>Run Quick Measurement Plan" menu shown below.



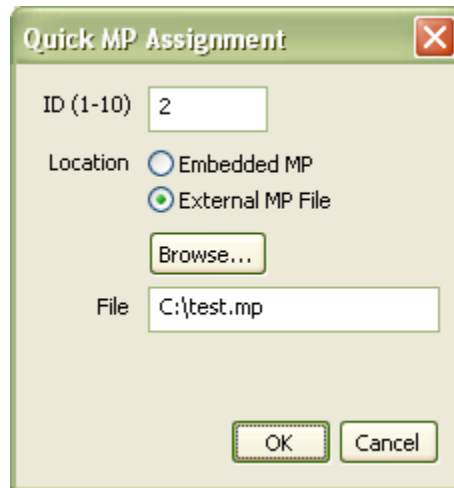
The above menu shows the 10 Quick MP slots for the currently open job file. Assigned slots indicate whether the MP is embedded or external and also provide the specific MP information. Slots which have not been assigned are indicated as such.

Configuring Quick MP Assignments

To configure Quick MP Assignments, select “Scripts>>Run Quick Measurement Plan>>Configure Quick Measurement Plans...”. This will open the “QuickMP Assignments” window shown below.



The above window allows adding new Quick MP assignments, editing existing Quick MP assignments, and removing Quick MP assignments which are no longer needed. When adding or editing Quick MP assignments, the “Quick MP Assignment” window is shown.



Using the above window, the Quick MP assignment can be configured. A numeric ID (1-10) can be set for the Quick MP, and the user can decide whether the Quick MP assignment refers to an embedded MP or to an external MP file. Finally, the user can provide the collection and name pair (for embedded MPs) or the filename (for external MP files).

Create/Edit Measurement Plan...

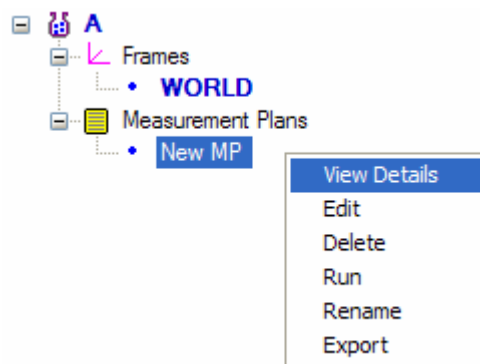
This menu selection allows you to create and edit an MP. The MP programming environment is loaded when you select this menu item.

Embedded Measurement Plan...

This menu selection allows you to embed measurement plans into the SA job treeview. Once embedded they can be executed and run from the Quick MP or by double clicking them on the treeview.

There are options when embedded a Measurement Plan. A new MP can be embedded to the SA job database tree. Another option is to embed an existing MP. This choice opens a File Selection dialog to select the MP to embed.

Once an MP is embedded right-click on it to View Details, Edit, Delete, Run, Rename, and Export the measurement plan.

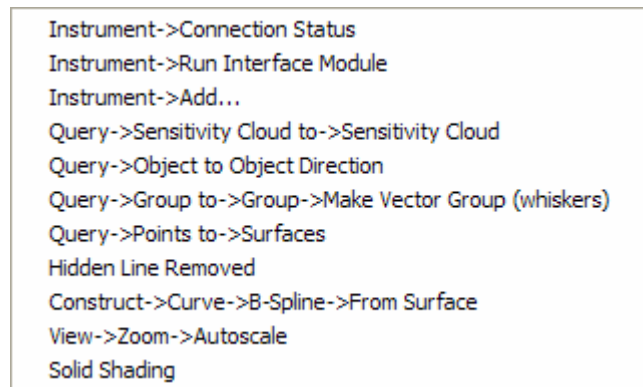


Repeat Command ... Ctrl+Tab

This menu selection allows you to repeat the last command or function you used. It a nice time saver when you need to repetitively use the same command. Please note that this function's hot keys are Ctrl + Tab together. Use them together to repeat the last command.

Command History ... Ctrl+Shift+Tab

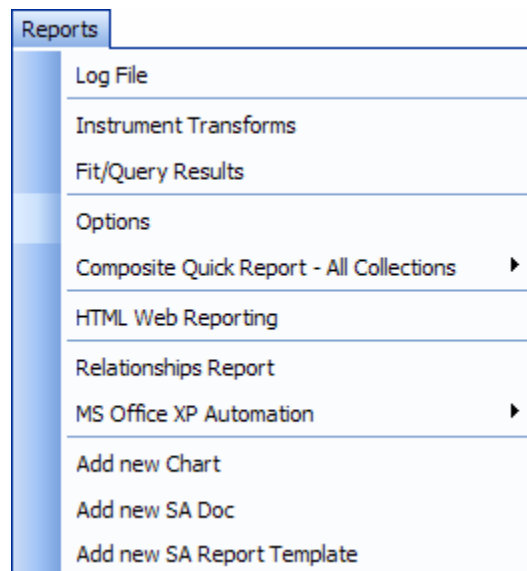
Accessing any of the last ten commands is available though the Command History functions. A list of the command is shown in a list for you to select one. An example of a command history is shown below in the figure.



Instrument->Connection Status
Instrument->Run Interface Module
Instrument->Add...
Query->Sensitivity Cloud to->Sensitivity Cloud
Query->Object to Object Direction
Query->Group to->Group->Make Vector Group (whiskers)
Query->Points to->Surfaces
Hidden Line Removed
Construct->Curve->B-Spline->From Surface
View->Zoom->Autoscale
Solid Shading

Command History example

Reports

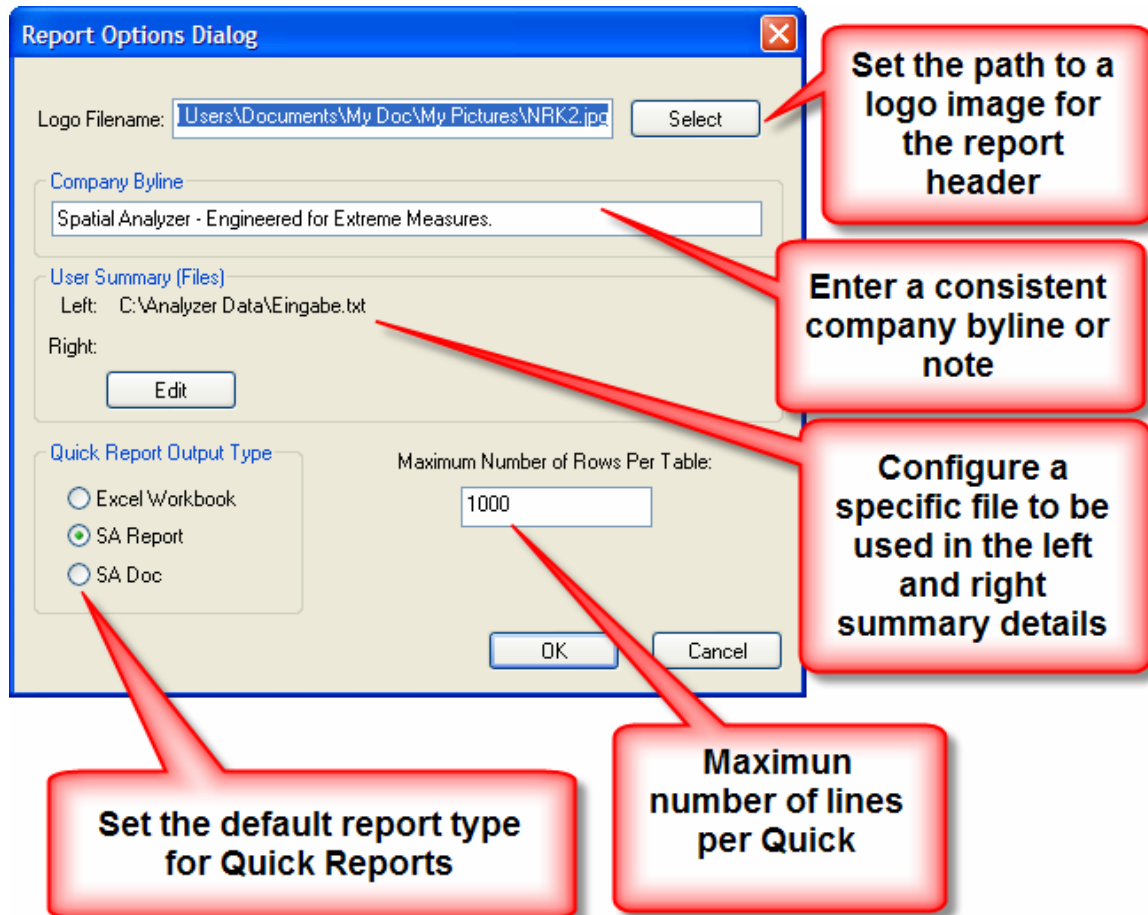


Reports Menu

Reports are covered in detail in the Reports section of this manual.

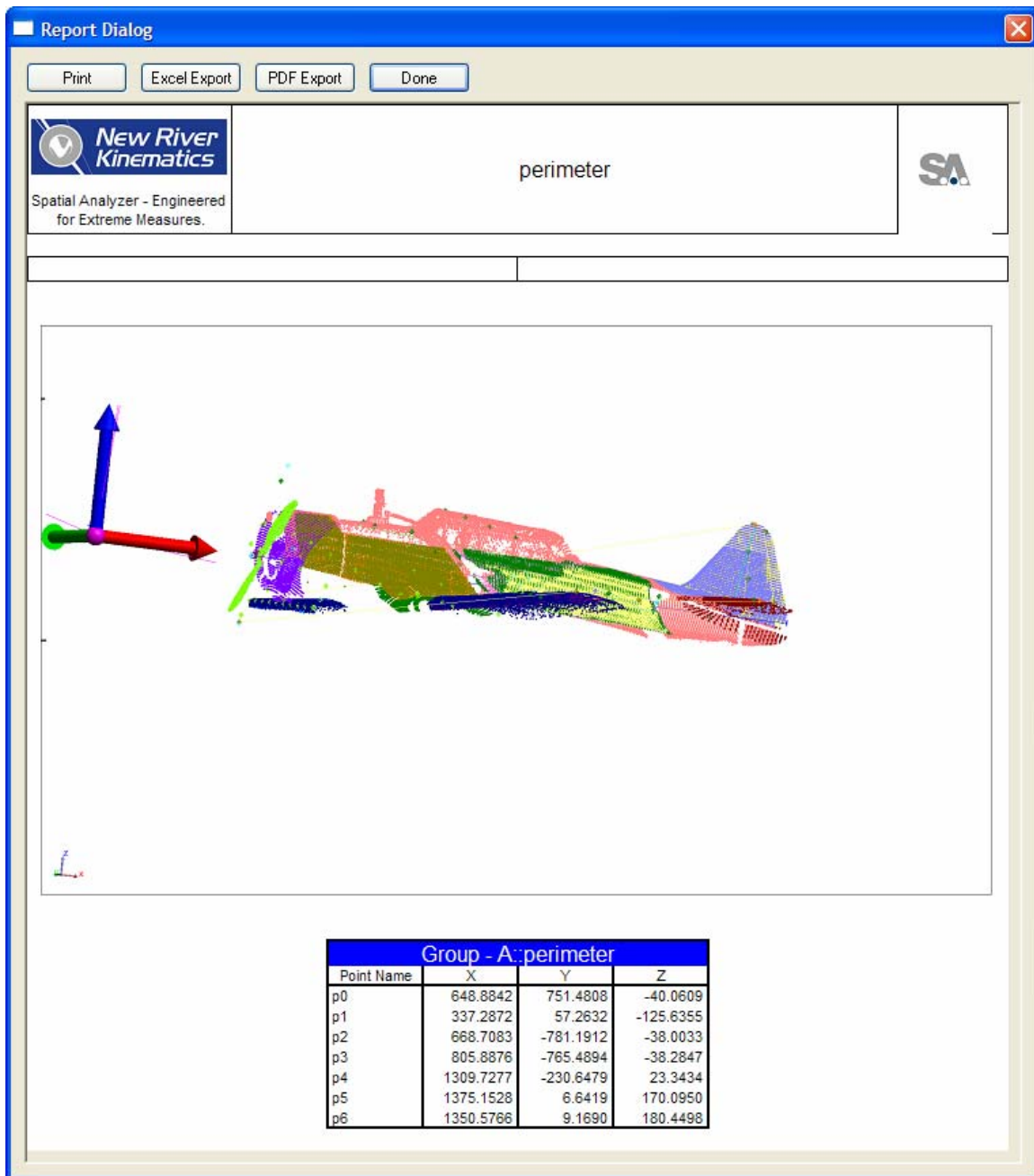
Configure Default SA Report Options

Configure the default SA report with the Report >> Option dialog. There are a number of configurable attributes, including the output type. Set this to automatically create an Excel Worksheet, SA Doc or SA Report.



The maximum number of rows per report default is 1000. Change this number to match your typical output requirements.

An example SA Report is shown below.



Example SA Report

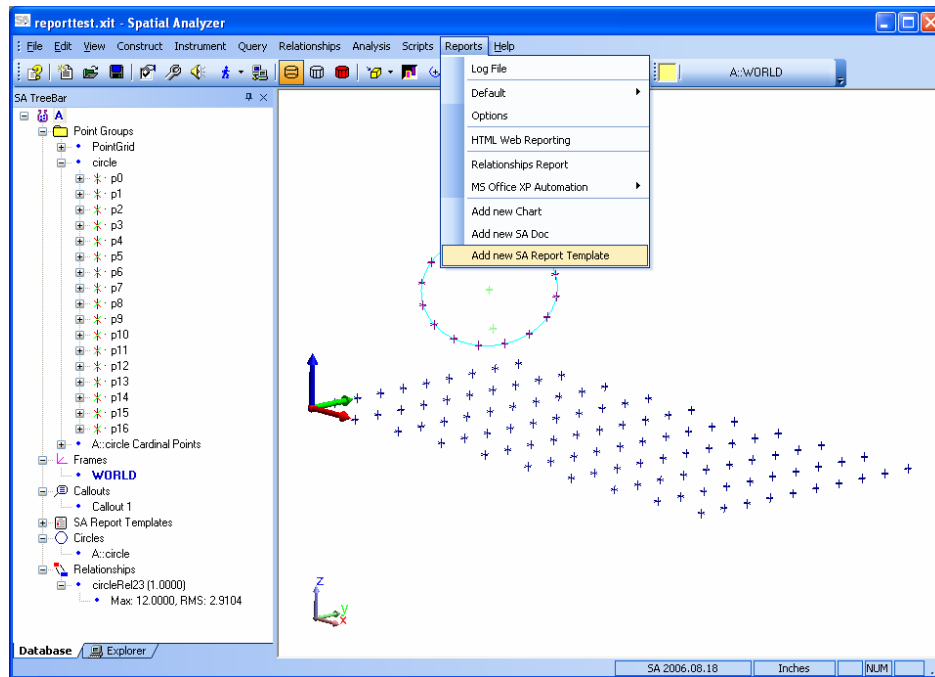
Report Templates

In previous SA versions, the report template allowed you to specify a title, user summary files, a graphical view, and a job item and these were sent to the report in a fixed order.

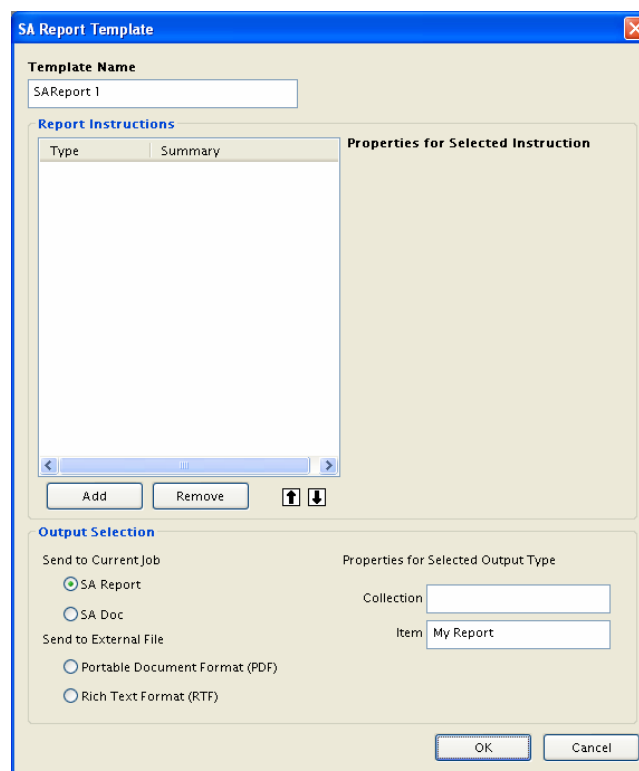
As of SA version 2006.08.18, the report template now allows you to specify multiple titles, user summary files, graphical views, and job items and control the order in which they get sent to the report.

Report Templates Walkthrough

Select Reports >> Add new SA Report Template from the menu.



The SA Report Template window is shown with a default name, empty instruction list, and default output selection

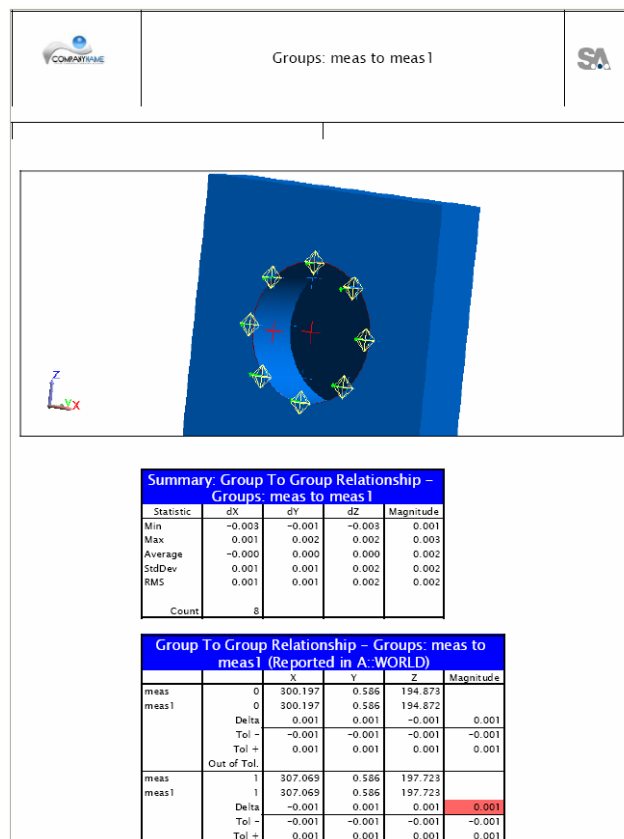


You'll need to add one or more Report Instructions to the template. Report Instructions are the graphical views, job items, title blocks, and user summary files which you want to be included in the generated report

Add as many Report Instructions as needed for the application. The order is configurable with up and down arrows. When the report is generated, the Report Instructions will be executed in the order shown in the list. Select the desired output type and destination info and then click OK. Locate the new report template in the Job database tree, right-click on the template, and select "Generate."

Quick Reports

SA's Quick Report offers a quick way to create reports that include a graphical element as well as tables. For example, in the following survey, a comparison between two point groups is reported.

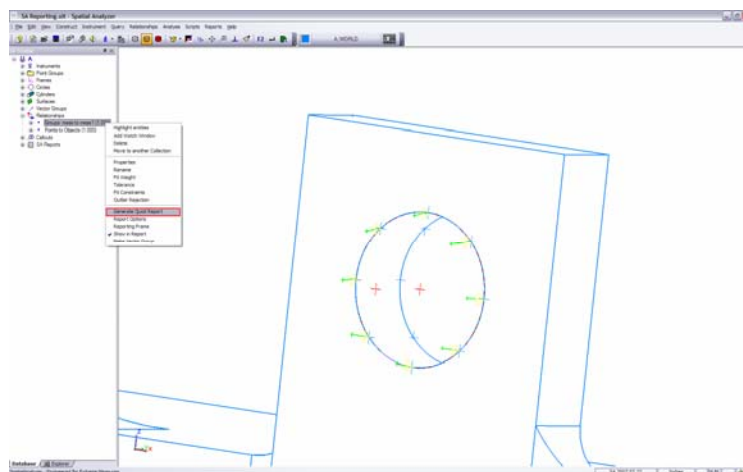


The red highlighting in the magnitude column of the second point indicates a tolerance violation. The amount in excess of tolerance is also displayed. Tolerance, ideally specified in the CAD model, may be set manually within SA. It may be asymmetric and independent for each component of a point. Tolerance may also be set generally for an entire relationship. When specified redundantly, the more specific setting overrides the less specific setting.

Here is the user interface that appears at both relationship and point levels. Additionally, point properties can be set once for any selected set of points (even across groups and collections).



Regardless of how tolerance information was specified, or even if tolerance unavailable, Quick Reports are created by simply right-clicking and Generate Quick Report from the pop-up menu as shown below.

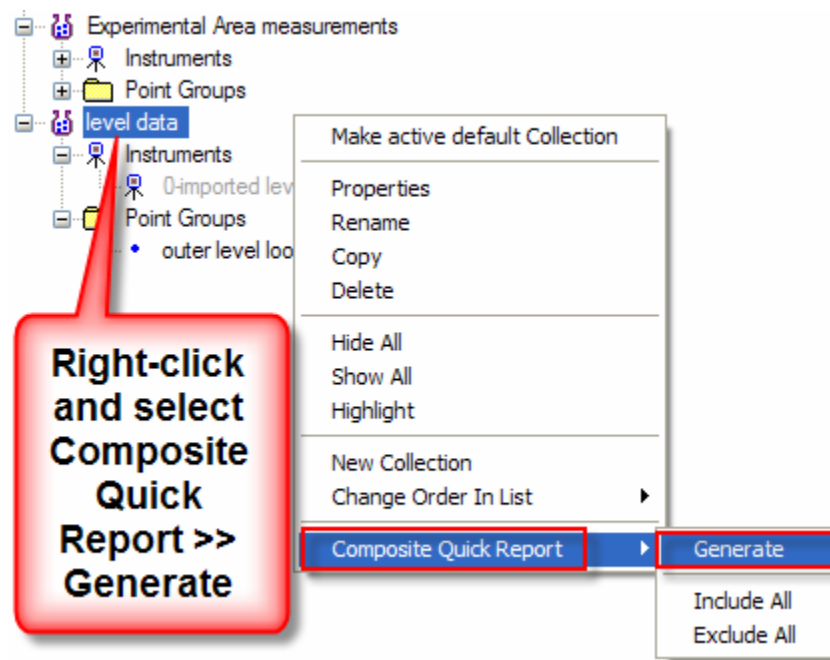


Notice the subsequent menu item, Report Options. Selecting it allows considerable customization. In Components, reporting of either or both of the points as well as the calculated difference can be suppressed. Under Format, a choice of horizontal or vertical orientation of X, Y and Z Cartesian components is provided, as well as the option to concisely report each point on just a single line. At the bottom, the tolerance information can be excluded from the report. And finally, the report can be sorted alphabetically by point name (by default data is reported in the order it was measured).

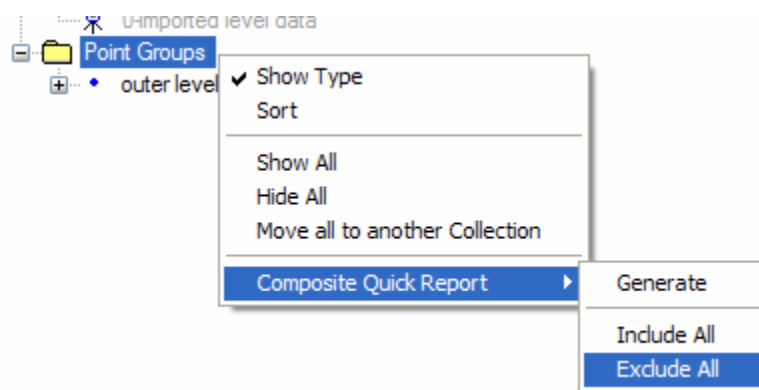


Composite Quick Reports

Making reports that include all the objects in a collection and for all the objects across collections is done by making Composite Quick Reports. The figure below shows the Composite Quick Report selection for a collection. The same functionality is available for individual object types.



Options to Include All or to Exclude All allow control the information that is included in Composite Quick Reports. All objects are included by default. To exclude all the object of a particular type within a collection, select the object type on treeview then right-click it to show the menu. Select the Composite Quick Report >> Exclude All option to eliminated them from Composite Quick Reports. The figure below shows an example.

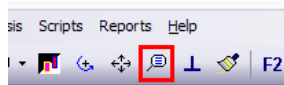


Including these objects back into future Composite Quick Reports is control using the Include All option. This mechanism works each category -- object grouping, instruments, and whole collections.

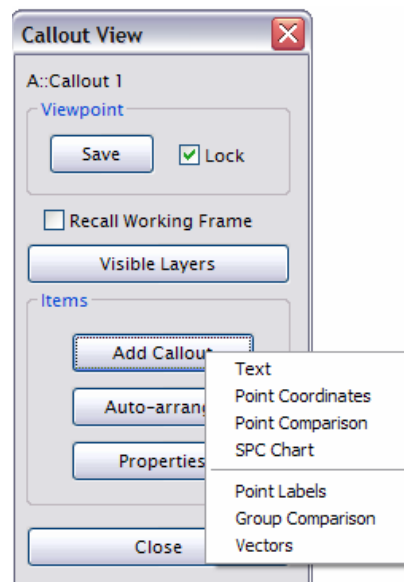
Extending Quick Reporting

As flexible as it is, the basic premise of Quick Reports is one picture followed by one table, where the table is selected from the Treebar and the picture is the current contents of the graphic window. What if the picture annotations in desired? What if more than one table should be reported?

Annotating the graphic window in SA is accomplished with the use of Callouts. Freeform text, point labels, coordinate values, point comparison and vector labels and values are some of the typical annotations Callouts support. Callouts are created by clicking the cartoon bubble on the Toolbar,



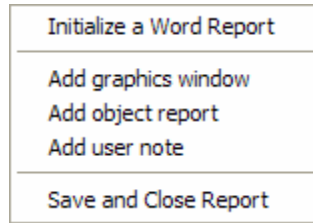
which opens the dialog below. Starting at the top, in the Viewpoint section, pressing Save causes the currently display graphics window to be displayed when the callout is displayed. Selecting Lock prevents the user from changing the viewpoint by panning, rotating or zooming. Recall Working Frame ensures that the viewer sees the same coordinate values as the callout creator (by changing working frame). The Visible layers button allows hiding and showing particular CAD layers (which can also be created in SA). Pressing the Add Callout button opens the pop-up menu facilitating various annotations. Text allow freeform labels, Point Coordinates and Point Labels support titling individual points while Point Comparison, Group Comparison and Vectors provide annotation of comparisons, such as between nominal and measured values. Auto-arrange provides a quick way to spread out the callouts for easier placement.



MS Office Reporting

SpatialAnalyzer provides an easy way for you to generate Microsoft Office document based reports with imbedded graphics, report tables, and user notes. This capability works if you have Office XP or a later version of Microsoft's Office Suite. The menu

shown in the figure below provides access to the individual steps for creating, adding to the report, and then to save and close the report. The Reports section of this manual provides a good description on how the process works.



MS Office Reporting Menu

INSTRUMENT INTERFACES

One of the most powerful aspects of **SpatialAnalyzer** is the modular means that interfaces to metrology hardware work between the core analysis platform and diverse instrument dependent communication requirements. These interfaces are able to work independently and together on a single computer as well as across networked computers.

Instrument interfaces communicate with **SpatialAnalyzer** by a robust TCP/IP network protocol. This affords several distinct advantages. First, these connections are extremely fast and reliable. Second, since this is a “network” protocol, the instrument interfaces may be run on either the same computer as **SpatialAnalyzer** or on another computer on the same network. Third, and perhaps most impressive, since this is the “Internet Protocol” (IP), the computers running the instrument interfaces could be in the same room, across the country, or around the world. Likewise, if activated, it is possible to remotely monitor a measurement job in progress using this same TCP/IP connection over the Internet or on a local network.

Modular instrument interfaces are the basis of the data acquisition aspect of **SpatialAnalyzer**. These provide a easy to use and friendly interface for the control of instruments during a measurement job.

Establishing the instrument interfaces as separate entities in the architecture affords several key advantages. First and foremost, this provides flexibility to adapt to tomorrow's instrumentation. The number of instrument interfaces available for **SpatialAnalyzer** is ever expanding, much like printer drivers and other peripheral devices for the Windows operating system. Second, this approach enables us to provide a common interface for each type or class of instrument. Thus a portable CMM manufactured by Company A would have the same top-level interface as a portable CMM manufactured by Company B. This allows users to easily select the instrumentation best suited for a particular task. It also allows users to easily transfer the expertise they have gained with one instrument to another. A direct benefit of this capability is to improve a user's flexibility when moving from task-to-task and it reduces the training time for a user to become productive on a different instrument. Third this allows for a more objective view when it comes time to replace or purchase new hardware since re-training is minimized by the common instrument interface in **SpatialAnalyzer**.

In all cases, the instrument interfaces perform four key functions.

They provide a common and simple user interface that can be used under **SpatialAnalyzer** control to guide a user through a particular measurement task.

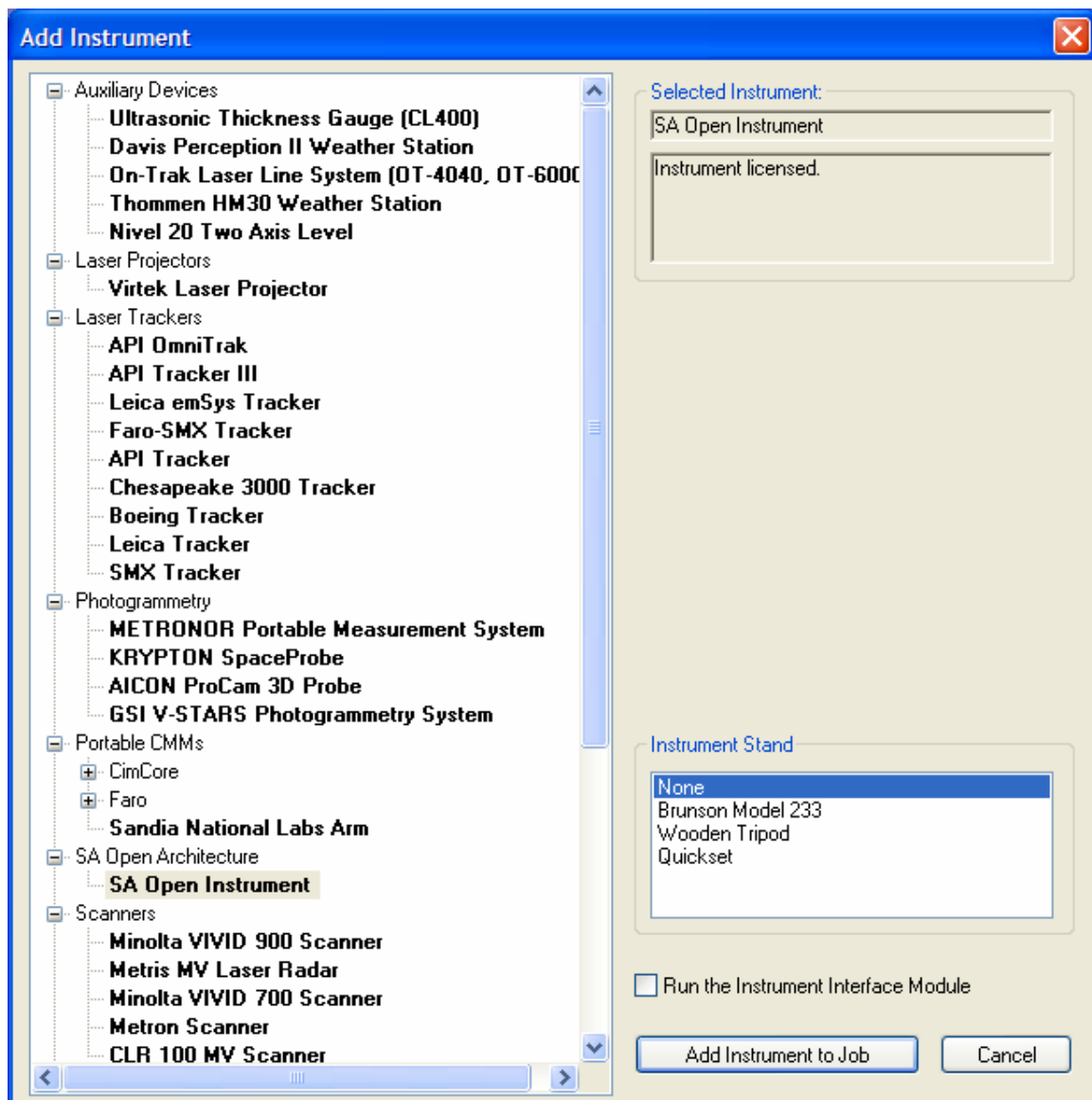
They communicate over a TCP/IP network to convey essential measurement data to **SpatialAnalyzer**.

They communicate via a native format to the instrument. For example: serial RS-232 connections for many theodolites, Network protocols for many laser trackers.

They provide an efficient means to collect the raw measurements, the conditions and techniques under which they were collected, and to perform operation checks on the instrument to establish that it is meeting its performance specifications. It also supports simulating measurements and measurement tasks before the job is performed. By simulating a measurement process before the job is performed, accuracy and time estimates can be made to help you make the best choices relative to optimizing the placement of the instrument(s) relative to the object.

Adding an Instrument to an SA Job

To add an instrument to an SA job the operator goes to the Instrument menu item in **SpatialAnalyzer**. Select Add from the Instrument menu. A dialog of all the instruments interfaces is shown below.



SpatialAnalyzer Add Instrument Dialog


Select the instrument that you want to work with from the list on the left of the Add Instrument dialog. You can choose a stand from a list of stands on the right. If you are not using an instrument stand, then choose the “None” option in the Stand list. If you have previously connected to the same type of instrument from the computer running the **SpatialAnalyzer** application, then you can choose to automatically run the instrument interface by checking the appropriate box on the dialog.

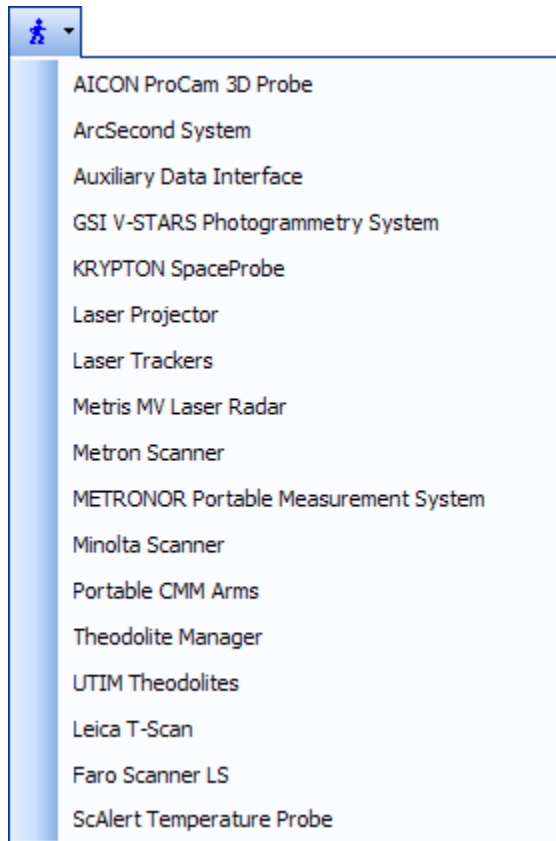
When you have completed your instrument and stand selections, hit the “Add Instrument to Job” button. This process adds a model of the instrument into the job. The default location for a newly added instrument model is (typically) 50 inches along the World frames X-axis. When adding a second instrument to a job it will initially appear 100 inches from the World frame. Each additional instrument that you add will appear at 50 inch increments along the X-axis.

The real-world location of the instrument can be set or determined from measurement using a wide variety of processes.

Starting an Instrument Interface

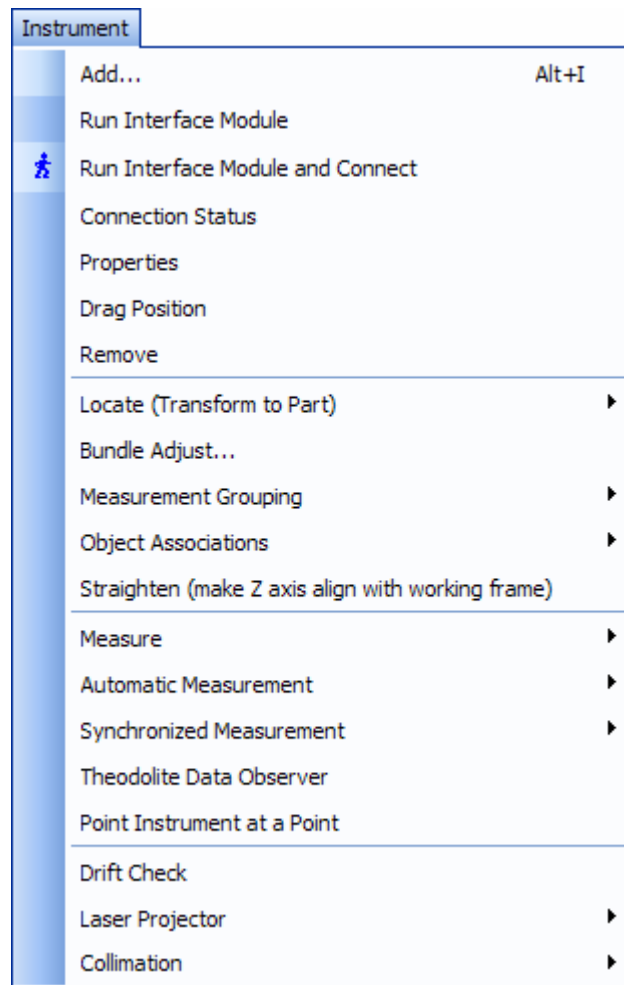
Once the instrument model has been added to the job, the instrument interface has to be started. An instrument interface is a separate program that manages the communication between **SpatialAnalyzer** and the instrument.

Instrument interfaces can be started in several different ways. The first option is available on the Add Instrument dialog. Checking the ‘Automatically run the instrument interface’ check box on the dialog, will automatically start the interface program and request that it establish a connection between the computer running **SpatialAnalyzer** and the specific instrument you are adding. It will attempt to establish the interface using the parameters from the previous connection. A second option for starting instrument interfaces is provided on the toolbar. By clicking  button on the toolbar, a list shows the instruments that are available to run. Selecting the appropriate instrument model starts the instrument’s logon procedure.



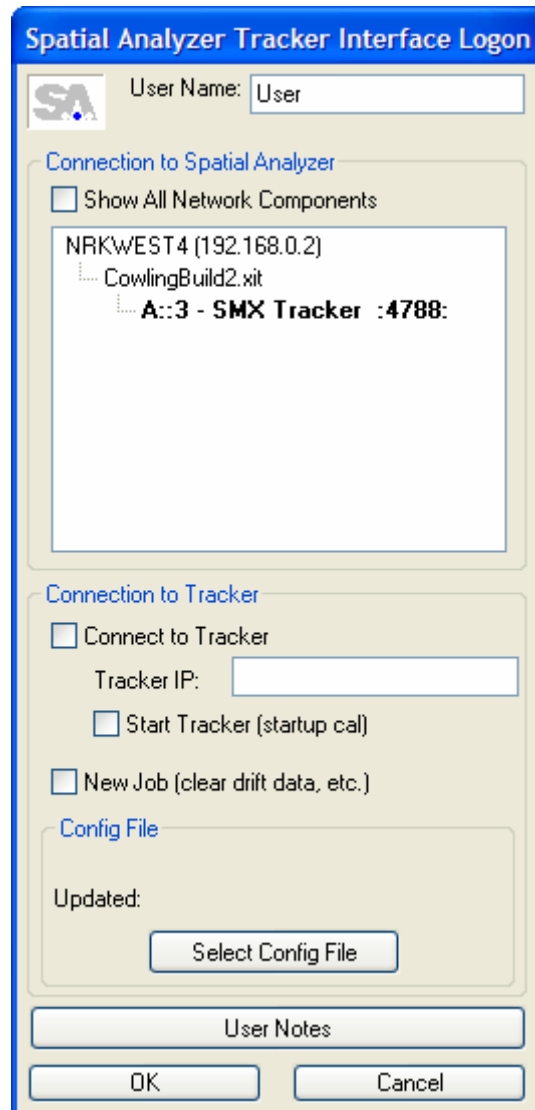
Run Instrument Interface Toolbar Dropdown

A third mechanism to start an instrument interface can be found on the standard Windows Start bar. First navigate to the SpatialAnalyzer folder under the Programs branch of the start menu. By selecting the Instrument Interfaces sub-folder you will see the menu shown below. Select the appropriate instrument from the menu. The particular method will start the instrument connection process with the basic logon procedure.



Instrument Interface Menu from Windows Start Bar

Each instrument interfaces typically requires some basic information to function properly at start-up. A typical instrument interface logon screen is shown below. Note that the operator is prompted to enter their user name, the instrument ID corresponding to the index within the **SpatialAnalyzer** model (these indices will start with 0 and proceed upward as more and more instruments are added.)



Spatial Analyzer Tracker Interface Logon

SA User Name:

Connection to Spatial Analyzer

☐ Show All Network Components

NRKWEST4 (192.168.0.2)
 ... CowlingBuild2.xit
 ... **A::3 - SMX Tracker :4788:**

Connection to Tracker

☐ Connect to Tracker
 Tracker IP:

☐ Start Tracker (startup cal)

☐ New Job (clear drift data, etc.)

Config File

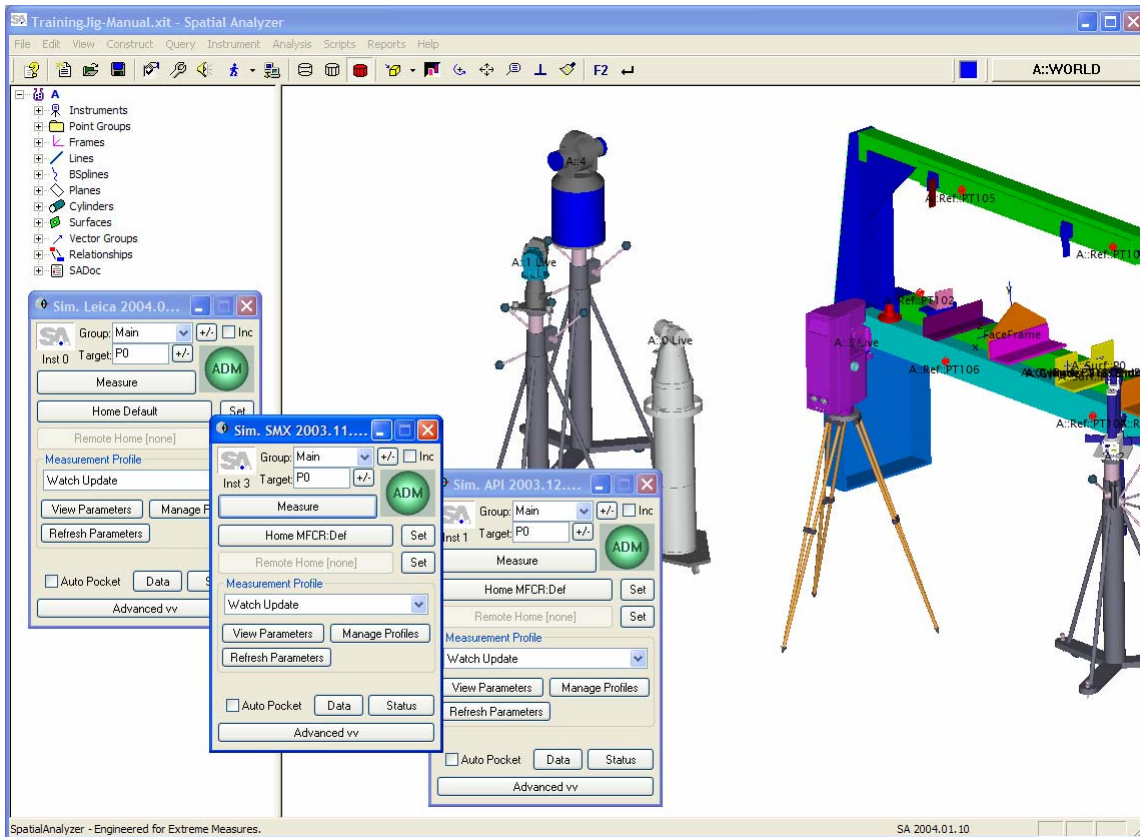
Updated:

A typical instrument interface logon screen

Typically the instrument interface and **SpatialAnalyzer** will be run on the same computer. However, the TCP/IP socket connection between the instrument interface and **SpatialAnalyzer** allows the operator to run the interface on one computer and **SpatialAnalyzer** on a separate computer. While not typically used when connecting to just one instrument; when two, three, or more instruments are needed on a job simultaneously or in remote areas, the connection technology in **SpatialAnalyzer** is able to run them all dependably across your companies network. The connection technology is the same as that used to connect computers across the Internet. **SpatialAnalyzer** is architected to allow you to take advantage of your company's network.

When the operator hits OK on the instrument interface logon dialog, the selected interface will start and perform routine checks of the network and hardware. In all subsequent starts of the instrument interface, this information will be retained and the

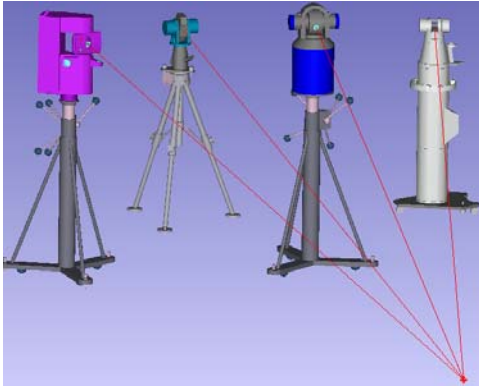
fields will be populated with this default information. The figure below shows three instrument interfaces running on top of **SpatialAnalyzer**. Menu options from within each instrument interface allow the user to control the tracker's functions, change units, and other instrument specific functions.



Several Instrument Interfaces Running 'On Top' of SpatialAnalyzer

The subsequent chapters provide the detailed information for each instrument interface. They are organized by instrument type.

LASER TRACKER INTERFACES: LEICA, FARO, SMX, API



The SpatialAnalyzer Laser Tracker Interface application is used to interface to one of three families of laser trackers. The interface communicates to SpatialAnalyzer using New River Kinematics' TCP/IP socket protocol and communicates to the given tracker using the manufacturer's communication protocol. Leica_Tracker.exe, the Leica tracker interface application, uses either NetBEUI or TCP/IP depending on the controller, SMX_Tracker.exe uses TCP/IP, and API_Tracker.exe uses serial communication. The trackers are quite different at the instrument level, but the interface applications look and feel identical to the user.

To run any of the tracker interfaces, you must first install the tracker specific configuration, or parameter file. This file is available from the given tracker manufacturer, and in fact comes with the tracker in all cases.

Laser Tracker Overview

A laser tracking interferometer system provides real-time three-dimensional measurements in large volumes. It is portable and easily moved to the measurement site. Laser trackers are spherical measurement systems that measure a three-dimensional location of a retro-reflector. A laser tracking interferometer system provides real-time three-dimensional measurements in large volumes. It is portable, allowing the user to move the system to the measurement site.

Trackers record the position of the retro-reflector using a single beam interferometer to find its range and optical encoders to find the horizontal and vertical angular positions. The major components of the system are the sensor unit (i.e., measurement head), laser tracker controller, application processor, environmental monitor, and the retro-reflector.

The sensor unit reads the raw angles and distances of the retro-reflector. The trackers used in this application are equipped with an Absolute Distance Meter (ADM). The ADM typically uses a modulated infra-red laser source to determine the distance to the retro-reflector. Each of the selected trackers also has a Helium-Neon (HeNe) laser interferometer for relative distance measurement. The laser tracker heads use either tilting a mirror or prism to steer the point beam (typically the HeNe) or a position sensing diode to track a retro-reflector.

IFM and ADM Measurement

Most trackers are also capable of ADM, or Absolute Distance Meter measurement. ADM measurements differ from Interferometer Measurements (IFM) in the method used to determine the range to the target. In either case, the trackers use precision encoders to measure the azimuth and elevation angles. The IFM measures range by literally counting wavelengths of visible light (the red beam) in the return laser beam. As a consequence, the IFM requires that its initial range is set for it and the laser beam remains locked on the retro-reflector. The IFM distance from tracker to retro must be established by locking the beam on the retro and setting the distance to a known value, then all subsequent distance information is relative to that original setting. However, if the line of site between the tracker head and the retro-reflector is interrupted, the distance must be reset.

The ADM system measures the range directly, hence the name Absolute Distance Meter. With ADM measurement, the tracker beam can be locked on the retro in any position, and physical properties of the light are varied in a predictable pattern and the return beam of light is contrasted against it to compute the distance to the target. The laser source for this measurement is typically in the infrared wavelengths (i.e., not visible). It tends to take longer to acquire the absolute distance (the IFM is constantly updating distance by counting light fringes).

IFM is generally faster (1,000 targets/sec) and more accurate when compared to ADM (4 – 8 seconds per target) acquired targets. However the ADM does not require a constant lock on the retro. ADM measurements are not as precise as the IFM, but in many cases, the precision is more than adequate for the application. Imagine a retro which is placed on the ceiling of a large arena for example – it would be worth a few fractions of an inch in accuracy, and a second or two of acquisition time not to be forced to climb scaffolding while being sure to point a retro back at a tracker.

System Architecture

The application processor is comprised of a personal computer and has routines for performing system alignments (i.e., the numerical compensation that corrects optical-mechanical alignment) in addition to providing the human interface.

The laser tracker controller contains its own central processing unit which applies corrections to the angle and distance values measured by the angle encoders and laser source. Environmental factors such as changes in atmospheric pressure, temperature and humidity effect the wavelength of the IFM and ADM are corrected by the controller. Accurate measurements of the environment are made by automatic input with an environmental monitor.

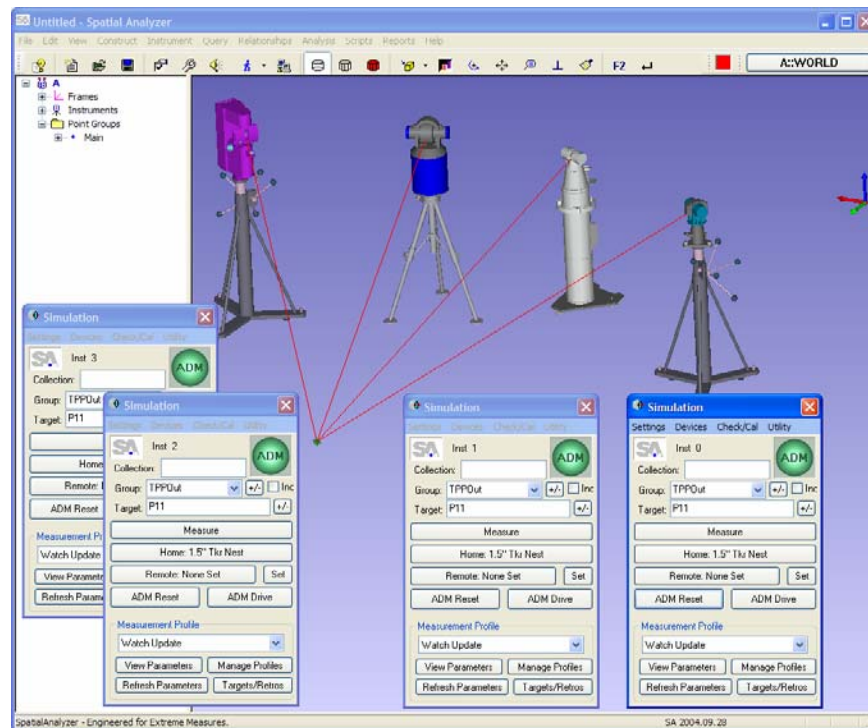
Retro-Reflector Targets

Laser trackers depend on a retro-reflector for measurement. There are two primary types of retro-reflectors. The Open-Air retro-reflector has an acceptance angle of $\pm 40^\circ$ for tracking. Open-air retro-reflectors are nice to use because they have no distance or pointing offset between the IFM and ADM measurements. A second retro-reflector type is the glass prism mounted with the apex in the center of a spherical holder. Retro-reflectors of this type have a measurable offset between the two distance measurement systems (i.e., IFM vs. ADM). When using a glass mounted retro-reflector (also known as

TBR) care must be taken to offset the difference in distance measurements and accurately point the retro-reflector back to the tracker. The offset is called 'ADM Offset.' Solve for this parameter with the ADM/IFM Field Check function in the interface.

Common Tracker Interface

The figure below shows four different trackers and their respective interfaces measuring a common point. The interface to each tracker is common enables operator to move between trackers with minimal or no training.



Starting the Interface

To run the tracker interface, as with any instrument interface in SpatialAnalyzer, the first step is to run SA and add the instrument model. This is more than just a model; it supplies the connection for the instrument, and serves as the real-time representation of the instrument in SA.

The establishing the connection between the interface and SA is typically a two part process. The first establishes a number of elements including which computer on the network, session SA and instrument within that SA job to connect too. All the computers on a network can be running SA and each computer might run multiple copies at the same time.

From the main menu in SA, select Instrument>>Run Interface Module.

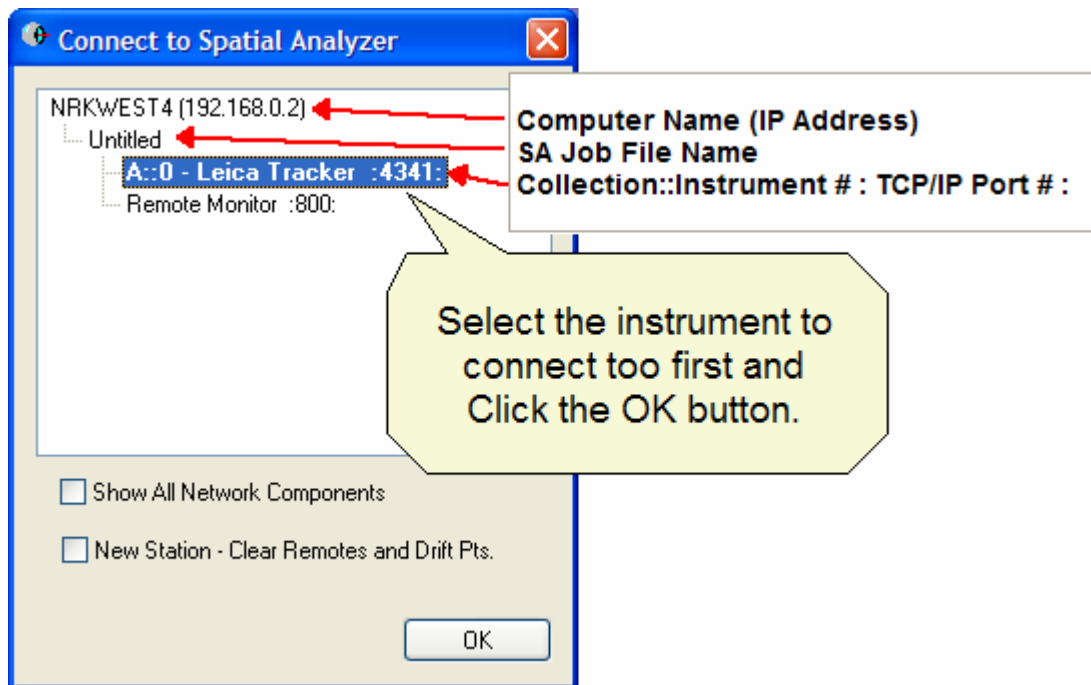
Select from the list of instruments.

To keep track of which computer and session of SA the Connection to SA has a treeview showing all the potential computers, jobs, and instruments that the particular interface can connect too. Find the instrument object in the SA job with the following attributes:

Computer Name (on the network) to connect this interface too

SA Job File Name on that computer which SA session to connect too

Collection and Instrument Number within the SA job file



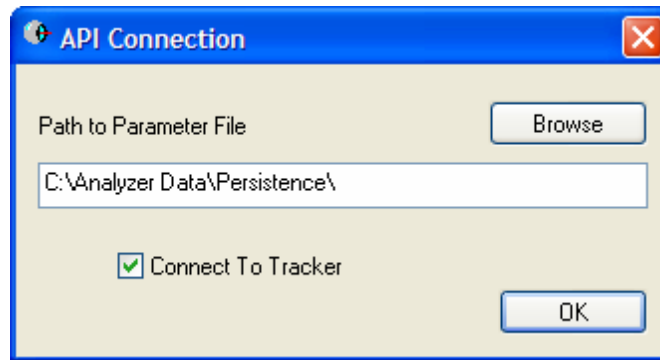
Select the Instrument in the list then click the OK button.

The second step in the connection process configures instrument specific connection details. Depending on which instrument type is connected to SA the logon dialogs will vary.

Configure the settings on Instrument Connection dialog to make a successful connection to the instrument, and Press OK. The interface completes the connection between SA and the instrument.

Initialize API and Simulation Trackers

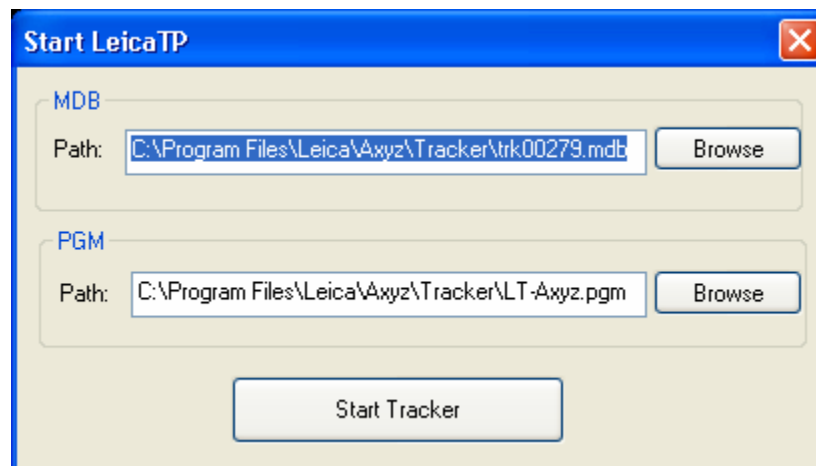
When using a Tracker or Simulation tracker select the Initialize button to start the tracker communication module.



Start Leica Trackers

If the Leica or Leica EmScon tracker is to be used and the instrument has not been started since it was powered up then you need to select the Start Leica button. If the instrument has already been started select the Initialize button to open communications with the instrument.

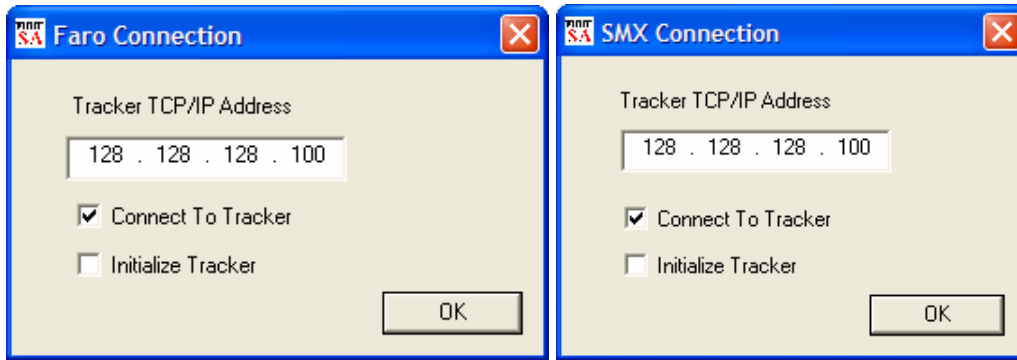
Starting the TP-Link based Leica trackers involves sending the trackers calibration parameters down to the controller. The Start Tracker dialog will ask you to Browse to the tracker's mdb and pgm files. These are typically in the [Programs Files\Leica\Axyz\Tracker] folder on the computer hard drive. Browse to these files and then select Start Tracker to start the process. The Start Tracker dialog is shown below.



Starting the Leica EmScon tracker does not require navigating to the mdb or pgm files since they are stored on the controller. To start EmScon instruments select the Start Leica button to initiate the process. Starting the instruments also initializes the interface so selecting the initialize button is not required after a start command.

Start Faro or SMX Trackers

When Faro or SMX trackers are used for the join select the appropriate model from the Tracker Type drop down box. Initializing the tracker interface is the first step in the process. The Faro and SMX tracker connection dialog box are similar in puts and options. The figures below show examples of both Connection dialogs for the Faro and SMX trackers.

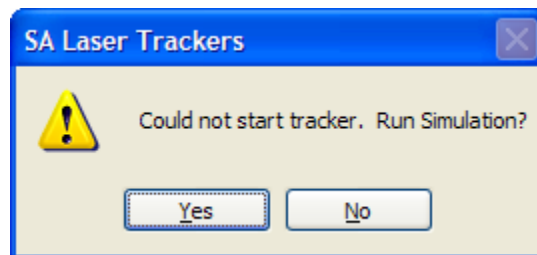


To initialize Faro and SMX trackers select the Initialize Tracker check box on the Connection dialog. After connecting to the tracker controller the interface will initiate a tracker startup procedure on the controller. The initialization process takes approximate 1-2 minutes to complete.

After starting up a tracker performing operational checks are recommended for the tracker (irrespective of the tracker type.)

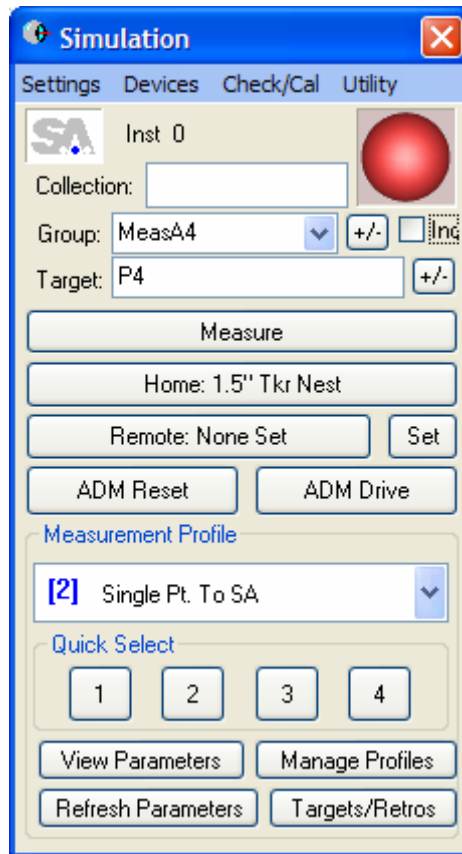
Tracker Simulator

If the connection fails to make a successful connection to an instrument, a dialog asking about running a simulation is presented. Select Yes if an instrument is not available.




If an instrument connected and ready the interface details need to be configured properly before continuing.

The Tracker Simulation Interface is shown below. For the purposes of this manual the simulator is used to illustrate functionality.

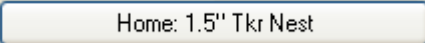


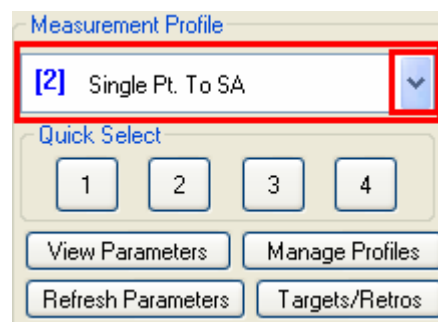
Close Instrument Interface


Close the instrument interface by pressing the  in the upper right corner. Closing the interface saves the settings file. When the tracker interface is started again all of the settings are loaded as it was when last closed.

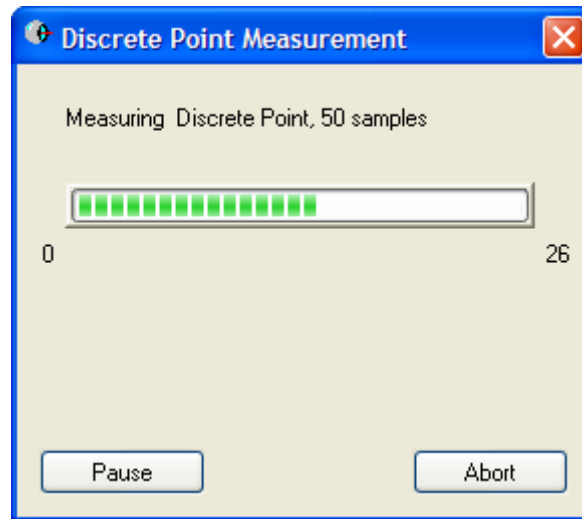
Taking Measurement Data

After the tracker interface is connected to SA and to the tracker and locked on a reflector, it is ready to take data. The first step is to Home the tracker, to do this, press

 button. Set the measurement profile to Single Pt. To SA.



The  button initiates the measurement process. The interface will begin a measurement according to the current measure mode and parameters for that mode. The measure progress dialog will appear showing the action of the measurement process:



In this case, the measure mode is set to Single Pt. To SA, and the corresponding measure parameters, Samples to Average and Delay Before Measurement, are set to 50 and 0 seconds, respectively. When the measurement is complete, the single point resulting from the average of 50 samples will be sent to SA, the measurement progress dialog will close, and the interface will be ready for the next action. So in general, any measurement will involve three steps – setting the measure profile, setting the corresponding measure profile parameters, and pressing the Measure button to start the measurement.

Auto-Measure commanded from SA

To initiate the auto-measure mode from SA use the Instrument >> Auto-Measure menu and select from a number of different functions. The SA interface sends commands to the tracker interface to point the tracker to a target from a list points. After pointing at the nominal location the interface then commands the instrument to search and lock on to it. After locking on the target the instrument measures it and the measured values are saved with the target. Once the target is measured the interface moves on to the next target in the list and begins the same process. The SA auto-measure interface continues to measure the points sequentially until either commanded to stop by the user or asked to stop via a command from SA.

A dialog to configure the acquisition parameters for measurement requests from SA is accessed from the tracker's interface menu. Go to the Settings >> Tracker >> SA Interaction >> Measurement Requests >> Single Point Parameters menu to set the acquisition parameters.

Overview of the Tracker Interface

The tracker interface divides the instrument controls into functional groups. These groups are:

Instrument Index and Target Naming

Measurement, Home, and ADM Controls

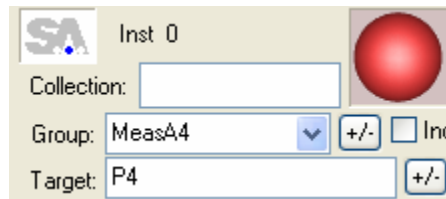
Measurement Profiles and Targets/Retro Interface

Tracker Interface Configuration Menus

Each group and its respective functions are described below.

Instrument Index and Target Naming

When an instrument is added to an SA job it is assigned an index. The index is used to associate measurements to instruments. Multiple instruments (of the same type) are added to a job to enable multiple station surveys. The instrument index number is incremented for each one added within an SA collection. The instrument index number that the interface is connected too is labeled just to the right of the SA logo on the interface dialog. In this example the instrument's index is zero [Inst 0].



The screenshot shows a portion of the Tracker Interface dialog box. At the top left is the 'SA' logo. To its right, the text 'Inst 0' is displayed. Below the logo is a 'Collection:' label followed by an empty text input field. To the right of the 'Collection' field is a red sphere icon. Below the 'Collection' field is a 'Group:' label followed by a dropdown menu showing 'MeasA4' and a small blue downward arrow. To the right of the dropdown are two buttons: '+/-' and 'Inc'. Below the 'Group' field is a 'Target:' label followed by a text input field containing 'P4'. To the right of the 'Target' field is a '+/-' button.

Controlling target names is an important part of the tracker interface. The interface has inputs for collection, group and target name. Understanding the behavior of the naming mechanisms helps make the process of completing measurement tasks easier and reporting results becomes a simple process to manage.

Collection Name

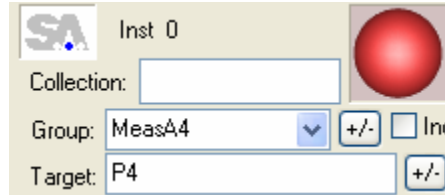
Measurements of targets or geometries made by the tracker interface are sent to SA. The first input on the tracker interface is Collection name. If a collection name is not specified on the tracker interface, measured targets are automatically added to the current collection in SA.


Group Name

Setting the Group name for measurements is controlled with Group field on the instrument interface. For a new Group name enter it into the field. To select an existing Group name from SA use the drop down list button on the right side of the field. A list showing all the group names in the active collection makes it easy to keep group names consistent.

Target Naming

Target naming is managed with the Target field. Enter the name of the target into this field. The combination of Collection::Group::Target work together to provide a easy way to manage data in measurement jobs. The mechanism is ignores the collection name for simple jobs and is also capable of handle complex setups. Getting comfortable with the naming process between the instrument interface and SA is an important step in using the all of capabilities of SA.


The screenshot shows a software interface for 'SA' (Spatial Analyzer) with 'Inst 0' selected. It features three input fields: 'Collection:', 'Group:', and 'Target:'. The 'Group:' field contains 'MeasA4' and has a dropdown arrow. To its right is a '+/-' button and an 'Inc' checkbox. The 'Target:' field contains 'P4' and has a '+/-' button. A red sphere icon is visible in the top right corner of the interface.

The  buttons to the right of the Group and Target fields make it easy to increment or decrement the names in these fields. Left-clicking them increment the names while right-clicking decrements the names down.

Enter MeasA4 into the Group field.

Left click the  button to the right of the Group field. The group name increments to MeasA5.

Repeatedly left-click on the button incrementing the Group name field up.


Right click the  button to the right of the Group field. The group name decrements back down.



Repeat the process with the Target field.

Enter P4 into the Target field.

Left click the  button to the right of the Target field. The target name increments to P5.

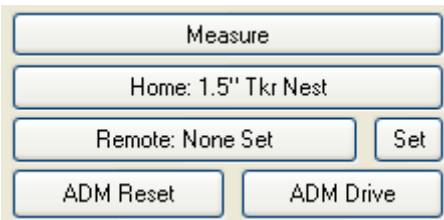
Repeatedly left-click on the button incrementing the Target name field up.

Right click the  button to the right of the Target field. The target name decrements back down.

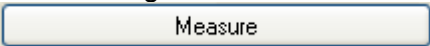
When the  option next to the Group name field is on  the Group name is automatically incremented after each measurement event. This is a handy feature when measuring groups of targets along a surface. Grouping each spline of targets along a surface makes it easier to develop surfaces from the measurement data. Other applications for automatically incrementing group names include making planes, cylinders, and other geometric shapes directly from the instrument interface. This functionality is covered in the Measurement Profiles section.

Measurement, Home, and ADM Controls

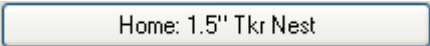
The controls for measuring, homing and driving the instruments ADM mode are grouped in one section of the interface.








Measure

After setting the Collection::Group::Target names and Measurement Profile click on the  button to start the measurement process. The interface starts the measurement process defined in the active measurement profile.


Home

The  button sends the instrument back to its home position and attempts to lock on the active reflector. The instrument interface will wait until it is able to successfully home or the operator aborts the process. A successful lock on the reflector sets the traffic light on the instrument interface to a green state. A table of traffic light status and their meaning is outlined below.


Description	Traffic Light
Tracker is not locked on a reflector. No range data.	
Tracker has a acquired the reflector but does not have a valid range to the target.	
Tracker is locked and the range was set with the interferometer offset value for the reflector/target.	
Tracker is locked and the range was set with the instrument's Absolute Distance Meter.	

The active reflector name is shown on the Home button's label. In the button shown above the active reflector is called [1.5" Tkr Nest]. Keeping the active reflector consistent with the one used for measurements is a critical responsibility for the operator. The active target/reflector is set using the  button on the interface.

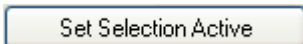
Remote Home Points

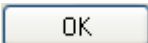
Establishing a remote position to reset the tracker's interferometer distance at a point is a widely used function. To define a remote home, press  in the main dialog. The Remote Home Set/Edit dialog will appear:

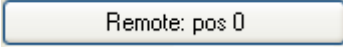


To add a remote home position, track the retro to the desired position, e.g., a hot-glued puck, etc., and press . The tracker interface will measure the position of the retro according to the parameters currently set in the Single Point measure mode, and the position will be added to the remote home database. The Remote Home dialog will then appear as follows:



Click on the  button to make this the active Remote Home position.

Press  button to save the remote database with its new entry.

The Remote Home dialog will close, and you'll notice that now the remote home button in the main dialog  button becomes active and indicates [pos 0] remote position is the active position. To try the remote home action;

Track the retro to, say, the tracker mounted nest.

Press .

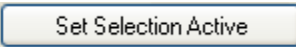
The Home/Remote Home dialog will appear, and the tracker IFM will be set to the remote home position, just as it is with the tracker mounted nest home position. To add another remote home position, just repeat the procedure.


Open the Remote Home dialog by click on the  button.

After pressing the Add button again, the Remote Home dialog will show two positions:



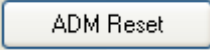
With two or more Remote Home position a third option, NEAREST, is added in the remote home database list. Any time two or more remote positions are defined, the NEAREST option appears. To try this option:


Select NEAREST by single left clicking in the list, then press  button.

The remote home button in the main dialog looks like . Indicating the active remote home position is now that which is closest to the retro at the instant the beam is broken.

Setting and configuring Remote Homes saves time measuring jobs when the tracker is not easily within reach of the object being measured.

ADM Reset and Drive

To set the distance of the tracker ranging system using the Absolute Distance Meter (ADM), point the tracker in the vicinity of the target then press the  button in the main dialog. The tracker then tries to lock on the retro-reflector using an outward-spiral search pattern until the target is located. When locked on the target the traffic light control turns green and has an ADM label. A timeout and search radius limits the length and extent of the search. Settings these limits are done with Tracker interface's menu option .

Driving a tracker to point at a target is done with the Keyboard Drive / ADM dialog. The  button opens the dialog up. The appearance of this dialog will vary slightly depending on the tracker in use, but is nearly identical for all trackers.



Keyboard arrow keys move the tracker the based on a number of other key board keys.

Holding the Ctrl key down while using the up arrow moves the instrument up vertically slowly while the down arrow key points it down the a similar amount.

The Shift key moves the instrument a little faster while the Ctrl + Shift keys held in while using the arrow keys moves the tracker head the fastest.

The right and left arrow keys similarly change the position of the tracker in the horizontal axes.

Point the instrument with the arrow keys within approximately 1 inches of the desired target.

Before clicking of the Search and Lock button set the seed distance field with an approximate distance between the instrument and target. The approximation should be within +/- 50 inches of correct value.

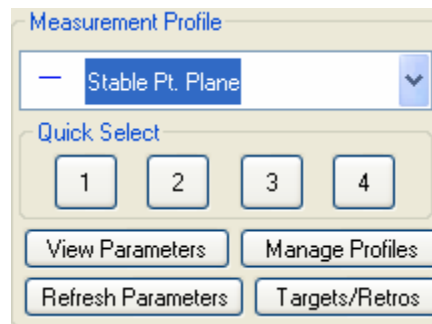
Click on the Search/Lock (F7) button to initiate the search for the retro-reflector. The F7 key is shortcut for this function.

After the tracker acquires the target and locks the traffic light on the main interface dialog will show a green state with the letters ADM in the middle.

Close this dialog after successfully acquiring the target.

Measurement Profiles and Target/Retro Interface

Configuring measurement parameters is an important feature control with the Tracker Interface. Measurement Profiles organizes measurement control into three distinct components, they are Target, Acquisition, and Operation. Each component is configurable and when setup together form a Measurement Profile. By dividing the control into these components the entire process is complete contained.

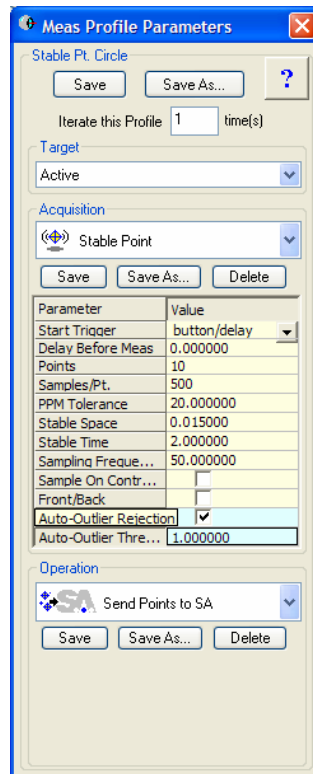


Getting familiar with configuring Measurement Profiles is an essential part of becoming proficient with the tracker interface. The initial effort to get profiles configured for a group of Tracker operators is valuable. Once a set of measurement profiles are defined using them to collect consistent data with different operators become much easier to manage. The sections below outline the process for configuring a number of different measurement modes with Measurement Profiles.

Statistical Filtering for Discrete Point Measurements

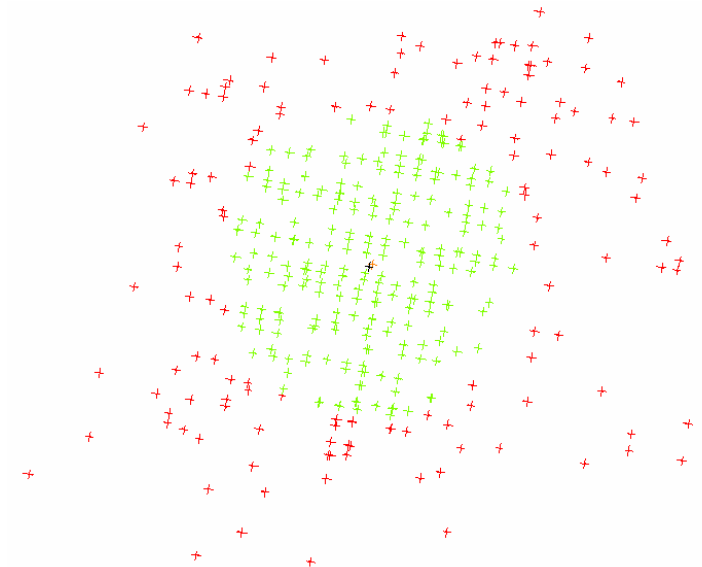
The SA Laser Tracker interface now allows for filtering outliers when measuring discrete, averaged points. This applies to the Discrete Point and Stable Point Acquisition modes, as well as discrete points solicited from SA, and discrete points measured during operational checks.

Shown here is a measurement profile with the new acquisition parameters in blue.



The full text for the threshold setting is “Auto-Outlier Threshold (Sigma)”, just hover over any clipped parameter label to see the full text. The default values are unchecked and 3, so that by default, behavior is unchanged. The threshold is an acceptance window size, so a smaller threshold means more filtering will occur.

Using the settings shown above, each sample was exported to SA



This is a view looking approximately along the normal of the tracker 'pancake'. Green points are used, red points are filtered outliers, orange is the initial average, and black is the final average, taken from only green points. The measurement was taken from a distance of 407.67", or about 34'.

For reference, here is a view looking from the side of the 'pancake'.



Below are the measurement details for the resulting point. 500 samples were taken, and 161 were rejected by the 1 sigma filter. In this case, the rms error BEFORE the sigma filter was applied was 0.001807". As seen in the measurement details, the rms after the filter was 0.001192", reduced by almost 1/2.

Polar Measurement

API 3290
RMS Error 0.001192 in
161 Samples Rejected by Sigma Filter
Weather: T=68.8F, P=29.5inHg, H=50.0%
Discrete Point Parameters:
Start Trigger= button/delay
Delay Before Meas=0.000000
Points=1
Samples/Pt.=500
PPM Tolerance=20.000000
Sampling Frequency=50.000000
Front/Back=False
Use Sigma Filter=True
Sigma Filter=1.000000

☒ Use In Calculation
☒ Draw Me

Values

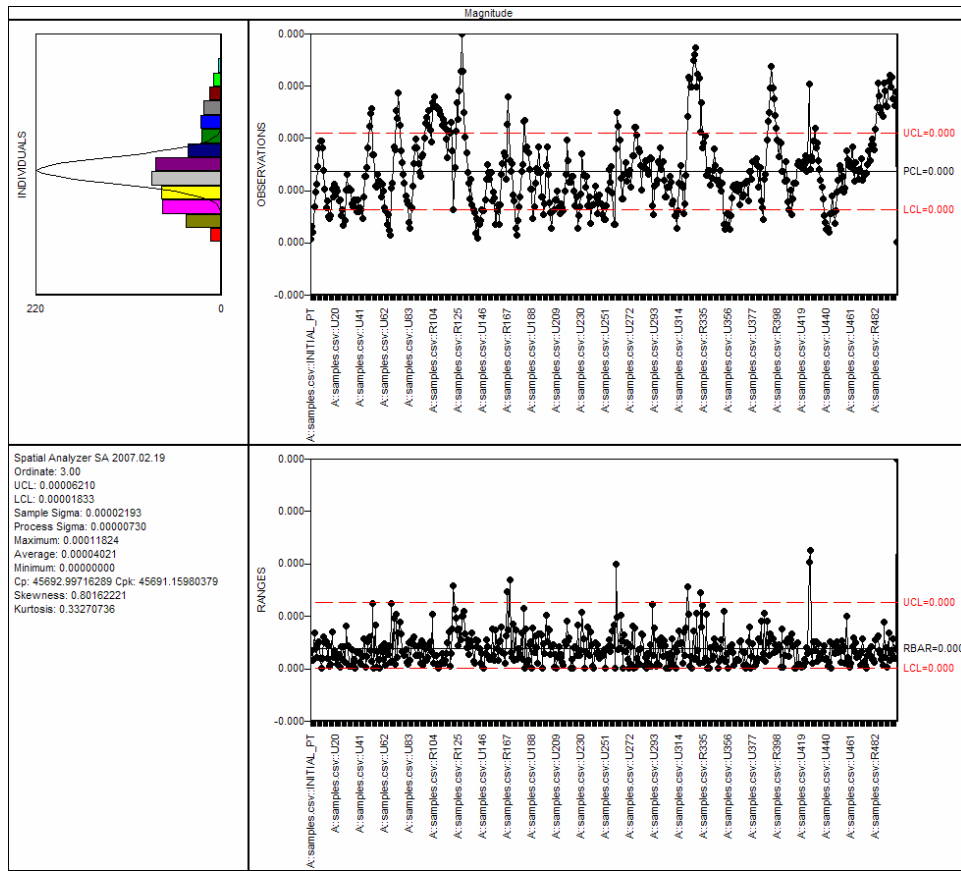
Theta	-154.79665422
Phi	96.77102954
Z	407.66832984

Target Offsets

0.75	0.75
------	------

OK Cancel

Shown below is the SPC chart for the sample data.



To reproduce a similar test, one can use the temporal scan mode in the tracker interface. Just scan points into a single group. Then in SA, Construct >> Point(s) >> Fit to Points using the scanned group. This makes a point at the average of the selected points. Now Query>>Points to>>Single Point. This makes a vector group. Then in the vector group properties dialog, pick "Make New Chart". This creates a chart like the one shown above.

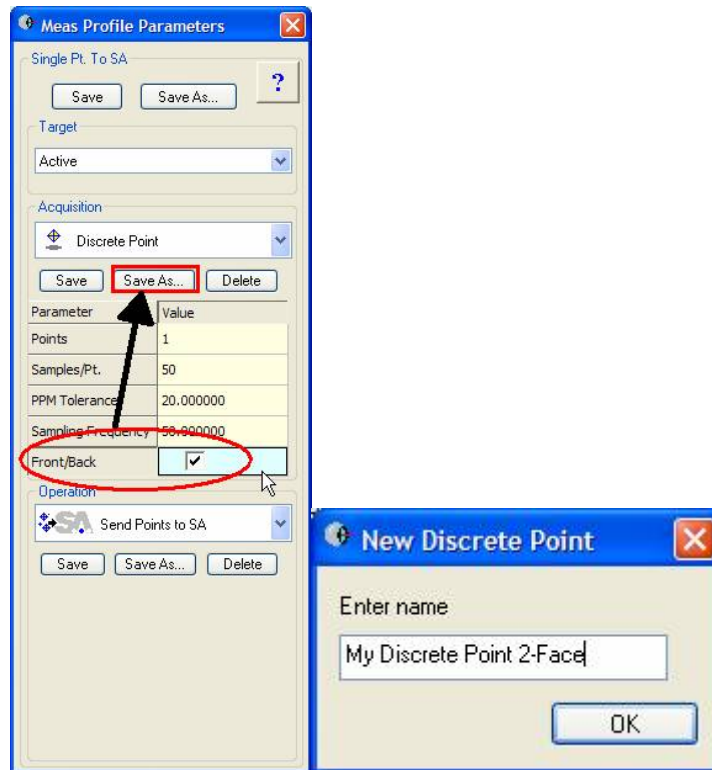
Two Face Measurements

Make the Single Pt to SA measurement profile active ... the default shortcut for Single Pt to SA is to click the number 2 button.

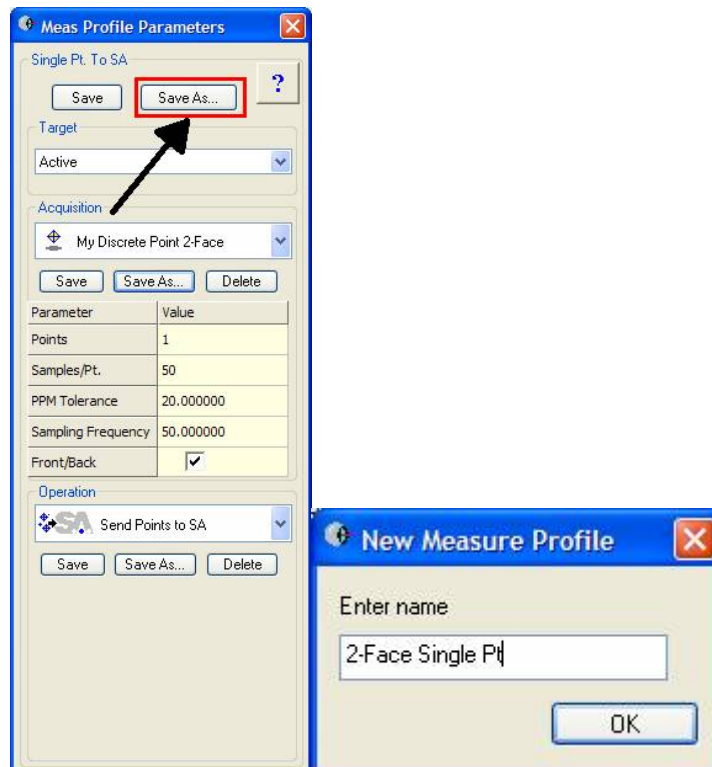
Click on the View Parameters button to configure the measurement.

Set the front/back sight measurement mode to true.

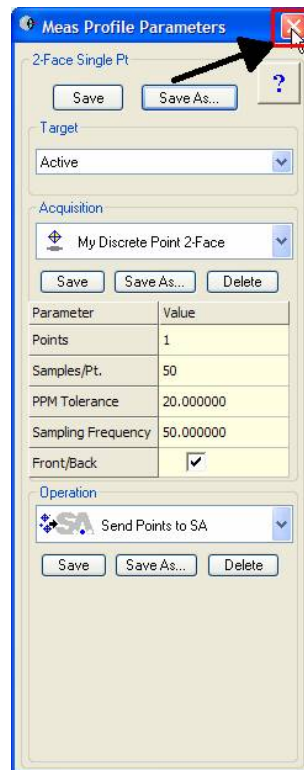
Set the Acquisition mode name to My Discrete Point 2-Face



Set the Measurement Profile to 2-Face Single Pt

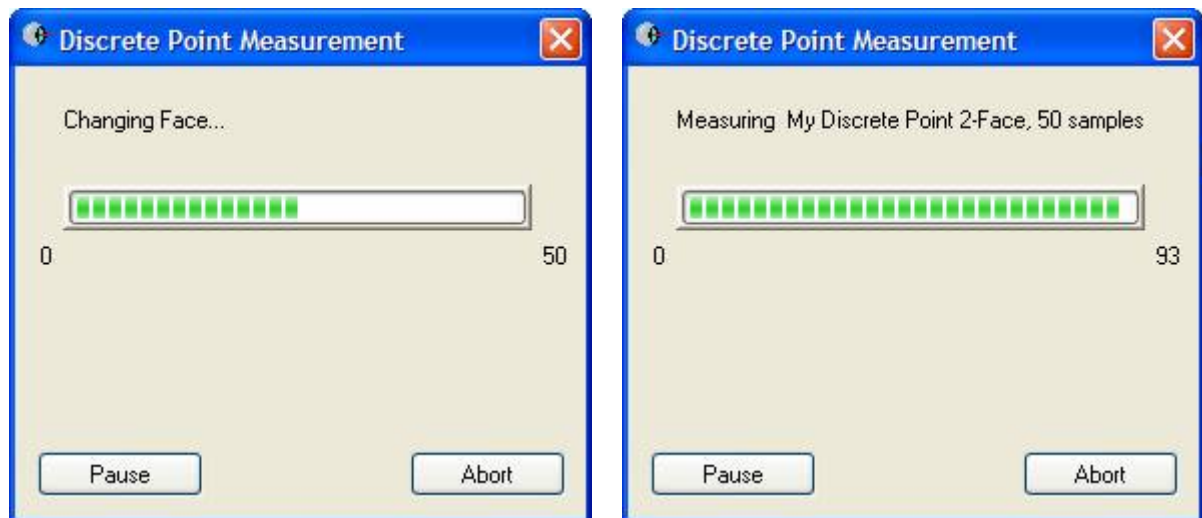


Close the Measurement Profile Box



With the new Measurement Mode active and the tracker locked on a reflector click on the measurement button.

The process should start, collecting measurements from the current face, then change face and keep going.



After the front sight measurement the Tracker changes faces. The dialog above

The interface completing it measurements

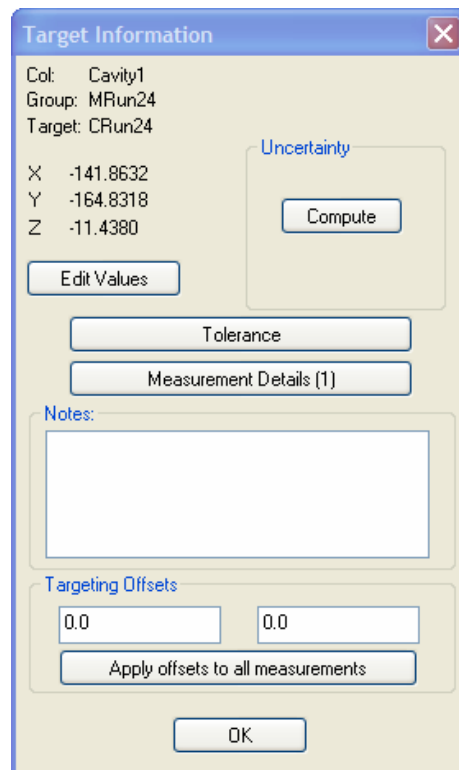
shows the status change.

on the second face.

Following the measurement on the second face the instrument returns to face one and re-acquires the target. The range to the target is typically set to the initial value measured range.

Go back to SA and open the Target Properties dialog for measurement point.

Select the Measurement Details button.

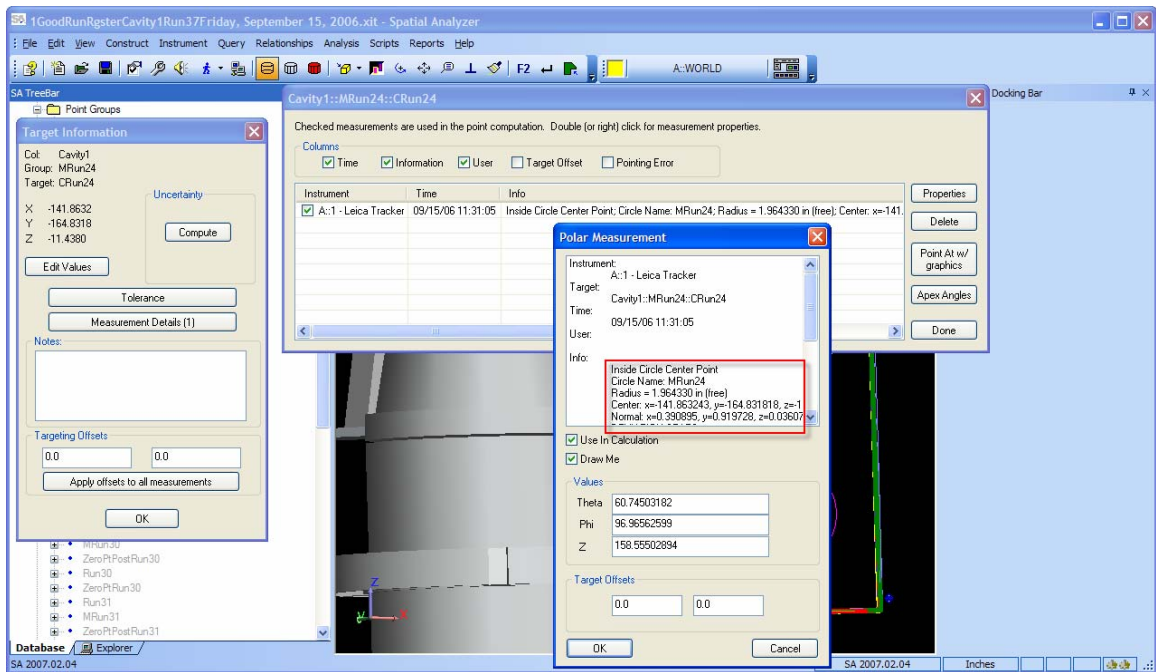


The image shows a 'Target Information' dialog box with a blue title bar and a close button. It contains the following fields and buttons:

- Col: Cavity1
- Group: MRun24
- Target: CRun24
- X: -141.8632
- Y: -164.8318
- Z: -11.4380
- Buttons: Edit Values, Compute (under Uncertainty), Tolerance, Measurement Details (1)
- Notes: (empty text area)
- Targeting Offsets: Two input fields both containing 0.0, and an Apply offsets to all measurements button.
- OK button at the bottom.

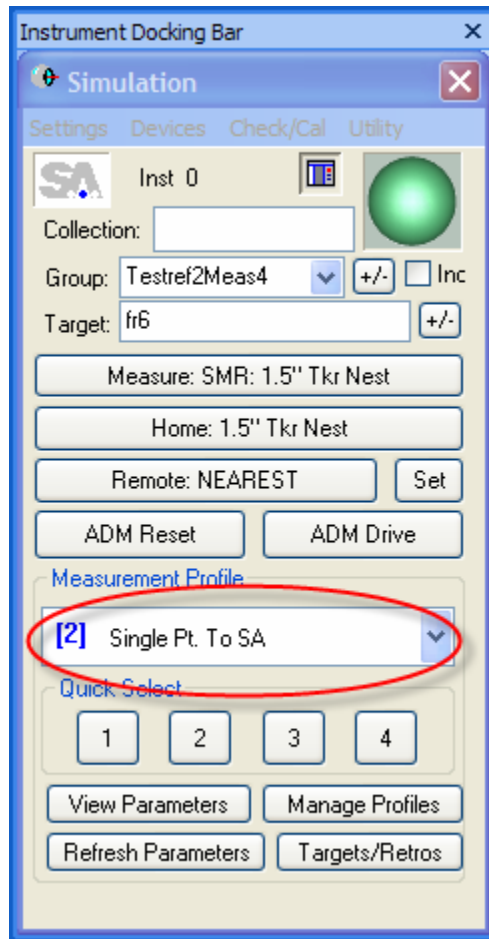
Double click on the measurement to view the properties for this observation.

Review the details in the Information View for the measurements.



Single Pt. to SA

The default Single Pt. to SA profile measures a single stationary point by averaging a specified number of samples.



Parameters

Samples to Average – number of samples to take and average to calculate resulting point

F/B – Front/Back; when checked, the specified number of samples is acquired in both front and back faces

Delay Before Measurement – amount of time in seconds to wait after Measure button is pressed and before measurement begins

Stop Trigger

of Points – Measurement stops when specified number of samples has been taken or interrupted by the user

Stable Point

The Stable Point profile measures a single averaged point when the retro-reflector has not been moving for a specified time within a specified spatial tolerance, then waits for retro to be moved again and starts over. Used for measuring discrete positions where the retro can be placed in pucks or other retro retaining devices.

Parameters

Samples to Average – number of samples to take and average to calculate resulting points

Stability thresholds – the temporal and spatial stability thresholds are preset for specific trackers, and cannot be changed by the user

Stop Trigger

of Points triggers the end of this acquisition mode

Watch Update

SpatialAnalyzer Watch Windows are used with instrument interfaces to show real-time feedback in SA's graphics environment. With a Watch Window on a Point Group open; move the SMR to some previously measurement points. The watch window updates the delta between the current SMR location and the existing point. A graphical arrow indicator in SA gives an indication of the current closest point.

The Watch Update measurement profile sends data updates (no points are created) to SA for Watch Windows. .

Parameters

Frequency – Updates per second to send to SA

Stop Trigger

Beam Break which is an Open ended – measurement continues until stopped by the user

of Point completes the acquisition mode after the specified number of points is collected.

Loop completes the acquisition mode when the last measured point approached the first. This mode is effective for measuring circles. The mode completes when the reflector make a complete 360 circuit around the circle.

Spatial Scan

The default Spatial Scan measurement profile sends a stream of measured points to SA. The points are separated by a specified spatial increment. This profile is commonly used for scanning surfaces or geometries to be analyzed in SA.

Parameters

Spatial Increment – point separation

Delay Before Measurement – amount of time in seconds to wait after Measure button is pressed and before measurement begins

Stop Trigger

Beam Break which is an Open ended – measurement continues until stopped by the user

of Point completes the acquisition mode after the specified number of points is collected.

Loop completes the acquisition mode when the last measured point approached the first. This mode is effective for measuring circles. The mode completes when the reflector make a complete 360 circuit around the circle.

Sphere Surface

The Sphere Surface measurement profile is typically used to acquire a specified number of measured points which are separated by a specified spatial increment. Upon successful completion of the measurement, the data is fit to a sphere, and the user is given an opportunity to accept or reject the fit. The resulting sphere is offset by the radius of the Spherically Mounted Retro-reflector. If accepted, the user can send the geometry and/or the measured points to SA.

Parameters

Spatial Increment – point separation

Number of Points – number of points to measure on the sphere to be used in the fit

RMS Tolerance – error magnitude per point which is acceptable to the user. If any point's error magnitude is greater than the tolerance, the user will be notified after the fit

Lock Radius – if checked, the value in the adjacent edit box is used to set the radius, and only the location of the sphere center is calculated in the fit. This is commonly used when measuring tooling balls of know radii.

Delay Before Measurement – amount of time in seconds to wait after Measure button is pressed and before measurement begins

Stop Trigger

of Points – Measurement stops when specified number of points has been taken or interrupted by the user

Beam Break which is an Open ended – measurement continues until stopped by the user

Loop completes the acquisition mode when the last measured point approached the first. This mode is effective for measuring circles. The mode completes when the reflector make a complete 360 circuit around the circle.

Scan Circle (Pin) and Scan Circle (Hole)

These two default measurement profiles are used to acquire a specified number of measured points then create a circle. The Pin profile projects the circle inward toward the center of the circle. An outward projection is performed with the Hole profile. The

user is given an opportunity to accept or reject the fit if the Show Dialog option is set within the Operation section. If accepted, the user can send the geometry and/or the measured points to SA.

Parameters

Spatial Increment – point separation

of Points – number of points to measure on the sphere to be used in the fit

Rejection Tolerance – error magnitude per point which is acceptable to the user. If any point's error magnitude is greater than the tolerance, the user will be notified after the fit

Delay Before Measurement – amount of time in seconds to wait after Measure button is pressed and before measurement begins

Stop Trigger

of Points – Measurement stops when specified number of points has been taken or interrupted by the user

Beam Break which is an Open ended – measurement continues until stopped by the user

Loop completes the acquisition mode when the last measured point approached the first. This mode is effective for measuring circles. The mode completes when the reflector make a complete 360 circuit around the circle.

Scan Plane

The Scan Plane profile acquires a specified number of measured points which are separated by a specified spatial increment. Upon successful completion of the measurement, the data is fit to a plane, and the user is given an opportunity to accept or reject the fit. If accepted, the user can send the geometry and/or the measured points to SA.

Parameters

Spatial Increment – point separation

Number of Points – number of points to measure on the sphere to be used in the fit

Rejection Tolerance – error magnitude per point which is acceptable to the user. If any point's error magnitude is greater than the tolerance, the user will be notified after the fit

Delay Before Measurement – amount of time in seconds to wait after Measure button is pressed and before measurement begins

Stop Trigger

of Points – Measurement stops when specified number of points has been taken or interrupted by the user

Beam Break which is an Open ended – measurement continues until stopped by the user

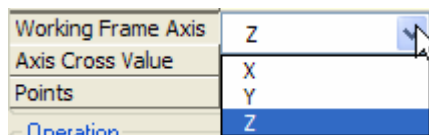
Loop completes the acquisition mode when the last measured point approached the first. This mode is effective for measuring circles. The mode completes when the reflector make a complete 360 circuit around the circle.

Cross Section

The Cross Section profile acquires a point whenever the retro-reflector center crosses a specified plane. The plane is defined by the current working frame in SA.

Parameters

Spatial Increment – point query separation



– plane of measurement. For example, if X is selected and the value is 5 inches, then points will be acquired when the retro-reflector center crosses the plane parallel to the YZ plane at X=5. Note: This plane is defined relative to the current Working Frame in SA.

Delay Before Measurement – amount of time in seconds to wait after Measure button is pressed and before measurement begins

Stop Trigger

of Points – Measurement stops when specified number of points has been taken or interrupted by the user

Beam Break which is an Open ended – measurement continues until stopped by the user

Loop completes the acquisition mode when the last measured point approached the first. This mode is effective for measuring circles. The mode completes when the reflector make a complete 360 circuit around the circle.

Modify Sphere Profile to configure Tooling Ball Profile

Measure spheres e.g., Tooling Balls can be accomplished quickly with the Tooling Ball measurement profile.

Meas Profile Parameters

Tooling Ball

Save Save As... ?

Target

Active

Acquisition

Spatial Scan

Save Save As... Delete

Parameter	Value
Start Trigger	button/delay
Delay Before Meas	0.000000
Pause @ BeamBr...	<input type="checkbox"/>
Stop Trigger	# points
Increment	0.050000
Sampling Frequency	50.000000
Points	100
Samples/Pt.	1

Operation

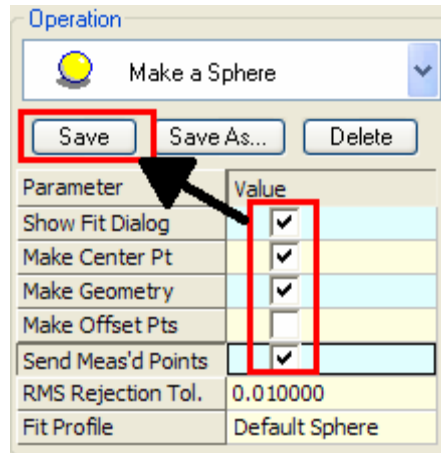
Make a Sphere

Save Save As... Delete

Parameter	Value
Show Fit Dialog	<input type="checkbox"/>
Make Center Pt	<input checked="" type="checkbox"/>
Make Geometry	<input type="checkbox"/>
Make Offset Pts	<input type="checkbox"/>
Send Meas'd Points	<input type="checkbox"/>
RMS Rejection Tol.	0.010000
Fit Profile	Default Sphere

The figure above shows the default Tooling Ball profile. Review these options.

Change the Operations check configuration to include Show Fit Dialog, Make Geometry, and Send Meas'd Points.

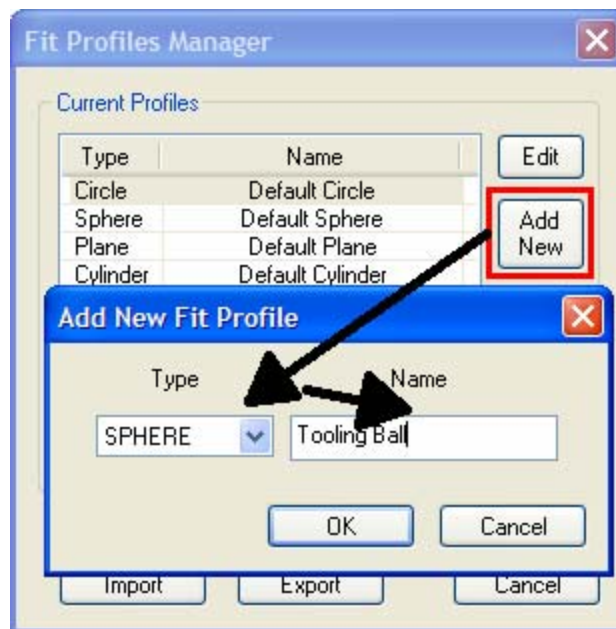


After setting the new options click on the Operations Save button.

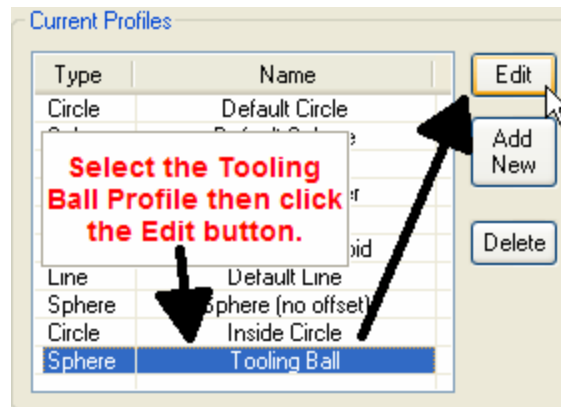
When solving for a specific geometry type configuring the profile to solve with specific parameters is accomplished with Fit Profiles. The next steps show the process for configuring for a 1/2" tooling ball. Once a fit profile is setup and configured changing the behavior of a measurement profile is easy.

Right-click in the Fit Profile field called Default Sphere.

Click on the Add New button.

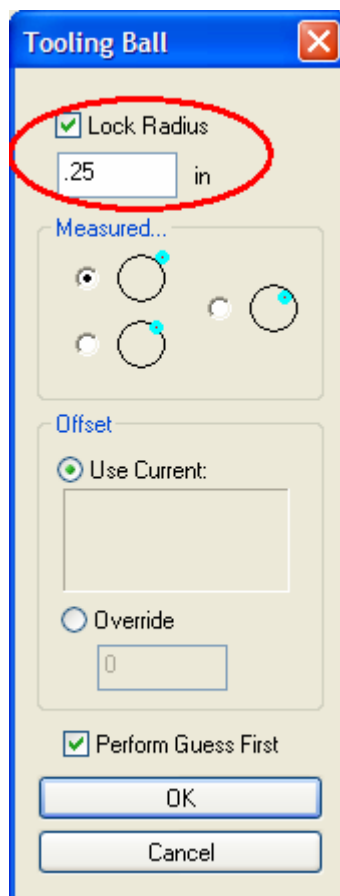


Set the new Fit Profile type to SPHERE then name it Tooling Ball. Click on the OK button.



Select the Tooling Ball profile then click the Edit button.

Lock the Radius to 0.25-inches for the Tooling Ball fit profile.

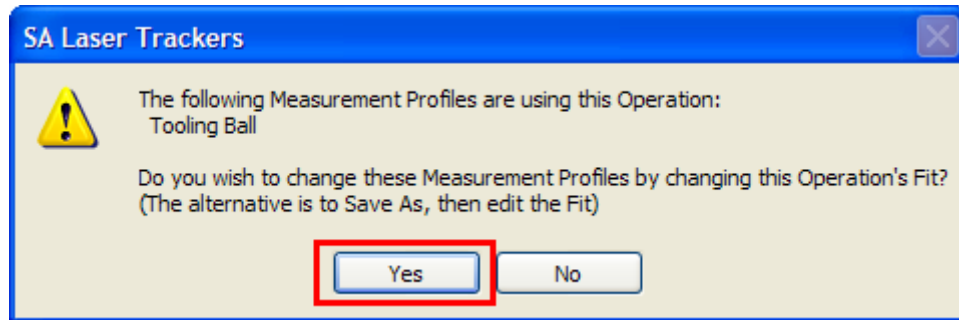


Leave the default reflector offsets as shown in the figure above.

Click OK on the Tooling Ball Profile.

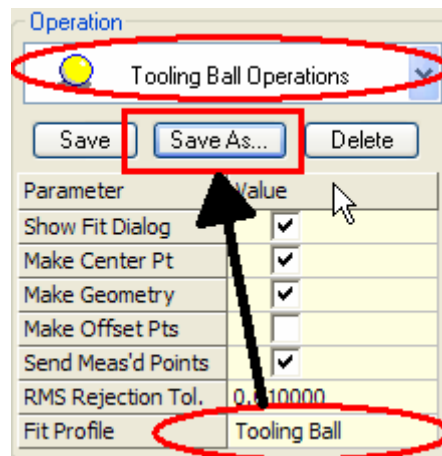
Click OK on the Fit Profiles Manager.

Select the Yes option when asked if using this Fit Profile for the current Measurement Profile.



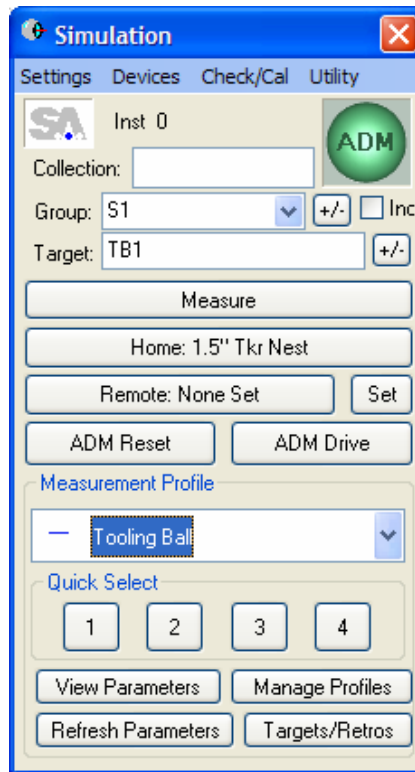
Select the Operation Save As... button on the Measurement Profile.

Rename the Operation to Tooling Operations then select the OK button.

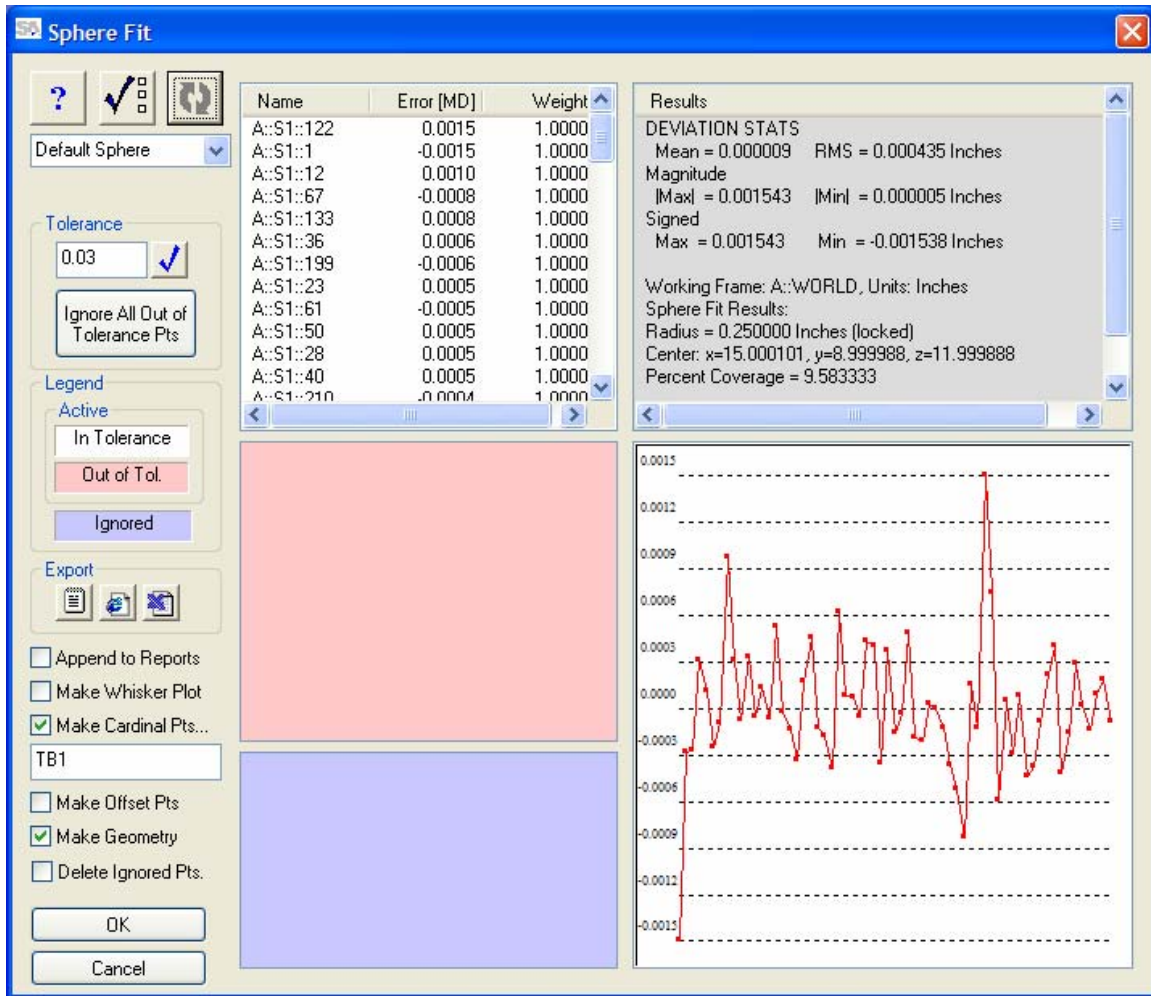


Select the Save button on the top of the Measurement Profiles dialog then close the Measurement Profiles dialog.

In the tracker interface, make "Sphere" the Group name and "P0" the Target name.



Setup a ½" tooling ball in a stable mount. With a measurement partner at the computer, tracker the retro-reflector out to the sphere. When ready, start scanning the surface of the tooling ball, then ask the partner to click the measure button on the interface.



Setting a Measurement Delay

Note the setting. This setting can be used for most all measure modes. When working alone, it sets a delay to allow you to track the SMR to the target, geometry, etc. before the measurement begins. Just set it to an appropriate value.

Press You'll see the measure progress dialog again, showing input parameters and counting down the delay before measurement if you set one.

If you are working alone, and you don't make it to the sphere before the measurement begins, don't worry. Just the measurement, or the residuals if the fit already occurred. Then you can reset the to a higher value to allow more time, and start again.

Touch the SMR to the surface of the sphere. When the measurement begins, smear the SMR over the sphere, being sure to maintain contact and attempting to maximize coverage of the sphere.

If you can see the computer screen, you'll notice the measure progress dialog counts up the points to show you where you are. If not, the interface gives a beep when the

measurement is done. This is true for any measure mode which requires a set number of points to be taken.

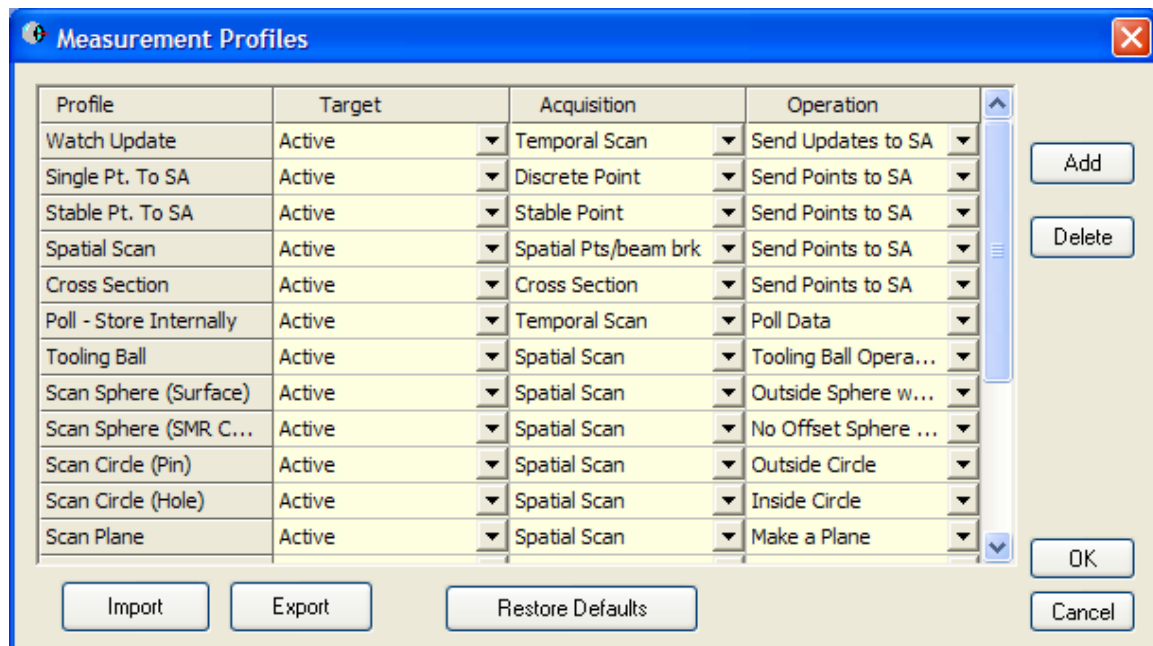
When the measurement is complete, place the SMR in a convenient puck or nest. You'll see the Residuals Dialog, which shows the fit report including geometry results, percent coverage, and errors.

Provided the results are acceptable, select and/or by checking the corresponding boxes. Press Accept.

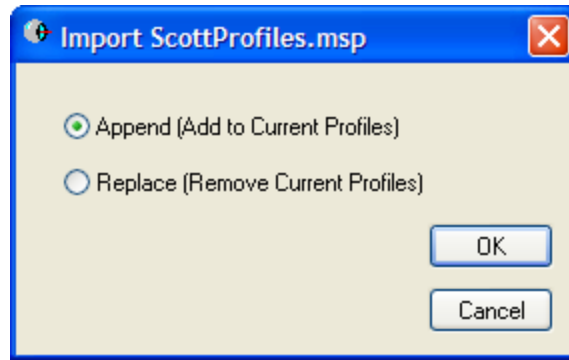
This completes the sphere surface measurement. The specified data will be sent to SA, and SA will update accordingly.

Managing Measurement Profiles

Use the Manage Profiles button to import and export measurement profiles between tracker interfaces.



Exporting profiles to configure all tracker systems to have the same set of measurement a process helps to minimize data collection variability. When importing measurement profiles an option for appending the new profiles to the existing ones verses replacing the current profiles help control the profiles on the target machine.



Tracker Interface Configuration Menu

Configuring the interface and performing field checks are accessed through the Tracker Interface Configuration menu at the top of the dialog.

Settings Devices Check/Cal Utility

In general the process to setup the interface is the same for each type of tracker. Differences in performing calibrations and compensations are typically managed by calls to the specific tracker manufacturer's libraries.

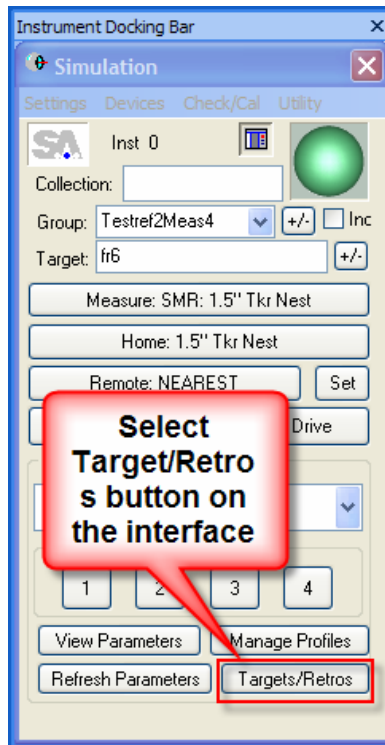
The Settings menu has controls for configuring typically tracker behavior like ADM search radius, point sampling requests from SA, units, and UI controls. Controls for environmental (e.g., weather) and levels compensations are accessed through the Device menu. Field checks and other compensations managed within the interface are accessed under the Check/Cal menu. Tracker status, utilities, and ASCII imports for angles and range data are available under the Utilities menu.

Review the functions and methods of each of these menus. Configure each option and then exit the tracker interface to save these settings. The next time the tracker interface is started each setting is recalled from a configuration file saved in the Analyzer Data\Persistence directory.

Adding Reflectors and Targets to the Interface

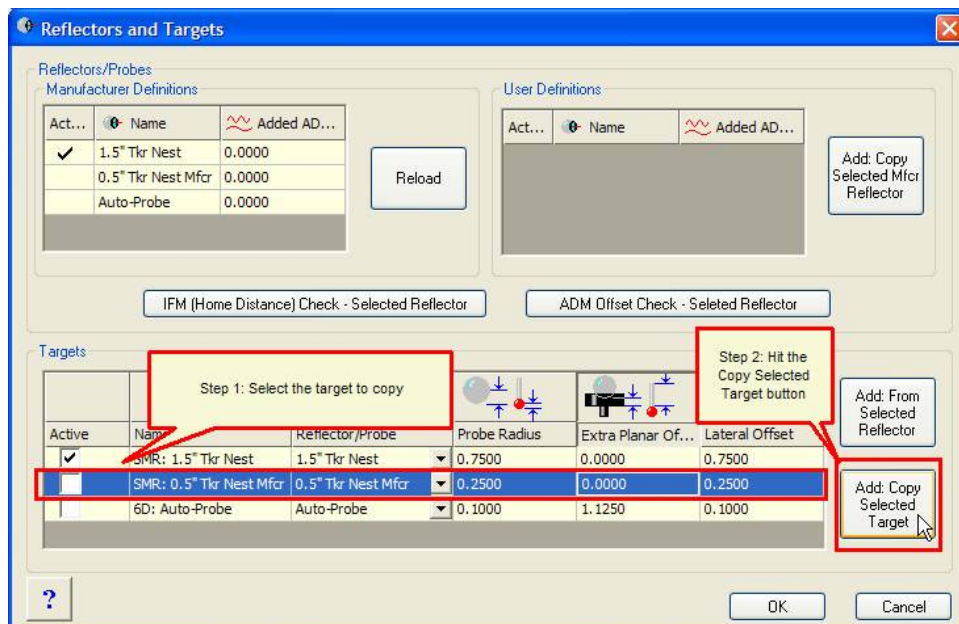
The details below show the process for adding targets and reflectors to the tracker interface.

To add target/reflectors select the Target/Retros button on the Tracker Interface.



Creating a new retro-reflector in the interface involves a process of coping a like type reflector and then setting its properties. After setting the properties; routines to check the properties are available within the interface. The operations checks are important for confirming the setting for each reflector used in the survey.

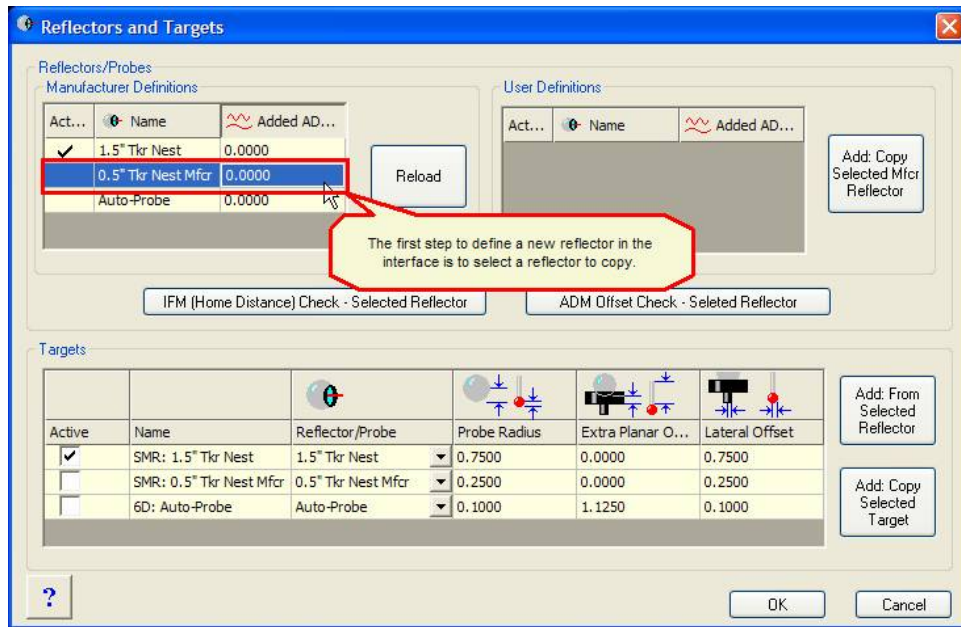
There are two processes to add a new target into the interface. The fast method is to select a reflector in the Targets table and hit the Add: Copy Selected Target button.



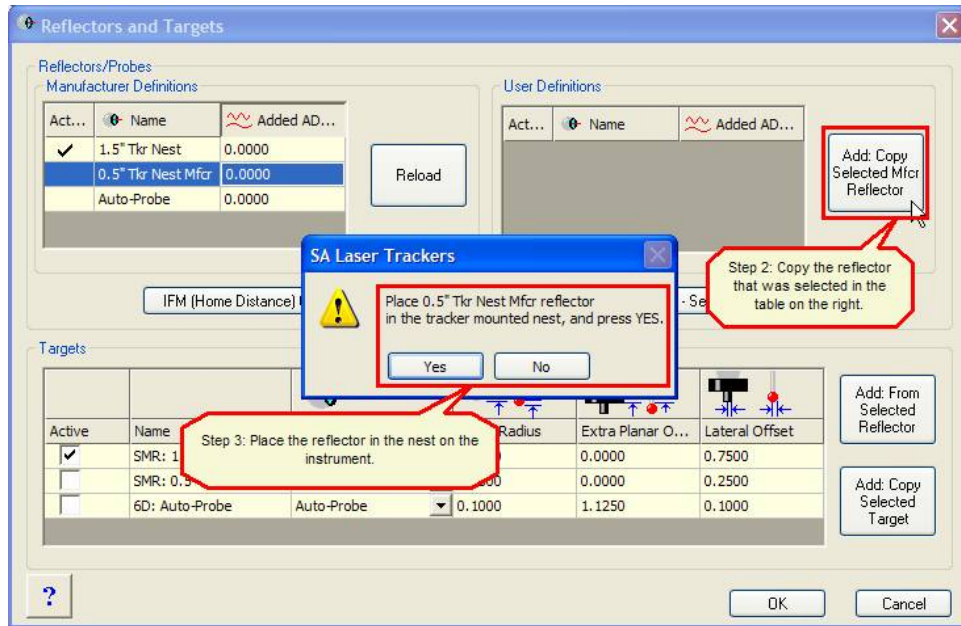
A new target is added to the list where you can set its mechanical offsets. This process doesn't allow a new reflector type. The procedure outline below is used to add new reflector types.

To add a new Reflector, as in a solid 0.5SMR solid(Prism Type) please follow the following process steps below. This procedure has been added to manual. A new release of the manual will be out in the near term.

Step 1: Select the reflector to copy...



Step 2: Add a copy of the selected reflector

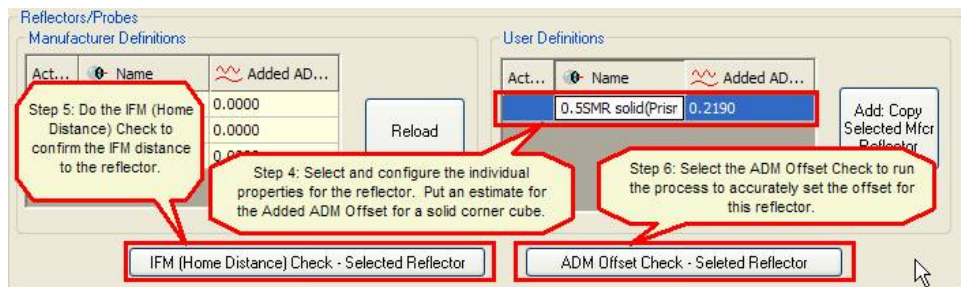


Step 3: Place the reflector in the tracker's nest and then hit the Yes button on the message box (see the figure above).

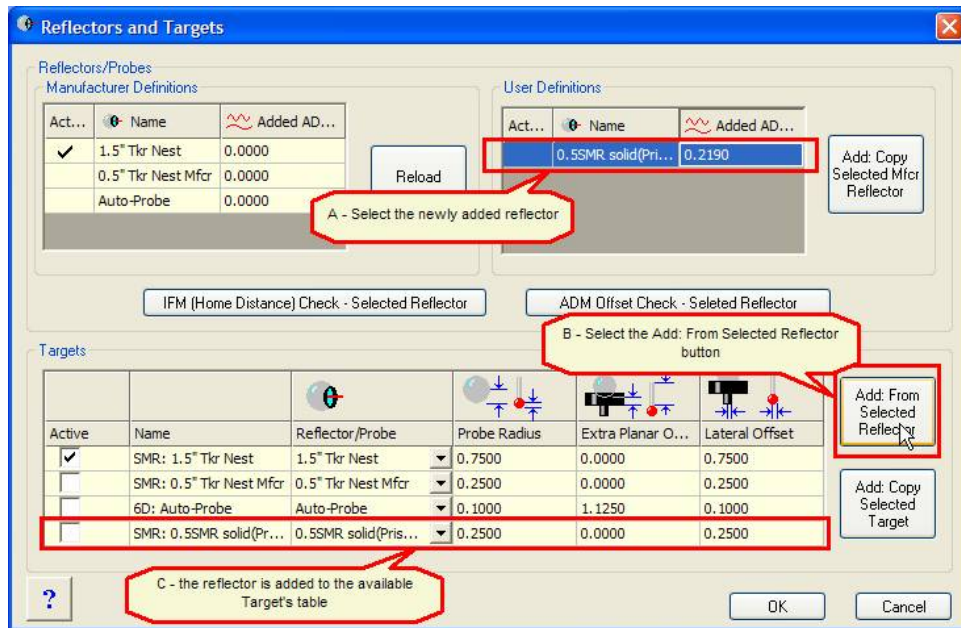
Step 4: Set the reflector properties for the newly added reflector.

Step 5: Run the IFM (Home Distance) Check for the reflector to confirm its initial Interferometer distance. The procedure should be repeated until the change to the initial distance is less than 0.0005 inches [0.013 mm].

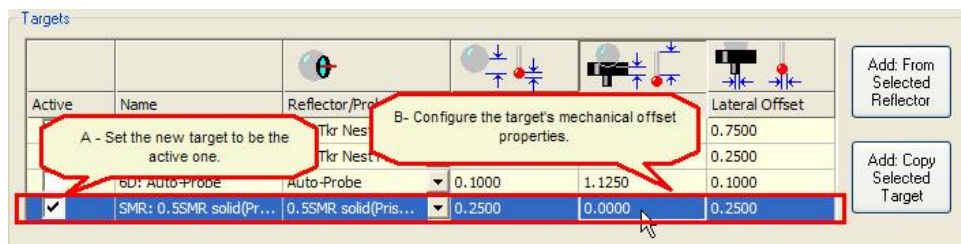
Step 6: Run the ADM Offset Check. A minimum of 4 IFM and ADM distances at different ranges are required to compute the ADM offset for a reflector. Please use distances that include minimum and maximum ranges used by the instrument.



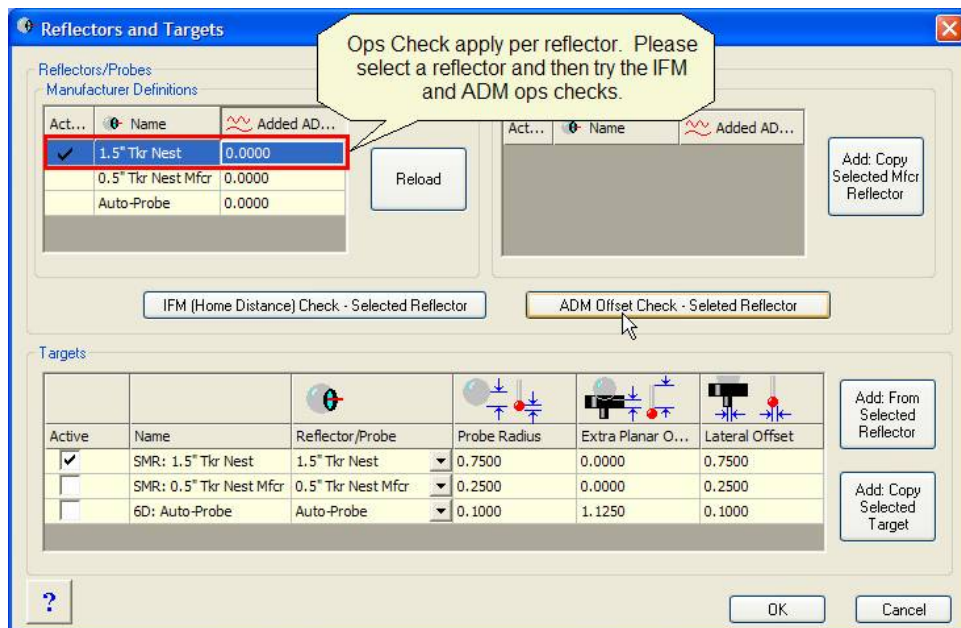
Step 7: To add the reflector into the list of targets. Select it and then hit the Add: From Selected Reflector button in the target section of the dialog.



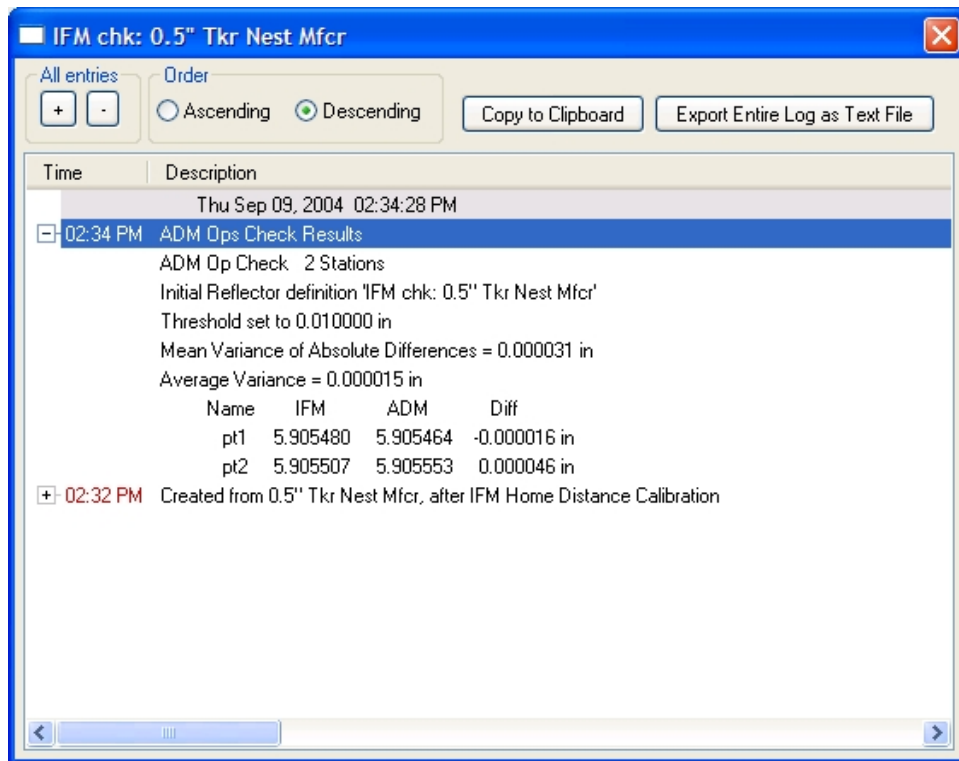
Step 8: Make the new reflector the active one and configure its specific offset properties.



Please remember that the Ops Check apply per reflector.



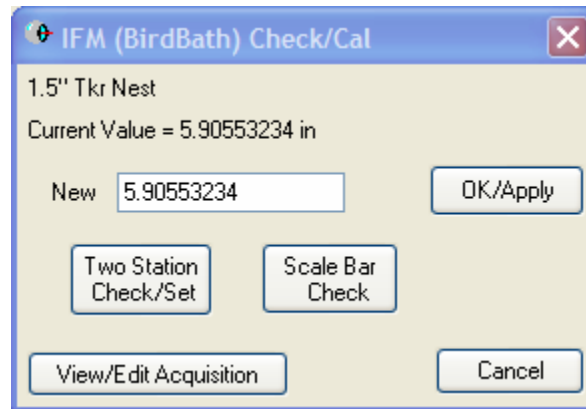
To review a reflector's history right clicking on it to see a dialog like that shown below.



Ops Checks / Field Checks

To check the pointing performances of the tracker perform a Two Face Operations Check. Specifically, the Two Face check gives us an indication of orthogonality of the azimuth and elevation axes in the tracker. It measures discrete positions from both the front and back faces, and presents the angular discrepancies in each measurement pair.

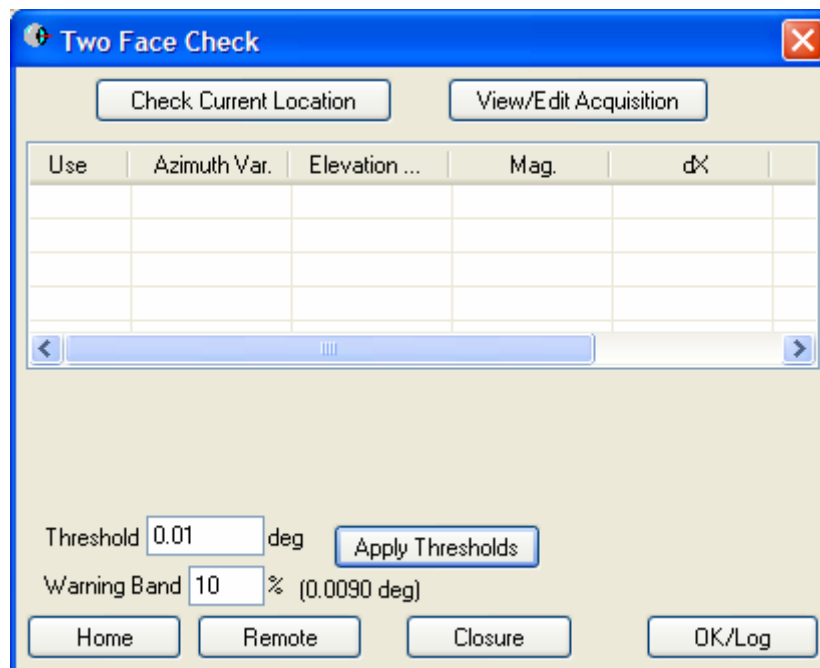
The IFM (Home Distance) check is used to confirm the initial interferometer distance that is used to set the bird bath distance. This is the basic reference distance used for all range measurement. Since each reflector e.g., 19.05 and 6.35 mm can be centered differently performing the check is recommended for each reflector. When the find the delta in the IFM check to be more than about 0.013 mm then you should run the tracker specific calibration/compensation functions to reset the initial distance.




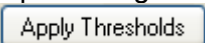
The ADM Offset Check compares the IFM distance against the distance measured by the ADM. If the distances between the two differ by more than the trackers spec then you should do a QVC to reset the ADM calibration. This function is also used to solve for ADM offsets in specific reflectors. Open air corner cubes have by default, a zero ADM offset. Solid corner-cubes behave differently when measured by the IFM verses the ADM. The ADM Offset Check can be used to solve for this delta between the IFM and ADM behavior on solid corner cubes. The difference is saved with the reflector. When that reflector is used in an ADM measurement the offset is applied to the distance SA gets from the tracker.

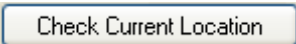
Go to the Check/Cal menu on the Tracker Interface to expand the ops and cal options of this instrument controls.

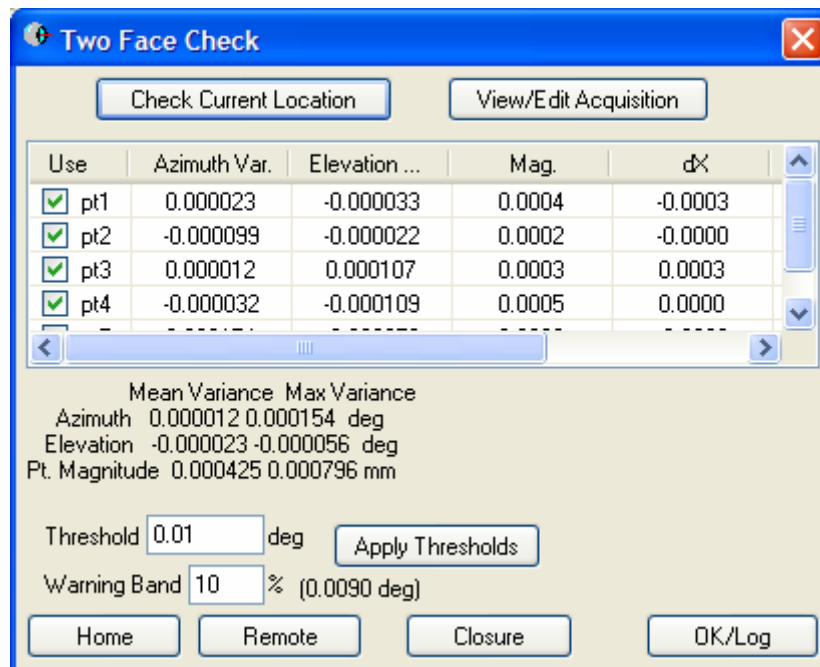
Select the menu option for Two Face Check will appear.



Track the SMR to a puck and press  button. The tracker will measure the current position, about face and measure again, and about face back. The angle deltas from the first measurement are displayed in the grid on the dialog.

Note that the and percentage define how the results are colored in the grid. If they change press the  button. The data display will change accordingly.

Continue to check locations by tracking to different puck positions and pressing . The key is to vary the azimuth and elevation as much as possible.




The dialog box titled "Two Face Check" contains the following elements:

- Buttons: "Check Current Location" and "View/Edit Acquisition".
- Table with 5 columns: Use, Azimuth Var., Elevation ..., Mag., dX.

Use	Azimuth Var.	Elevation ...	Mag.	dX
<input checked="" type="checkbox"/> pt1	0.000023	-0.000033	0.0004	-0.0003
<input checked="" type="checkbox"/> pt2	-0.000099	-0.000022	0.0002	-0.0000
<input checked="" type="checkbox"/> pt3	0.000012	0.000107	0.0003	0.0003
<input checked="" type="checkbox"/> pt4	-0.000032	-0.000109	0.0005	0.0000
- Summary statistics:

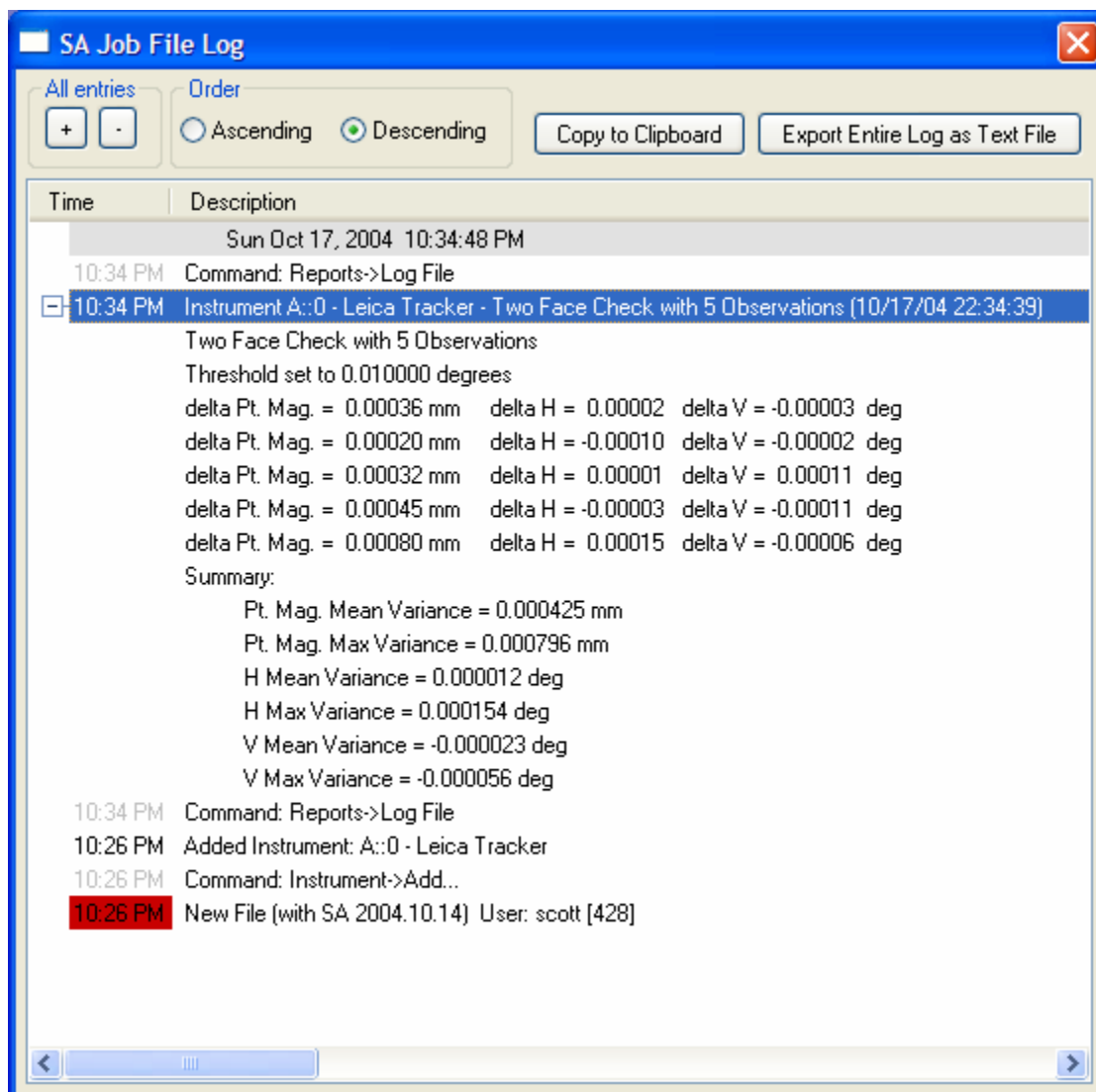
Mean Variance Max Variance
 Azimuth 0.000012 0.000154 deg
 Elevation -0.000023 -0.000056 deg
 Pt. Magnitude 0.000425 0.000796 mm
- Threshold settings:

Threshold deg 

Warning Band % (0.0090 deg)
- Bottom buttons: "Home", "Remote", "Closure", and "OK/Log".

When you have measured several locations, press the  button.

In all cases, Ops Checks results are logged in the SA log file. To view the log file, go back to SA and select the menu option. The results of each ops check completed in this job are stored in the job file for later review.



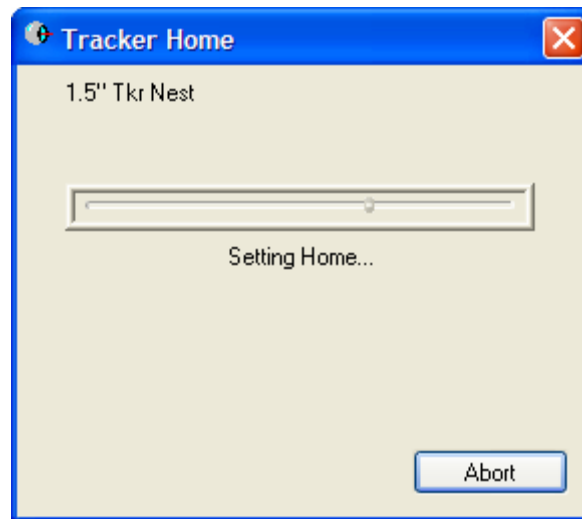
Continue through the rest of the Operational Checks on the Check/Cal menu on the Tracker Interface.

Setting and Monitoring the IFM Beam

Due to the need for locking to a known distance for IFM measurement, there are many provisions for re-locking and monitoring the state of the IFM beam in the SpatialAnalyzer Laser Tracker Interface.

All trackers are fitted with a retro-reflector nest on the body of the tracker. These are commonly referred to as TMN's (Tracker Mounted Nests) or Bird Baths. The TMN provides what is commonly called the Home position. Given the proximity and rigidity with respect to the tracker head, the TMN is the most dependable and accurate way to set the initial IFM distance value. In fact, when the interface is started and after connection and initialization, the interface automatically homes the tracker as described previously in "Starting the Interface". However, the IFM can be reset to the home

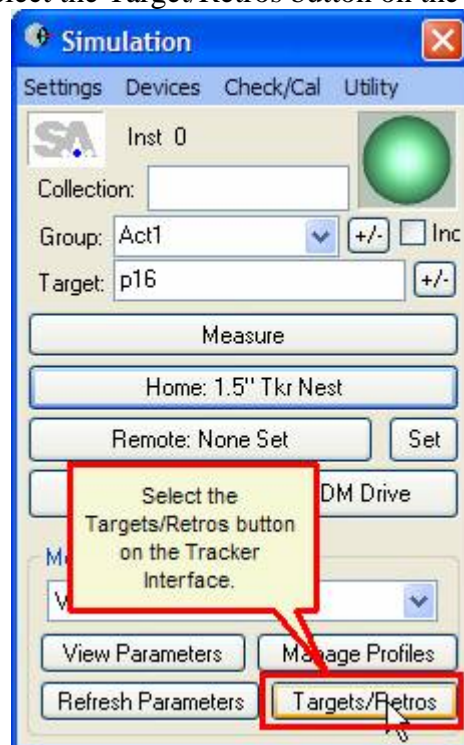
distance at any time. To try it, just press Home: 1.5" Tkr Nest in the main dialog. The Home Progress dialog will appear:



Adding Reflectors and Targets to the Interface

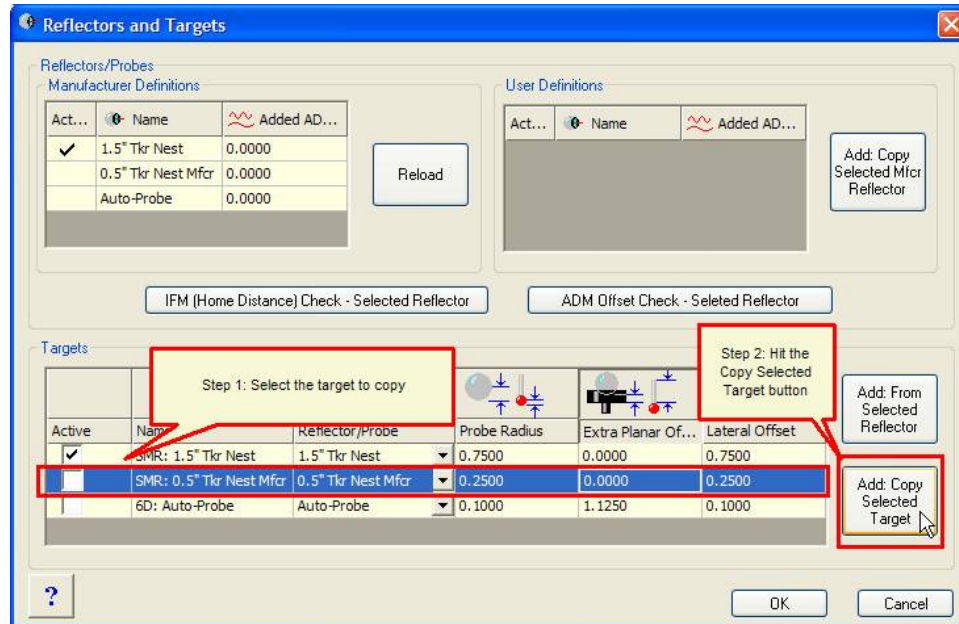
The details below show the process for adding targets and reflectors to the tracker interface.

To add target/reflectors select the Target/Retros button on the Tracker Interface.



Creating a new retroreflector in the interface involves a process of coping a like type reflector and then setting its properties. After setting the properties; routines to check the properties are available within the interface. The operations checks are important for confirming the setting for each reflector used in the survey.

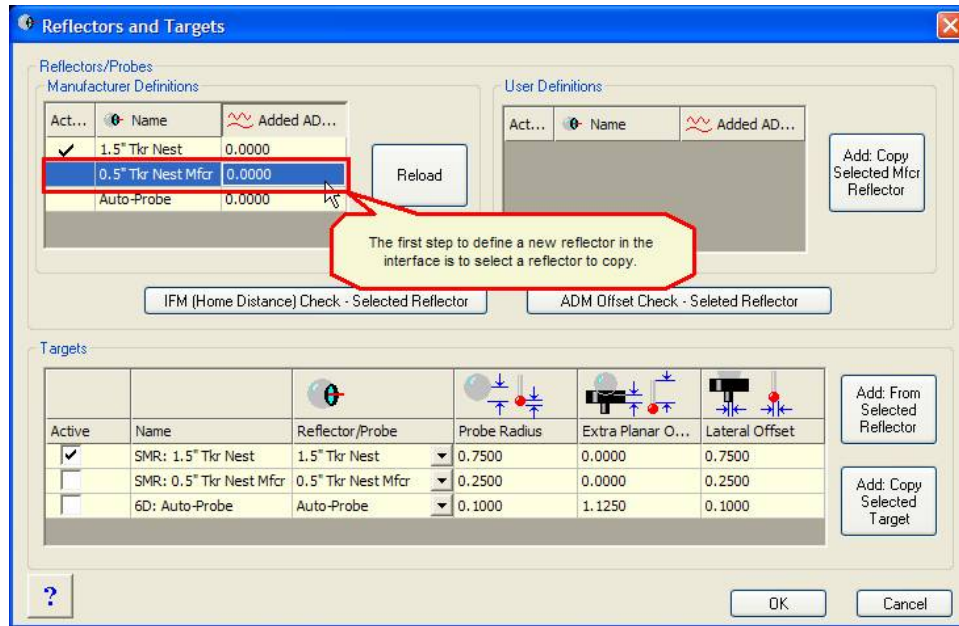
There are two processes to add a new target into the interface. The fast method is to select a reflector in the Targets table and hit the Add: Copy Selected Target button.



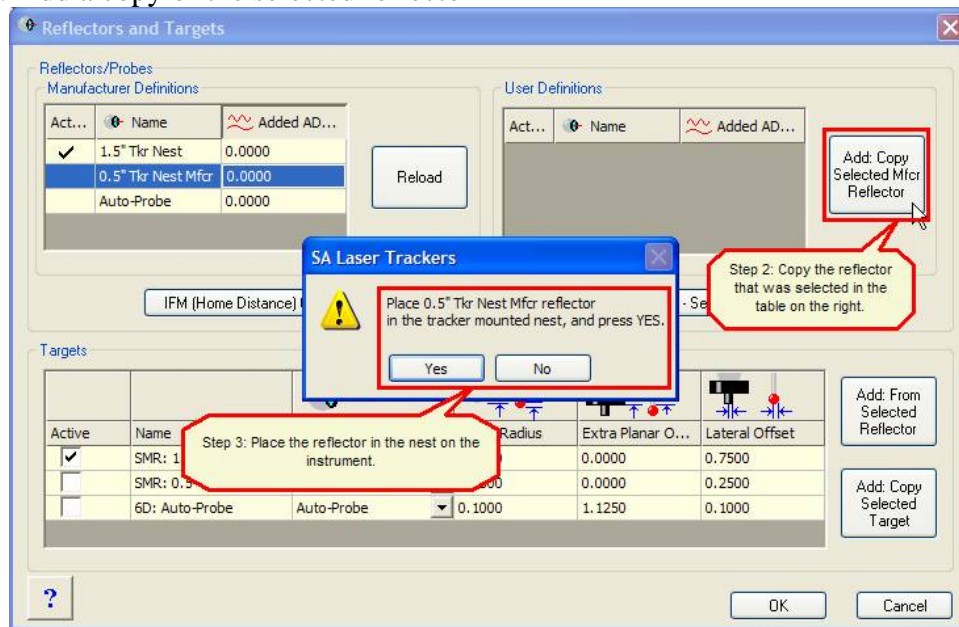
A new target is added to the list where you can set its mechanical offsets. This process doesn't allow a new reflector type. The procedure outline below is used to add new reflector types.

To add a new Reflector, as in a solid 0.5SMR solid(Prism Type) please follow the following process steps below. This procedure has been added to manual. A new release of the manual will be out in the near term.

Step 1: Select the reflector to copy...



Step 2: Add a copy of the selected reflector

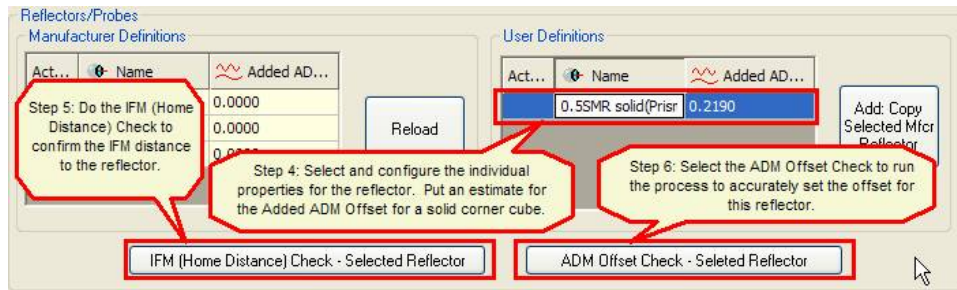


Step 3: Place the reflector in the tracker's nest and then hit the Yes button on the message box (see the figure above).

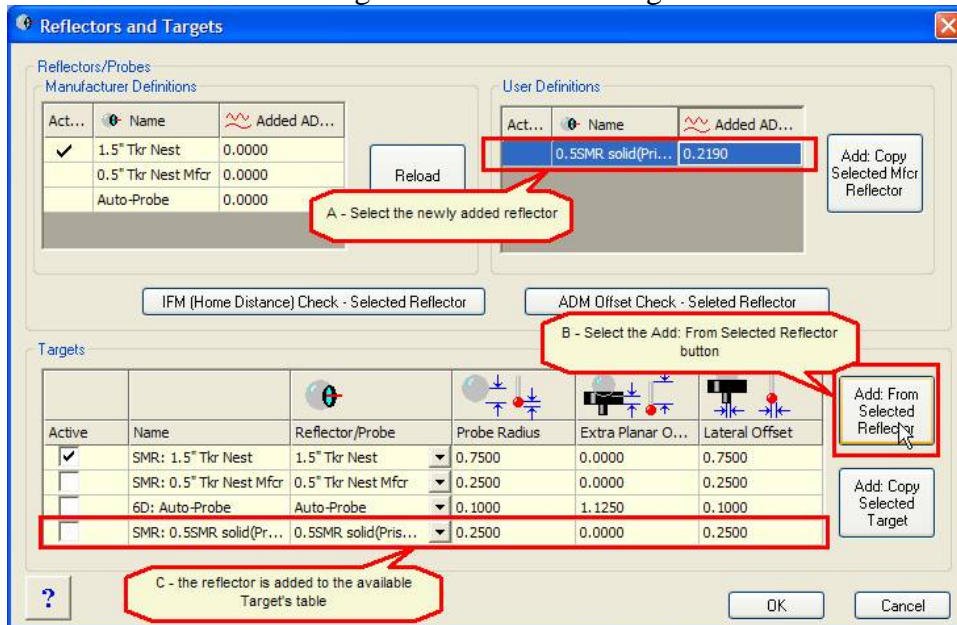
Step 4: Set the reflector properties for the newly added reflector.

Step 5: Run the IFM (Home Distance) Check for the reflector to confirm its initial Interferometer distance. The procedure should be repeated until the change to the initial distance is less than 0.0005 inches [0.013 mm].

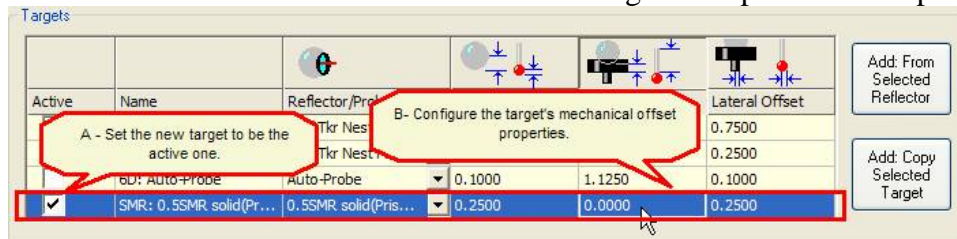
Step 6: Run the ADM Offset Check. A minimum of 4 IFM and ADM distances at different ranges are required to compute the ADM offset for a reflector. Please use distances that include minimum and maximum ranges used by the instrument.



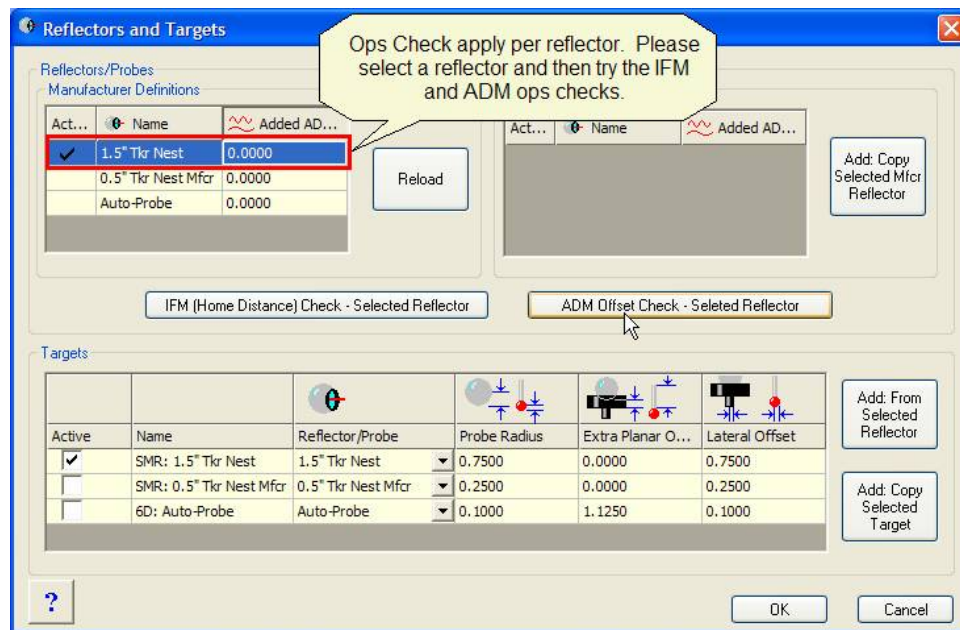
Step 7: To add the reflector into the list of targets. Select it and then hit the Add: From Selected Reflector button in the target section of the dialog.



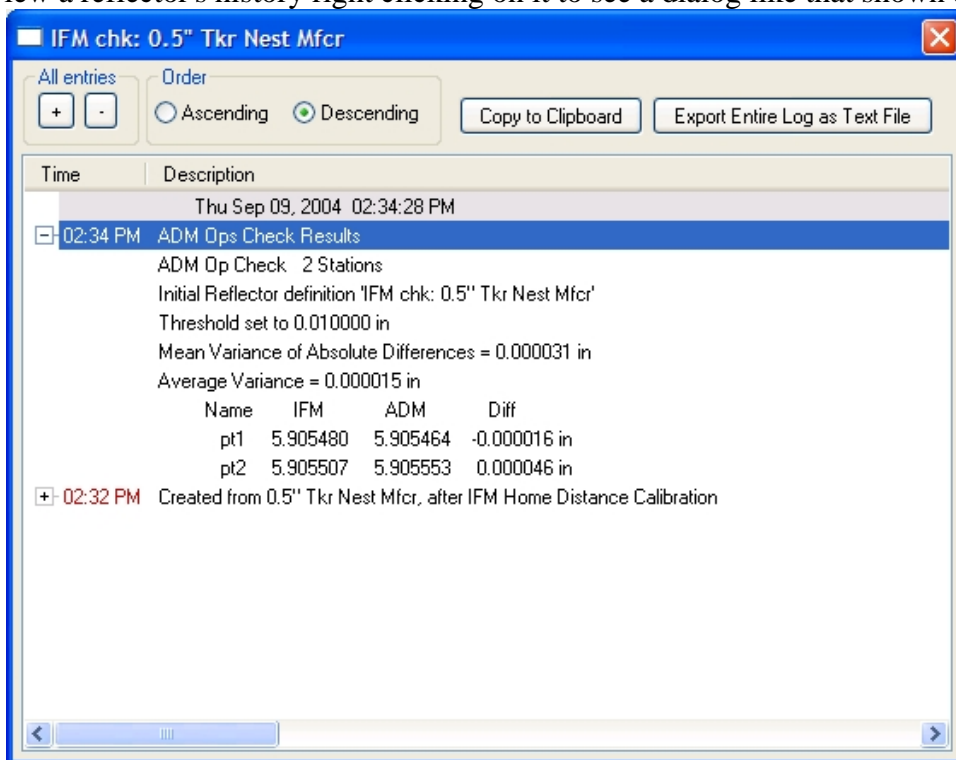
Step 8: Make the new reflector the active one and configure its specific offset properties.



Please remember that the Ops Check apply per reflector.



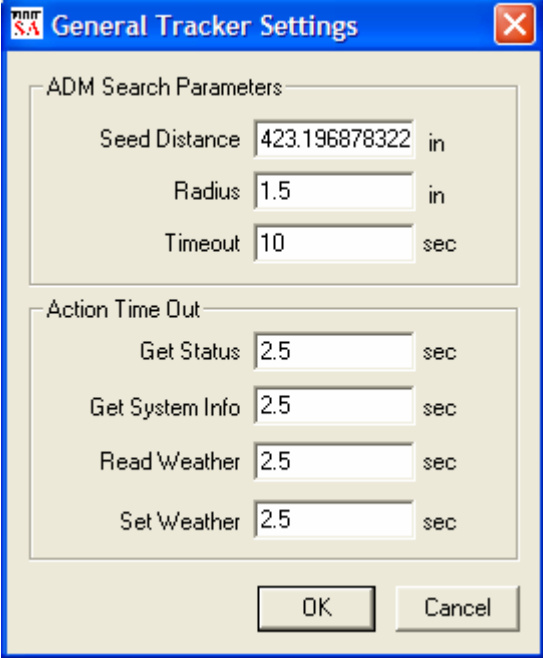
To review a reflector's history right clicking on it to see a dialog like that shown below.



Advanced Features

General Tracker Settings

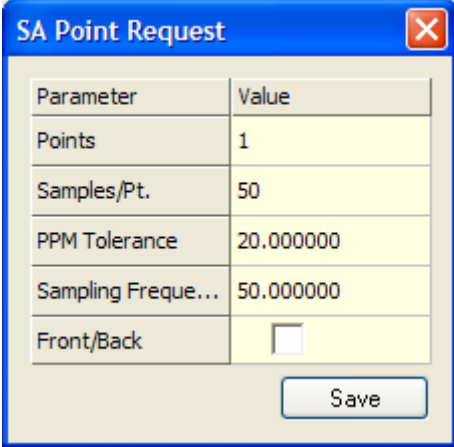
General Tracker Setting are access with menu Settings >> Tracker >> General Settings. The menu option opens a dialog that allows changes to Search Parameters and Action Time Out settings.



The dialog box titled "General Tracker Settings" contains two sections. The first section, "ADM Search Parameters", has three input fields: "Seed Distance" with the value 423.196878322 and unit "in", "Radius" with the value 1.5 and unit "in", and "Timeout" with the value 10 and unit "sec". The second section, "Action Time Out", has four input fields: "Get Status" with the value 2.5 and unit "sec", "Get System Info" with the value 2.5 and unit "sec", "Read Weather" with the value 2.5 and unit "sec", and "Set Weather" with the value 2.5 and unit "sec". At the bottom are "OK" and "Cancel" buttons.

SA Point Request

When SA requests a point measurement from the tracker interface a special measurement profiles is used. Configuring the parameters for this mode is accessed from the tracker's menu. Go to the Settings >> Tracker >> SA Interaction >> Measurement Requests >> Single Point Parameters. Configure the acquisition settings for these measurements on the dialog like that shown below.



The dialog box titled "SA Point Request" contains a table with two columns: "Parameter" and "Value". The table has five rows: "Points" with value 1, "Samples/Pt." with value 50, "PPM Tolerance" with value 20.000000, "Sampling Freque..." with value 50.000000, and "Front/Back" with an unchecked checkbox. A "Save" button is located at the bottom right.

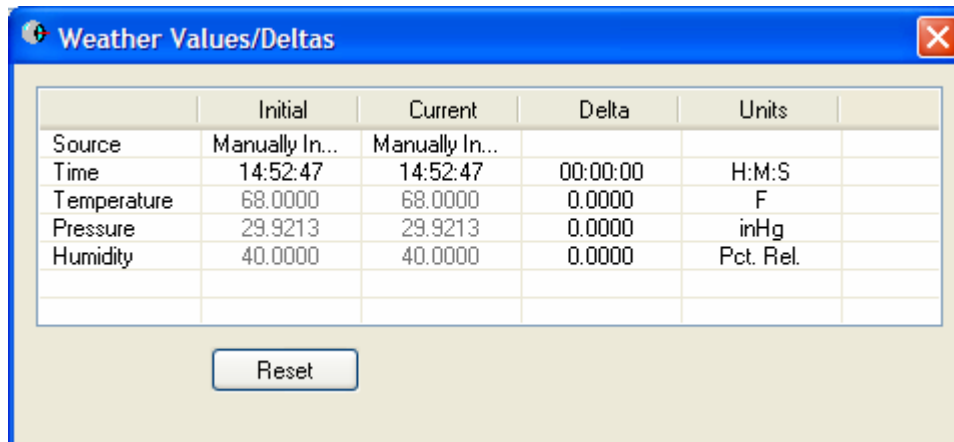
Parameter	Value
Points	1
Samples/Pt.	50
PPM Tolerance	20.000000
Sampling Freque...	50.000000
Front/Back	<input type="checkbox"/>

This mode only sends points to SA when prompted by SA.

Environmental Monitoring

A weather station is used to compensate a tracker range measurements. This is important for accurate range measurements. The current weather data is used to get an accurate wavelength for the laser light being used to measure the range to the target. The values are recorded in the SA log file.

The weather station components are acquired directly from special hardware or manually entered. The Values Currently Set In Tracker show the last weather parameters which were applied and used to recalculate the refractive indices.

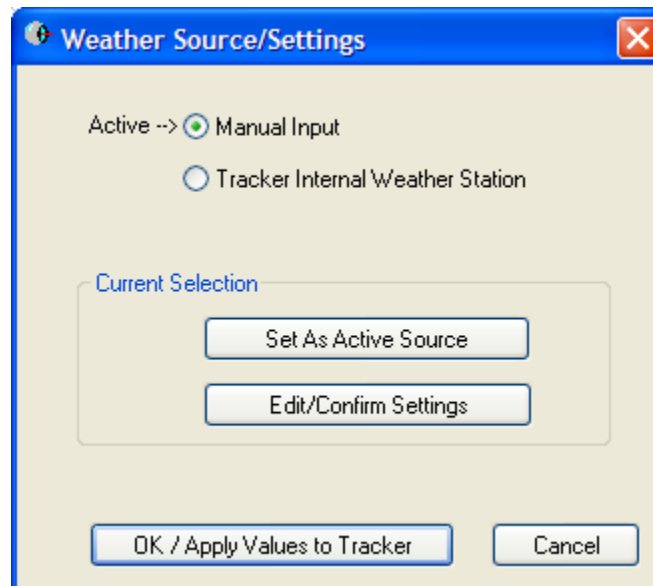


The Weather Values/Deltas dialog box displays a table with weather data and a Reset button.

	Initial	Current	Delta	Units	
Source	Manually In...	Manually In...			
Time	14:52:47	14:52:47	00:00:00	H:M:S	
Temperature	68.0000	68.0000	0.0000	F	
Pressure	29.9213	29.9213	0.0000	inHg	
Humidity	40.0000	40.0000	0.0000	Pct. Rel.	

Reset

Weather Source/Setting dialog is used to control how often new weather data acquired from the weather station. If Manual Input is selected as the input method, the Hardware Settings area will become disabled. For some trackers, the Hardware Settings area will also contain controls for interfacing to external weather station devices.



The Weather Source/Settings dialog box allows users to select the weather source and manage settings.

Active --> ☒ Manual Input
☐ Tracker Internal Weather Station

Current Selection

Set As Active Source

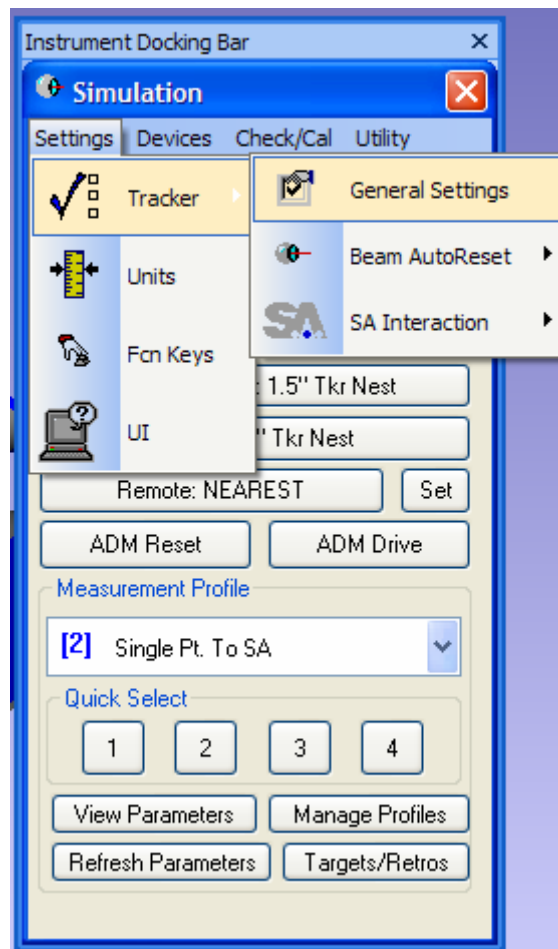
Edit/Confirm Settings

OK / Apply Values to Tracker Cancel

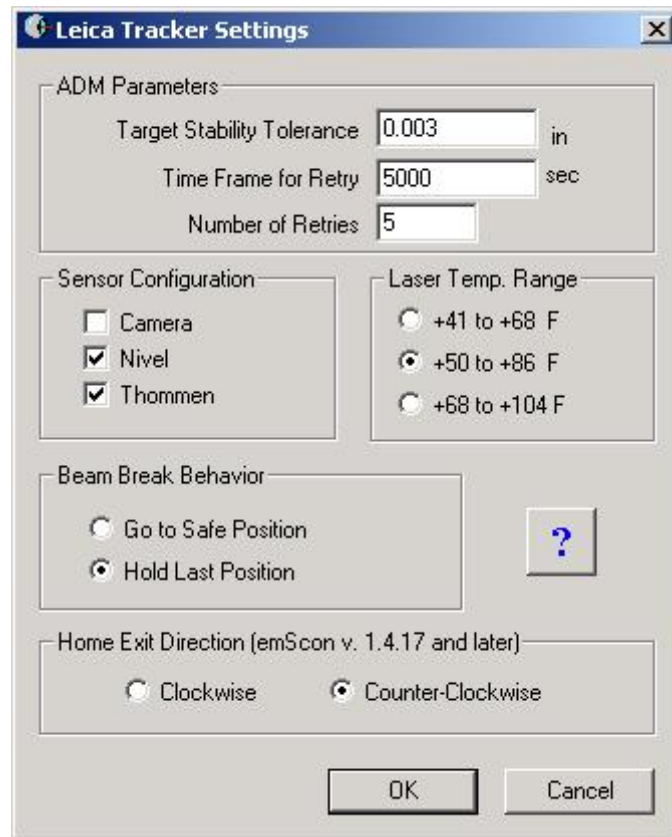
Gravity Measurement

Enable the Gravity Measurement device for the instrument (e.g., Nivel for Leica, Integrated Level sensor for FARO and API).

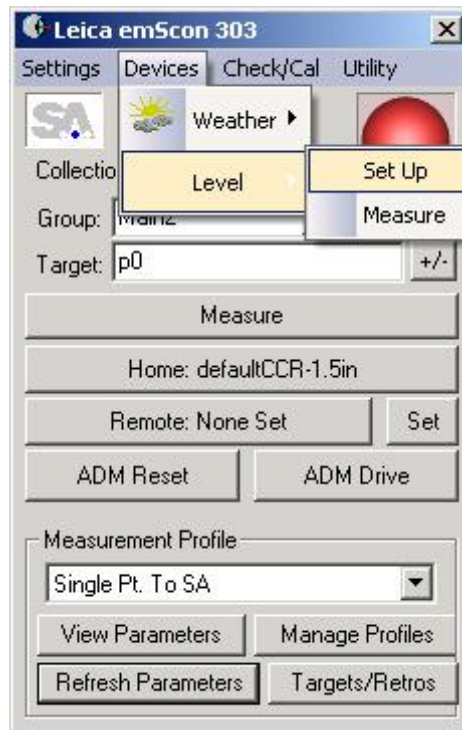
In the main interface dialog's menu, go to Settings->Tracker->General Settings



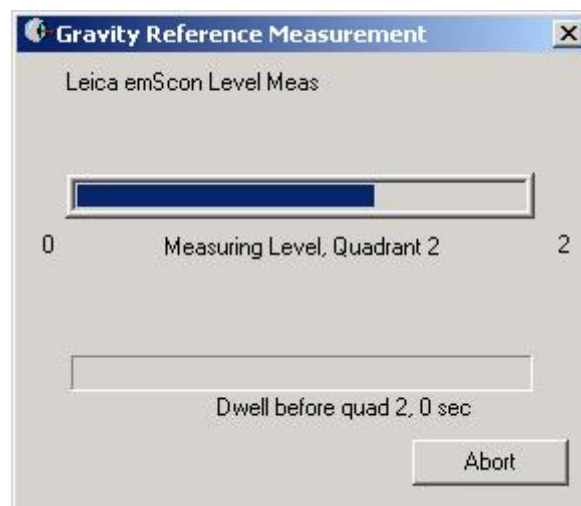
From the General Settings dialog, press the Instrument specific configuration (e.g., emScon) button for the tracker specific settings



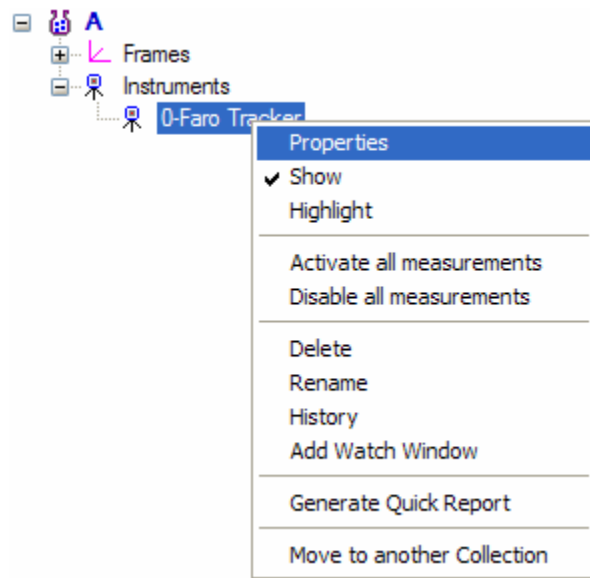
Note: the (e.g., Nivel) Sensor is checked. Also note the “?” button – whenever this appears, additional information is available. Press OK to enable the sensor (e.g., Nivel). Now go to the ‘Devices’ menu back in the main dialog. Select Level->SetUp to edit samples per quadrant, dwell time, etc.



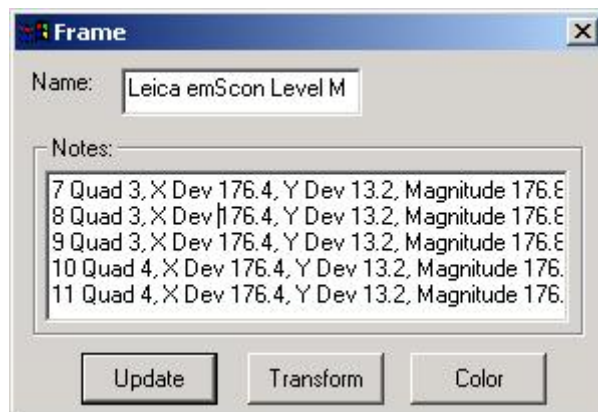
Select Measure in the same menu to begin the gravity measurement.



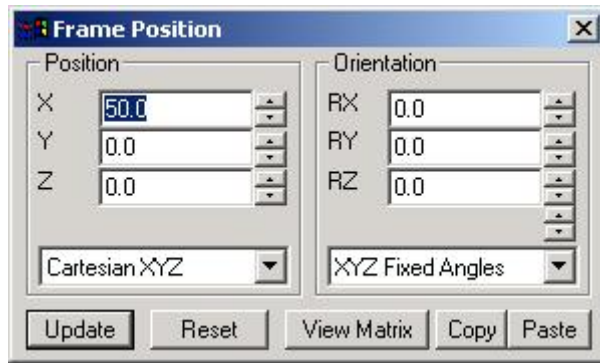
When the measurement is done, the frame will be sent to SA bearing the name of the source. Right click the new frame, and select 'Properties'



The properties dialog provides a detailed look at the samples taken. Units are seconds of arc.

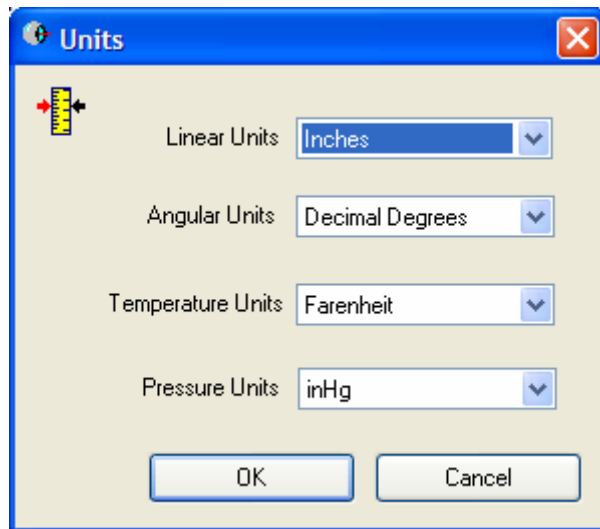



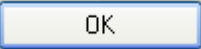
For this example, the Nivel was not actually mounted on the tracker, but instead just placed on a level platform. Therefore, when 'Transform', is selected a 'perfect' level frame is shown, since all the quadrant measurements exactly cancel out.



Tracker Interface Units

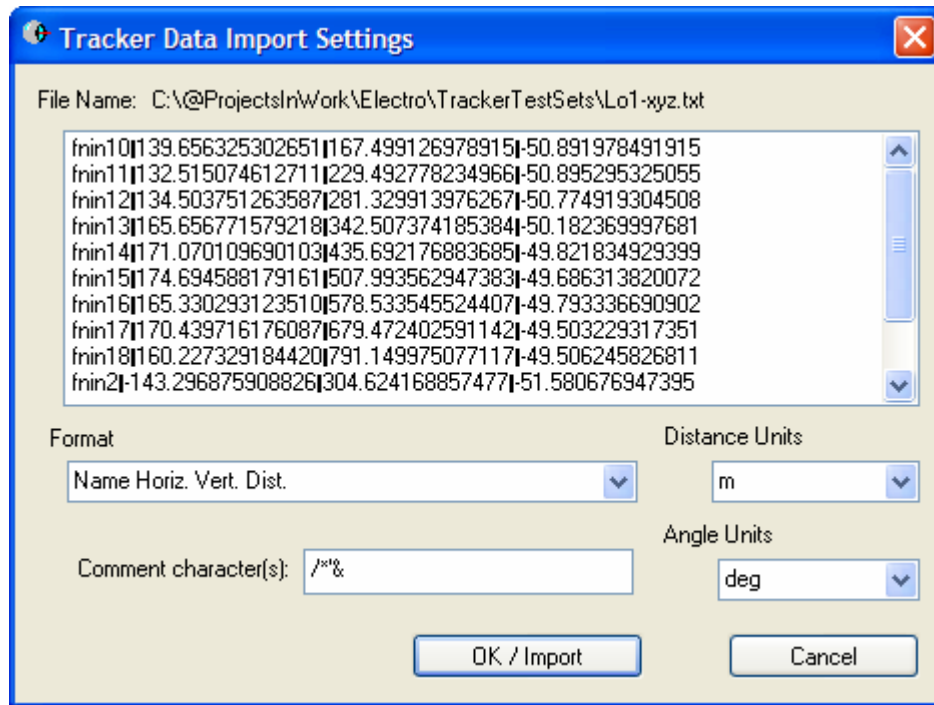
The Units tab provides a way to set the displayed units for the tracker interface to your preference. Left click on the Units tab, and the Settings Dialog will look like:



Just press  in any of the Units drop lists to select the display units of your choice. As always, press  to save the new settings.

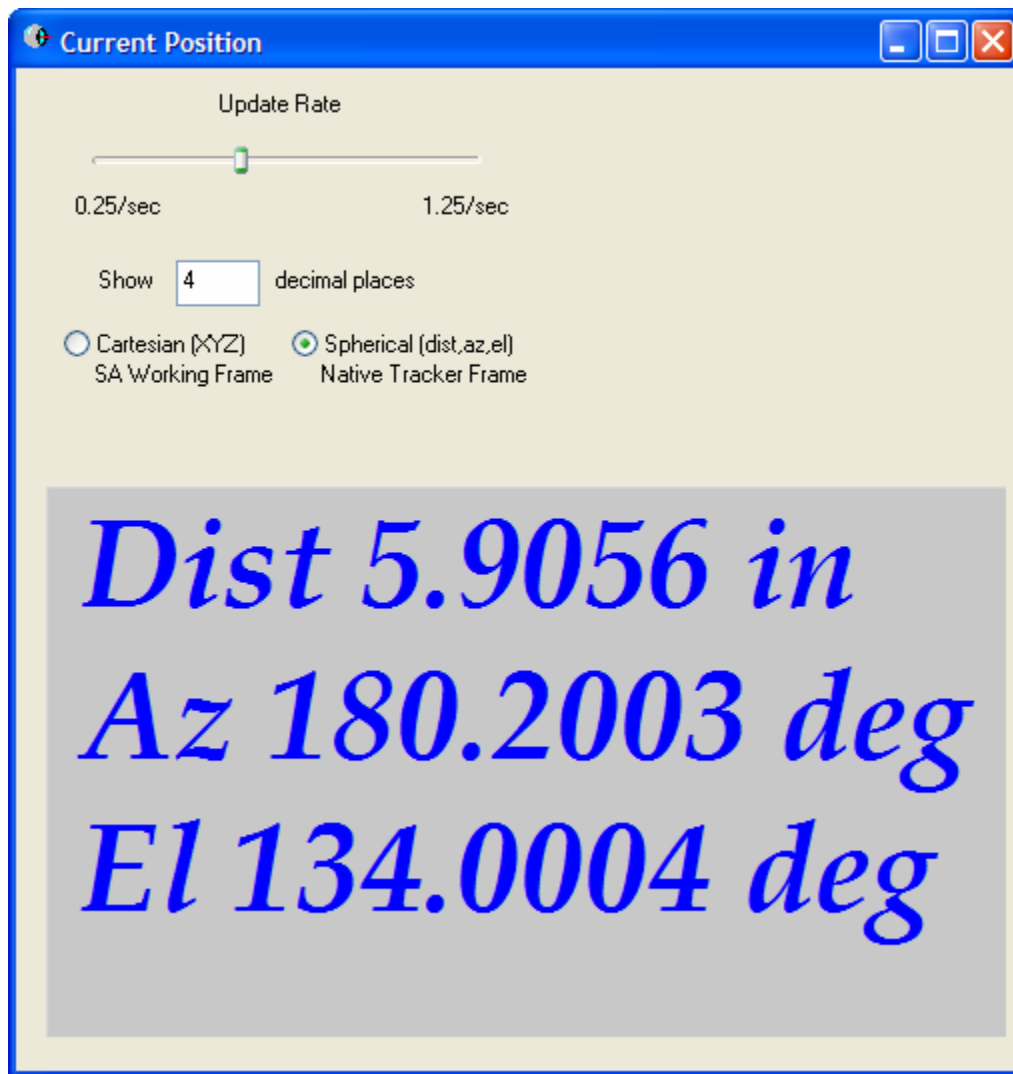
Importing Tracker Data


The tracker interface supports importing tracker data (e.g., point name, horizontal, vertical, and distance) directly from ASCII files. Use the tracker menu Utilities >> ASCII Import option to open files of raw tracker measurements. The data will be imported into SA and each point set to SA with all of its measurements. Two-Face measurements are also supported when importing the data. An example import is shown below.



Viewing Current Position

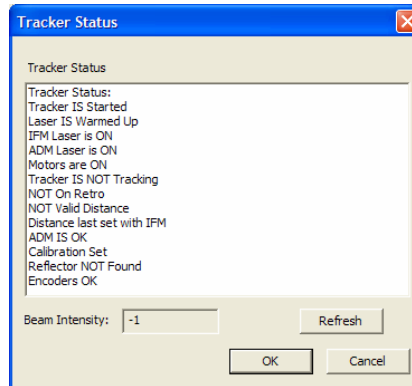
The current position of the retro-reflector can be observed at any time. To view the current position, go to the tracker menu and select Utilities >> Live Data.. The Current Position dialog will appear:



The Current Position dialog can be resized for viewing at distances. To close the Current Position dialog, press  in the dialog.

Tracker Status

The tracker status can be checked at any time. The status report is different for specific trackers, but generally includes information pertaining to the laser tube, the lock state of the laser beam on the retro-reflector, and some information about the tracker controller. To check the status, select the menu Utilities >> Tkr Status option. The Status report dialog will appear:



Importing Polar Data into SA

The following sections describe the process using an Axyz example dataset.

Condition the data for import

Using the example of the accelerator ring data, and starting with the Excel file...

Workpiece	PointID	Station	Horizontal (deg)	Vertical (deg)	Distance (M)
DEFAULT	123PSA	1	72.3590	92.4279	5.572243
DEFAULT	123PSC	1	73.4544	92.3426	5.813305
DEFAULT	123PSD	1	74.1478	92.4135	5.640094
DEFAULT	BTSB2B	1	-106.2040	91.5356	13.010014
DEFAULT	BTSB2C	1	-103.7447	91.6882	11.913730

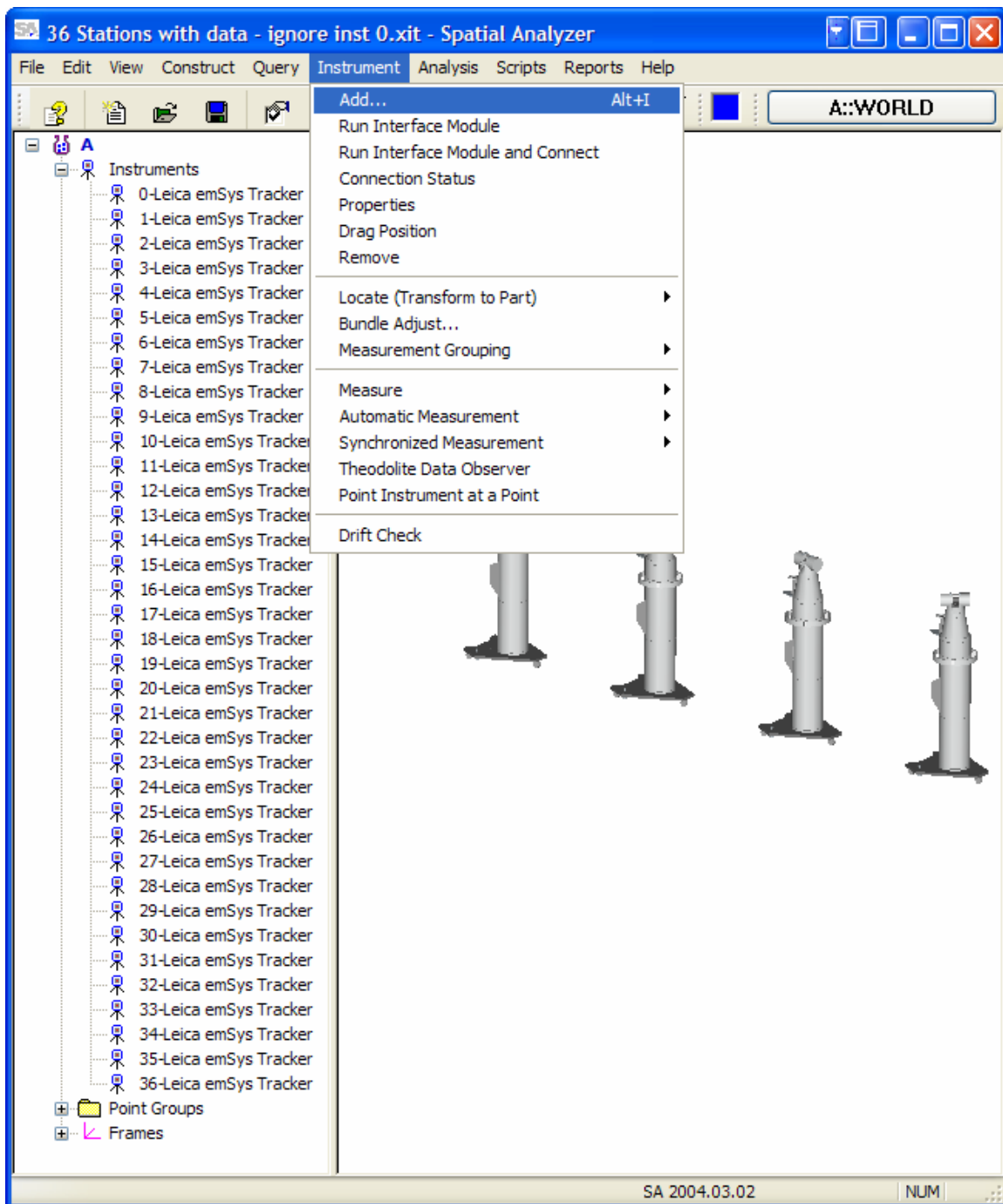
Remove the workpiece and station columns, and separated each station's data into its own excel worksheet. E.g., worksheet1 looked like...

123PSA	72.3590	92.4279	5.572243
123PSC	73.4544	92.3426	5.813305
123PSD	74.1478	92.4135	5.640094
BTSB2B	-106.2040	91.5356	13.010014
BTSB2C	-103.7447	91.6882	11.913730

Save each worksheet as tab delimited ASCII text (.txt), named Station1.txt, ... Station36.txt.

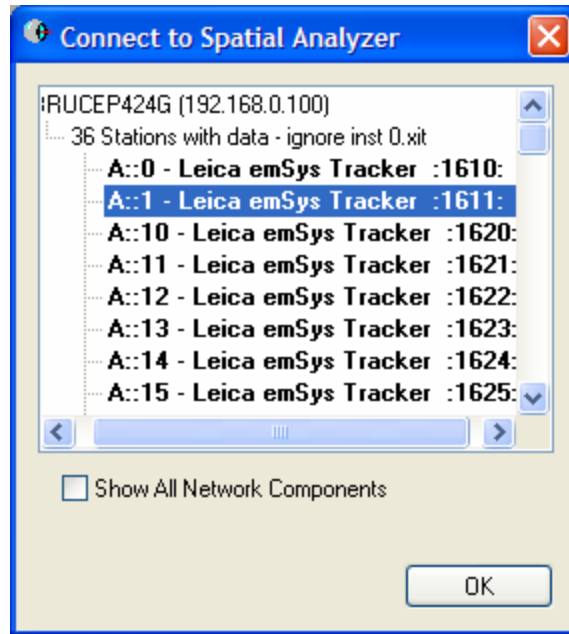
2) Add stations to SA

Start SA, and add an instrument for each station, e.g., 37 instruments – the data set names start at '1', instrument id's start at '0'. To avoid confusion consider leaving instrument 0 as a placeholder.

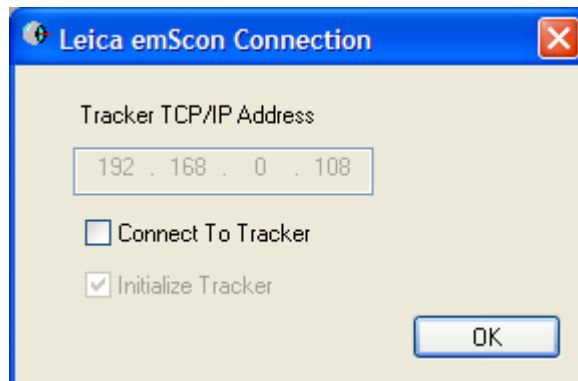


3) Connect the interface (included SA Laser Trackers.exe) to a station

Run SA Laser Trackers.exe, The SA connection dialog is shown below:

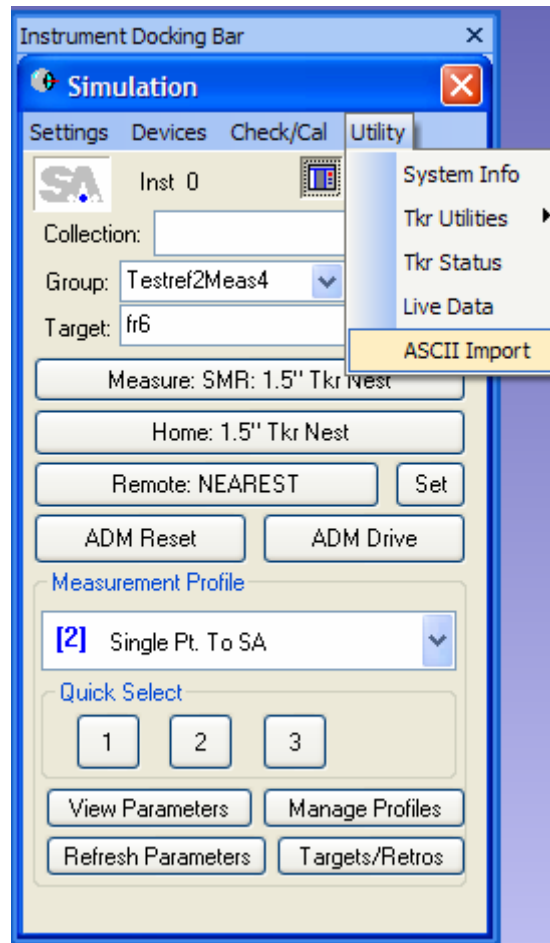


Select Instrument 1, and hit OK. Then the hardware connection dialog is shown:

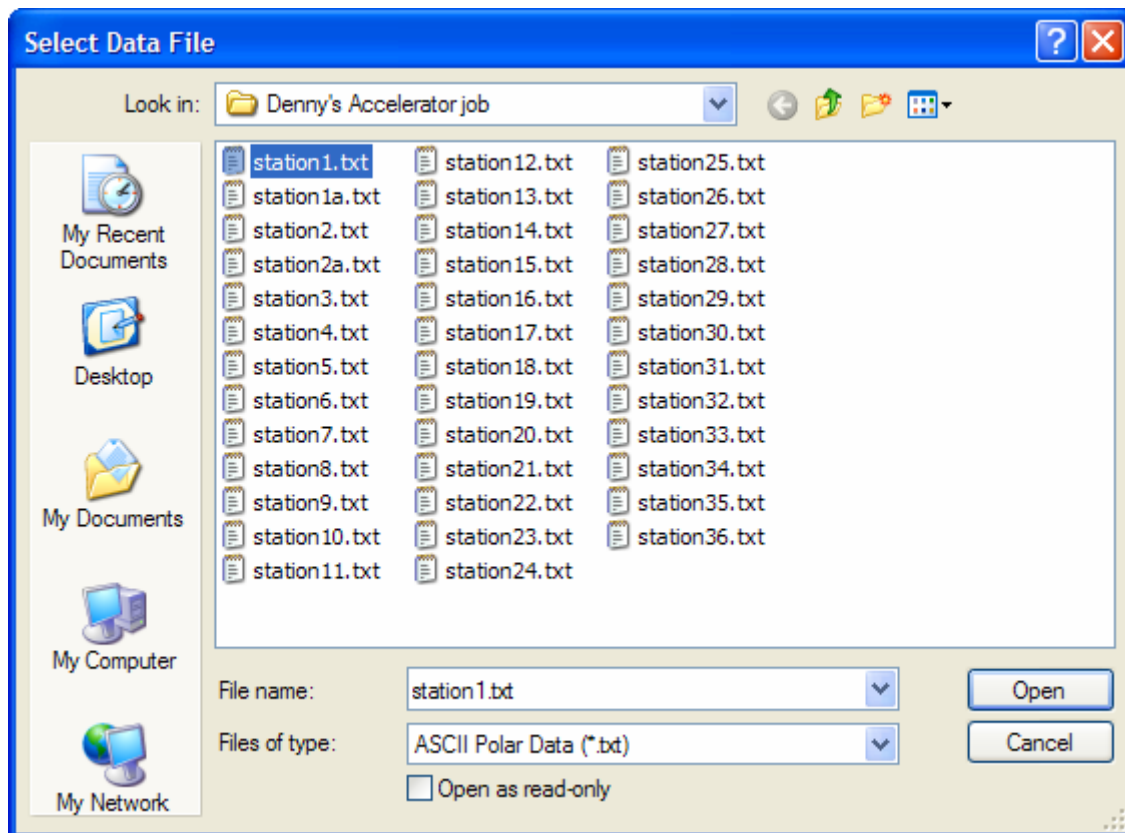


Un-check Connect To Tracker as shown, and hit OK. The interface will 'fake' starting a tracker, and pop up in Simulation mode...

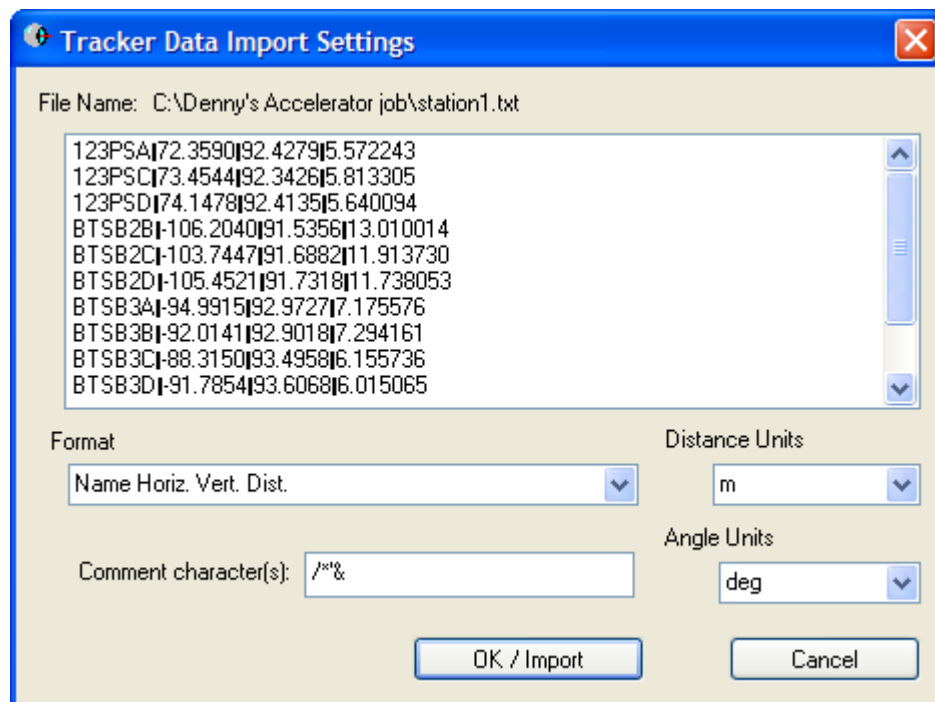
4) Import the data



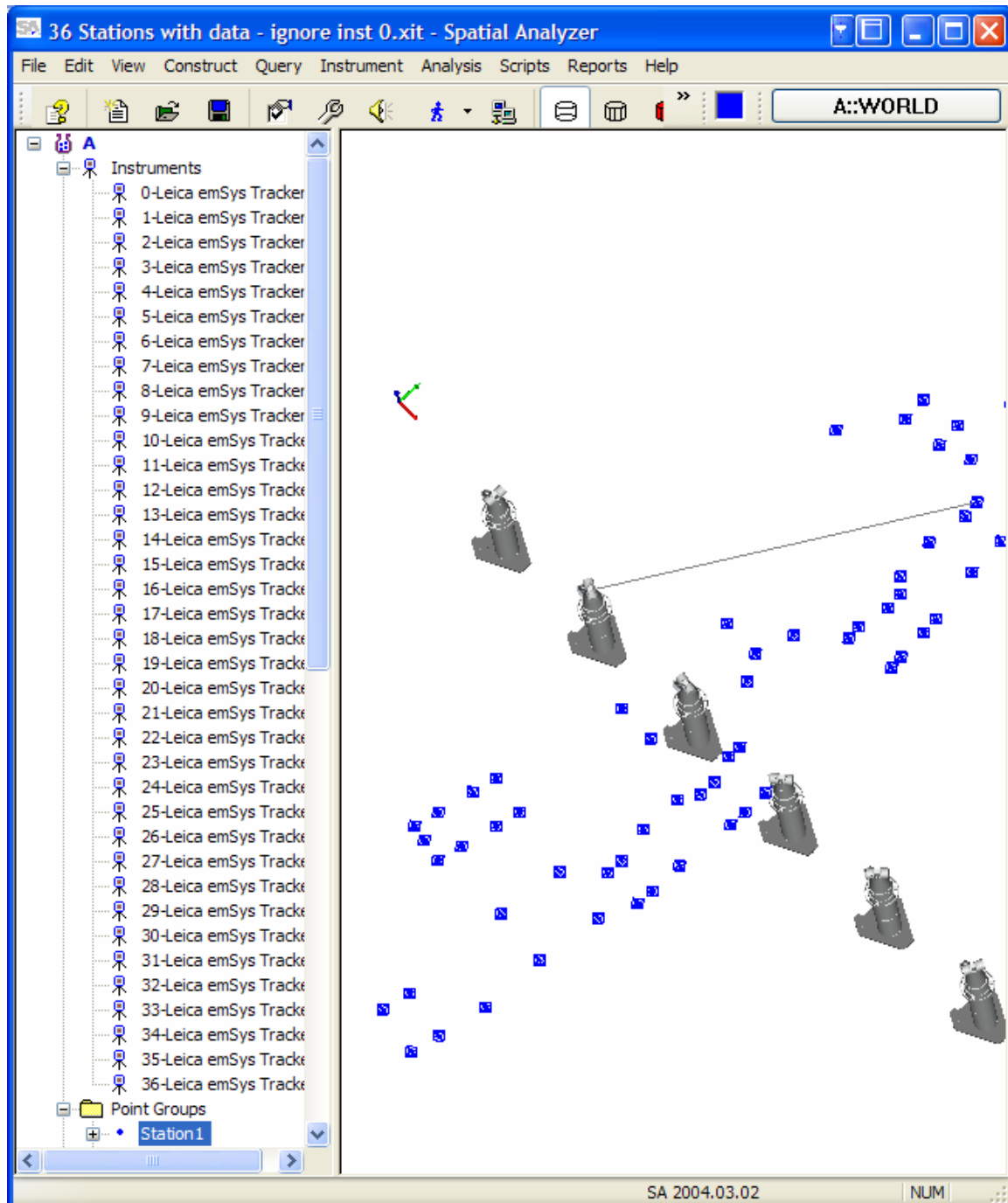
Note the 'Inst 1' next to the SA stamp – this confirms the connection to Instrument 1 in SA. Set a unique group name, “Sation1” seemed appropriate here. From the Utility menu, select ASCII Import. You’ll get the file selection dialog:



Browse to the desired file (station1.txt in this case) and hit Open. The import settings dialog is shown:



The settings just happen to default to your data format, so just hit OK/Import. The data will appear in SA in the group you specified above, as though it had been measured by the specified instrument:



5) Repeat steps 3 and 4 for all station/data set combinations

NOTE: You must run a new instance of the tracker interface to start step 3 and connect too the next station in SA. Closing the existing tracker interface before adding the next station is recommended too minimize the number of open interface instances.


MP OpCheck string commands for trackers...

Command	Description
"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 []
"Motors On"	Turn motors on
"Motors Off"	Turn motors off
"Start"	Re-start tracker
"Home"	Home tracker
"Two Face Here"	Perform Two Face ops check at current location
"ADM/IFM Here"	Perform ADM ops check (w.r.t. IFM) at current location
"IFM"	Start IFM ops check/cal. (two station or scale bar) at current location
"Closure"	Perform Closure check at current location w.r.t. current home
"Drift"	Start Drift ops check
"ADM"	Start ADM ops check/cal. ui for multiple locations (w.r.t. IFM)
"ADM Drive"	Equal to pressing the 'ADM Drive' button
"ADM Reset"	Equal to pressing the 'ADM Reset' button
"Ball Bar"	Start Ball Bar ops check
"Valid Distance"	Success if valid adm or ifm distance, fail if not

Leica NETBEUI (TP-Link) Setup Process

This document details the guidelines to follow when using Spatial Analyser software on Leica TP-Link Trackers. TP Link Trackers generally are numbered SMART310, LT500 and LTD500. Later model Trackers such as the 600, 700 and 800 series generally connect through emScon

TP-Link → Setting-Up the PC

These instructions are also on the tracker interface dialog in SA. Select the help button on the interface.  in the Laser Tracker Interface. “Instrument > Run Interface Module > Laser Trackers”

1. Ensure NetBEUI is installed and ready on the PC (See instructions below, skip this step if NetBEUI is already installed).
2. To start the tracker, you will need the parameter file, Trk[serial num].mdb, and the pgm file, LT-Axyz.pgm.
3. You can get v. 2.89 of the pgm from <ftp://ftp.kinematics.com/pub/SA/Misc/Leica TP-Link PGM 2.89>
4. By default, both the mdb and pgm files are found in your Axyz install folder, e.g. C:\Program Files\Leica\Axyz\Tracker\

Preparing an XP PC to run Leica TP-Link Tracker

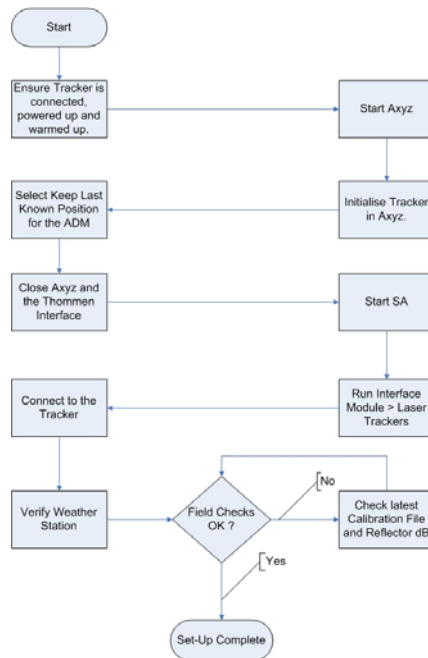
1. Install the NetBEUI protocol following the instructions below. Instructions copied from Microsoft web site... <http://support.microsoft.com/default.aspx?scid=kb;en-us;301041&sd=tech>
2. Use the LANASettingWin2000.exe app from LeicaTP-LinkNetBiosLanaUtility.zip on <ftp://ftp.kinematics.com/pub/SA/Misc/> to set the LANA for the NETBEUI card to 0. That is currently required for SA to communicate to Leica LTD-500/300 trackers.
3. The files that you need for installing the NetBEUI protocol on Windows XP are Netnbf.inf and Nbf.sys. To install NetBEUI, follow these steps:
4. Insert your Windows XP CD-ROM in the CD-ROM drive and then locate the Valueadd\MSFT\Net\NetBEUI folder.
5. Copy the Nbf.sys file to the %SYSTEMROOT%\System32\Drivers folder
6. Copy the Netnbf.inf file to the %SYSTEMROOT%\Inf hidden folder.
7. Note To make a hidden folder viewable, follow these steps:
8. Click Start, click Run, type Explorer, and then click OK
9. Click Tools, click Folder Options, and then click the View tab
10. Under Advanced Settings, click to select Show hidden files and folders under the Hidden files and folders Folder.
11. Click Start, click Control Panel, and then double-click Network Connections
12. Right-click the adapter you want to add NetBEUI to, and then click Properties
13. On the General tab, click Install
14. Click Protocol, and then click Add
15. Click to select NetBEUI Protocol from the list and then click OK
16. Restart your computer if you are prompted to do this
17. The NetBEUI protocol is now installed and working

Connecting to the Tracker

There are two ways to connect a Leica TP-Link Tracker to SA. Connection just through SA or connection in Axyz first, this retains the ADM beam break behaviour. A process flow chart is shown below.

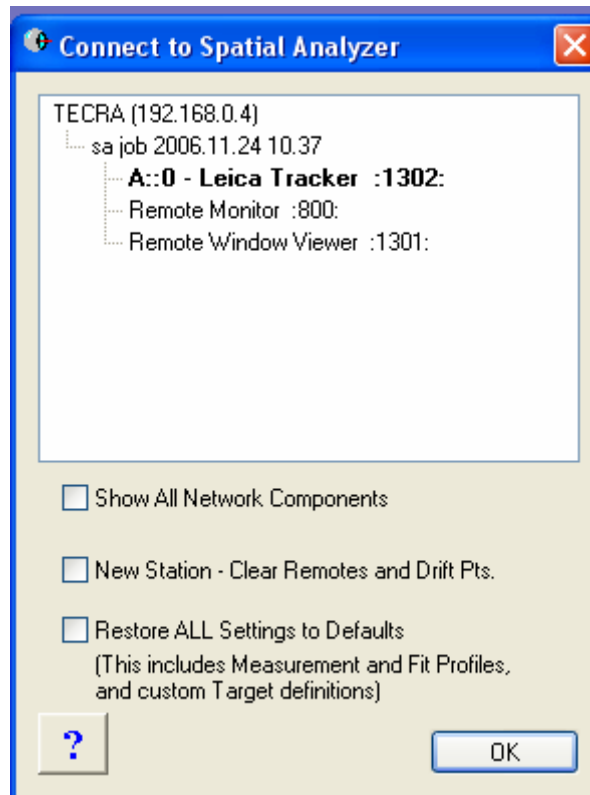
Use Spatial Analyzer with Leica TP-Link Trackers

TS101 – 2 Initialise in Axyz

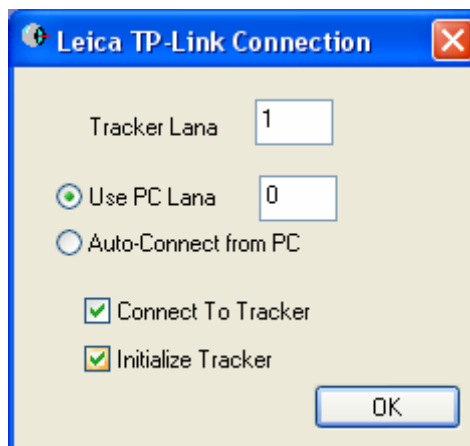


Initializing the tracker from SA

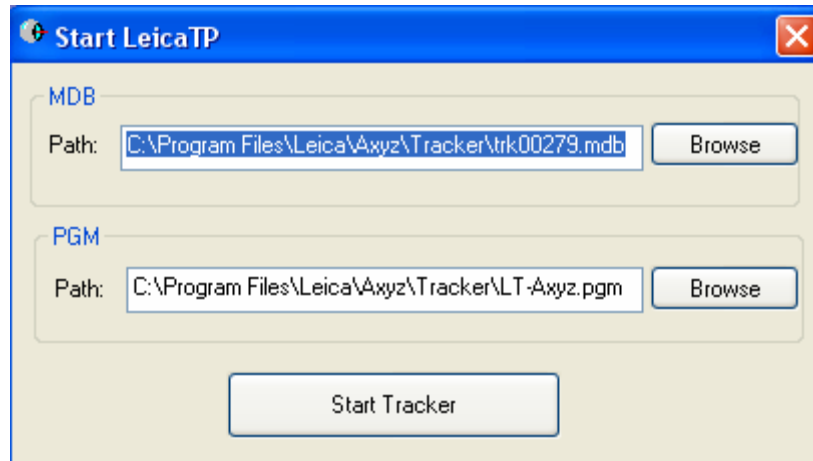
- Connect the Tracker, power up and allow to warm-up.
- Start SA
- Select “Instrument > Add > Leica Tracker” and Add instrument to the job.
- Do Not Select Leica emSys Tracker
- Select “Instrument > Run Interface Module > Laser Trackers” and highlight the Tracker you want to connect to – in bold.



- Press OK
- On the next window Check the “Initialise Tracker and Connect to Tracker” boxes and press OK.



- Browse to the Correct Laser Tracker Calibration file (mdb file) and the Leica .pgm file and then select “Start Tracker”
- Note : The latest file should be in the folder C:\ProgramFiles\Leica\Axyz\Tracker\trkXXXXX.mdb. If there are multiple .mdb files on the PC check with your Leica representative as to which is the latest file.

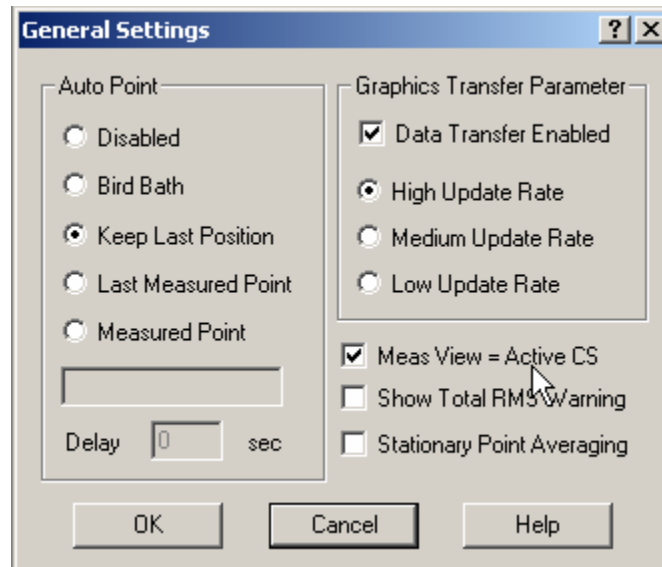


- Note - The above connection can sometimes be a little temperamental. Just reselect the .mdb and .pgm file if the connection fails. It does work - but it may take a couple of tries to connect.
- Note – Do not use version 2.86 of the Leica .pgm file. Use version 2.89 – available from the NRK web site. www.kinematics.com
- Check the Weather Station Connection as defined in Section 4.0
- You will now be connected to your Tracker in SA.

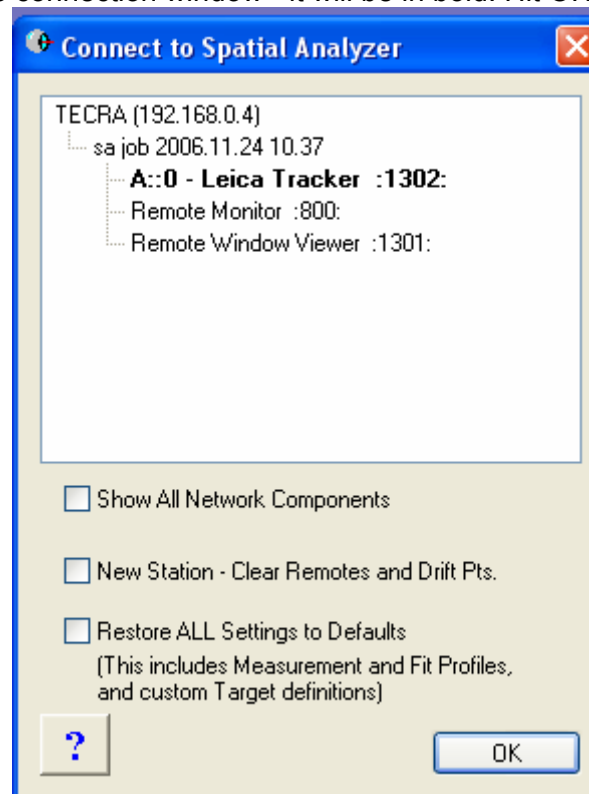
Initializing in Axyz before starting SA

Unfortunately, although SA can initialise the Tracker, ADM beam break behaviour can only be set through Axyz. Hence, if you wish to use the ADM in this manner, you will have to initialise the Tracker in Axyz first as described below.

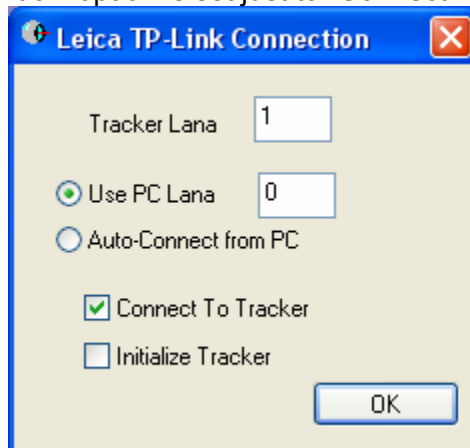
- Connect the Tracker, power up and allow to warm-up.
- Start Axyz.
- Initialise the Tracker and under “Tracker > Settings” set the beam break behaviour to “Automatically Keep Last Point”. Click onto something else and back onto “Automatically Keep Last Point” just to ensure that this is recognised in the software.



- Close Axyz
- Close the Weather Station Software interface – if this started automatically.
- Start SA
- From the menu select “Instrument > Add > Leica Tracker” to the job file – if one is not already there with the default file.
 - ***Do Not Select Leica emSys Tracker***
- Select Instrument > Run Interface Module > Laser Trackers and highlight your Tracker in the connection window - it will be in bold. Hit OK.



- Ensure that the window option is set just to “Connect To Tracker” and hit OK.

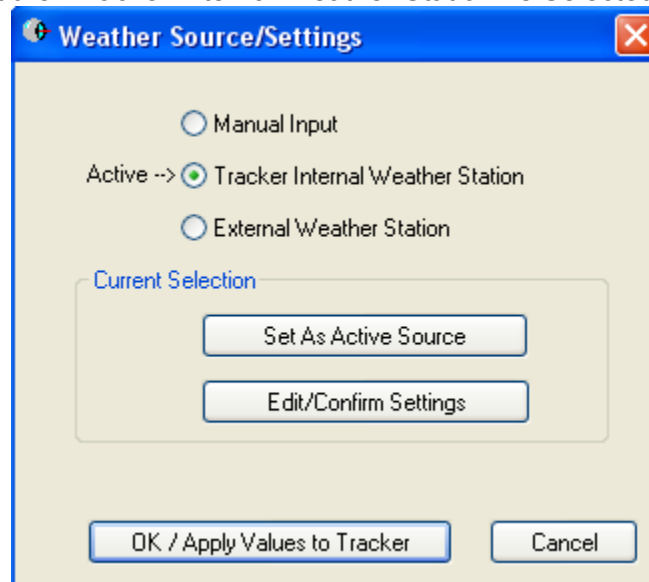


The Tracker interface will open and you will be connected to the Tracker and have correct ADM beam break behaviour.

Connection of The Weather Station

It is important to check that the Weather Station is enabled in the instrument interface when running Leica Trackers with the TP link as described below.

- Check that the Weather Station is plugged into the serial port on the PC and is switched on before starting SA.
- Select “Devices > Weather > Current Values/Deltas” and check that the Weather Station is being read.
- If the Weather Station is not displayed, Select “Devices > Weather > Set-up” and ensure that the “Tracker Internal Weather Station” is Selected

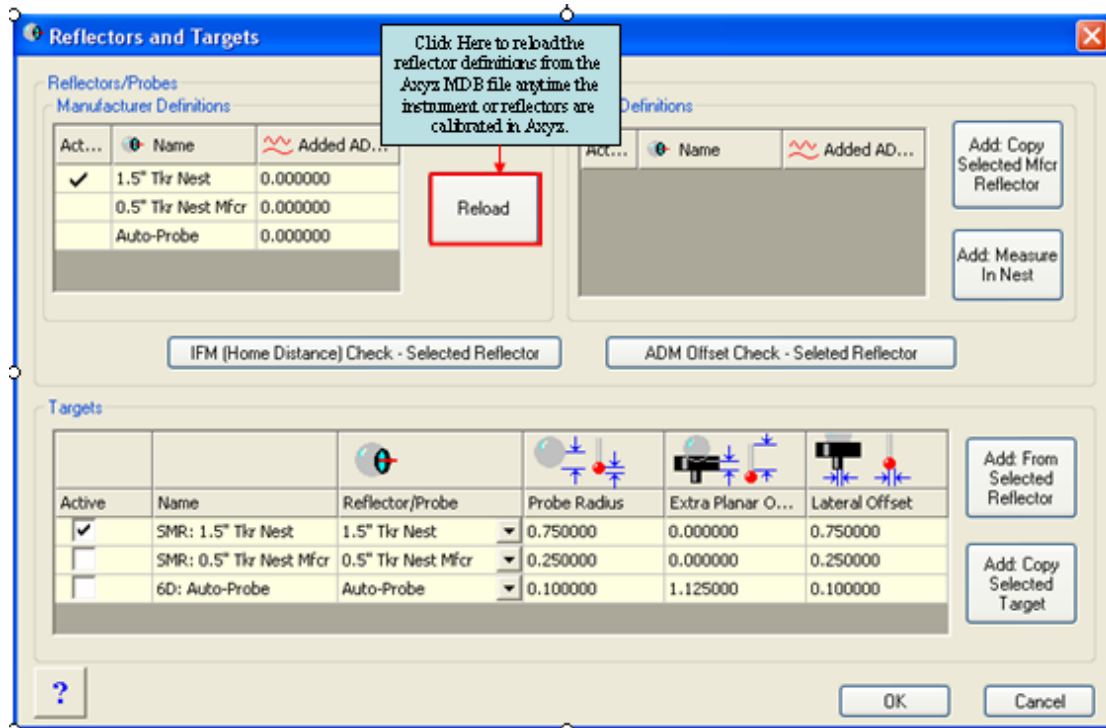


- Press “Set As Active Source” and then “OK/Apply Values to Tracker”
- Check the values again by selecting “Devices > Weather > Current Values/Deltas”

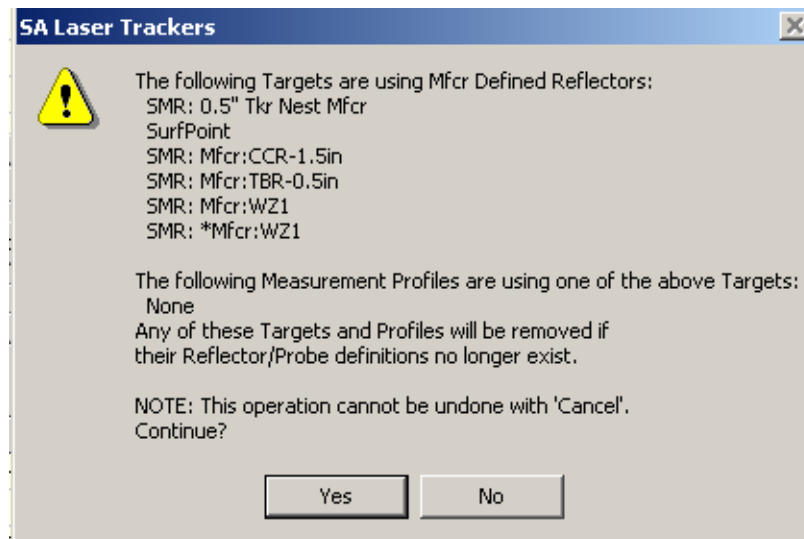
Tracker Calibration

Calibration of a TP-Link connected Tracker is performed in Axyz.

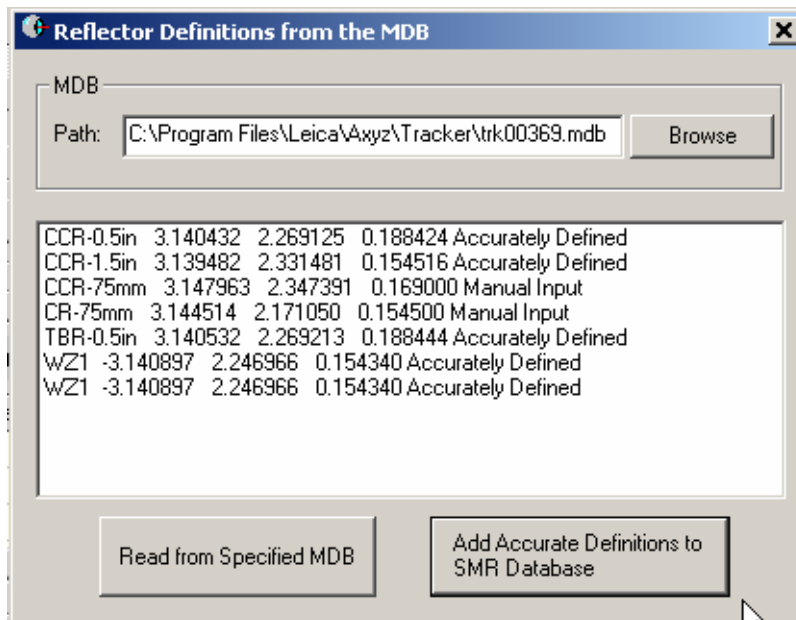
It is important that after Tracker or reflector calibration the reflector definitions are reloaded from the MDB file by selecting the option under “Targets/Retros” from the User Interface as shown below.



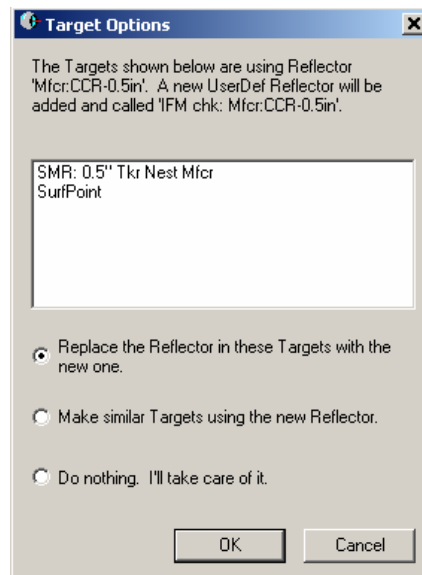
A window warns that this op cannot be undone – select “Yes”



Select “Read from Specified MDB” if the reflector definitions are not visible in the window and then “Add Accurate Definitions to SMR Database”



Select “Replace the Reflector in these Targets with the new one” and press “OK”



PORTABLE CMM INTERFACE: FARO, ROMER, SANDIA NL



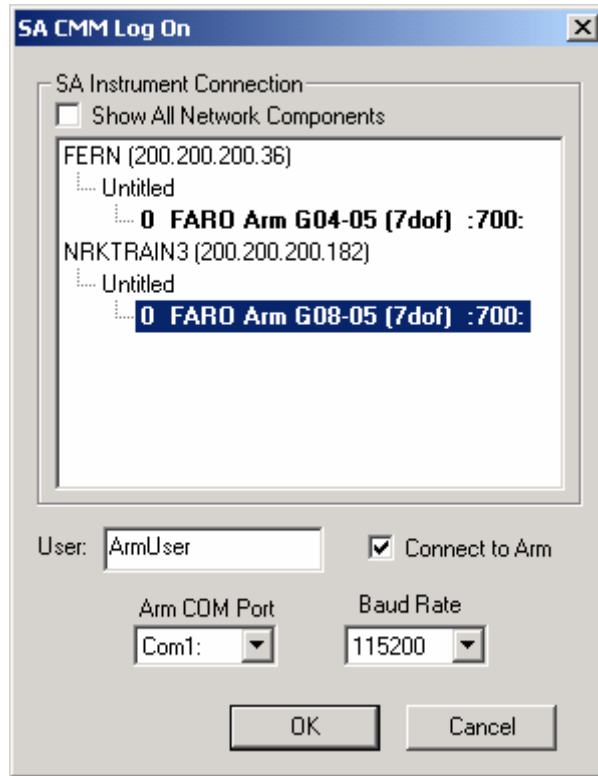
The SpatialAnalyzer Portable CMM (Coordinate Measuring Machine) Interface application is used to interface to one of three families of Portable CMM's (also called arms). The interface communicates to SpatialAnalyzer using New River Kinematics' TCP/IP socket protocol and communicates to the given CMM arm using the manufacturer's communication protocol. Faro.exe and Romer.exe use serial communication. The Sandia National Labs interface, SandiaArm.exe, uses an interface to a Galil motion control board. For the Romer and Faro arm interfaces, the buttons on the arms are used in combination with the graphical user interface to run the arms. Specific keyboard keys are used with the Sandia National Labs arm in place of arm buttons. The arms are quite different at the instrument level, but the interface applications look and feel identical to the user.


To run any of the arm interfaces, you must first install the arm specific configuration, or parameter file. This file is available from the given arm manufacturer, and in fact comes with the arm in all cases. Some manufacturers provide programs which load the files to the arm controllers.

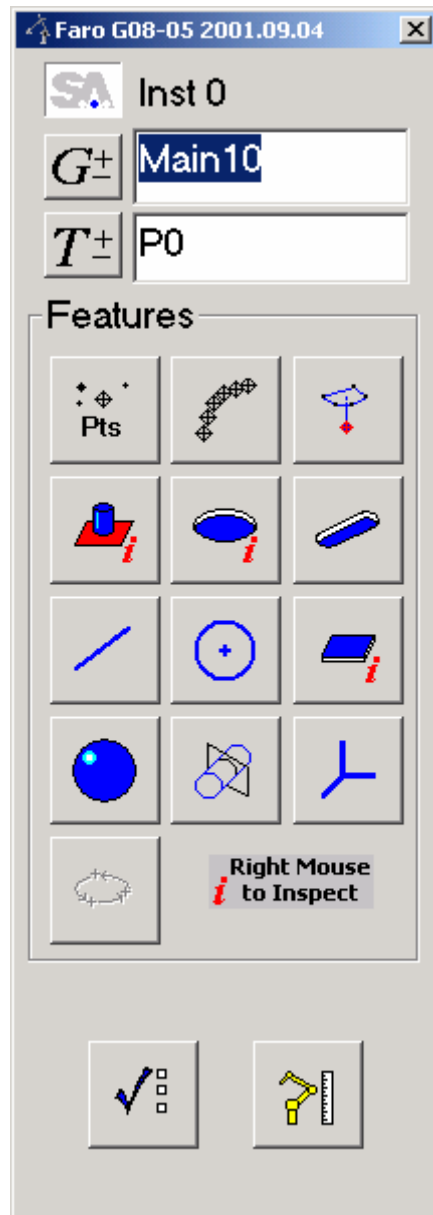
You must also have SpatialAnalyzer installed. The SA install includes all three arm applications, so only one install is required.

Starting the Interface

To run the CMM arm interface, as with any instrument interface in SpatialAnalyzer, the first step is to run SA and add the instrument model. This is more than just a model; it supplies the connection for the instrument, and serves as the real-time representation of the instrument in SA. Once the appropriate arm model has been added, run the arm interface. We'll use the Faro G08-05 as an example, so we'll run Faro.exe. The Spatial Analyzer CMM Log On Dialog appears with available SA connections shown:



In the case of the Faro interface, you need to select the serial port at which the arm is connected and the Baud rate for the arm connection. As with all interfaces, you'll need to type in your user name and select the desired SA connection from the SA Instrument Connection list (only one connection shown in this example). As with most instrument interfaces, we need to tell the Log On if we are connecting to an arm or running in simulation mode. Assuming an arm is available, check "Connect to Arm". Now press  to connect to SpatialAnalyzer and the arm. The connections will be made, and the main SpatialAnalyzer CMM Interface dialog will appear:



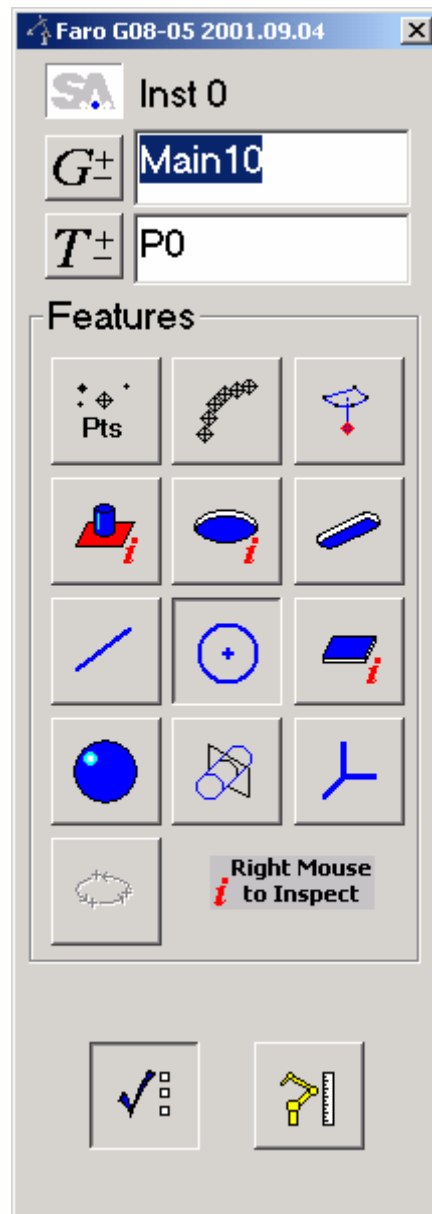
Note the title bar contains the arm designation and the interface version number (release date). Next to the SA logo, the interface displays the SpatialAnalyzer Instrument index. We are connected to Instrument 0 in this example.

The better part of the CMM dialog is taken up by the Features (measure mode) buttons. These buttons are quite large due to the fact that some CMM arms support a mouse mode. In other words, the arm can be put in a mode that allows you to control the windows cursor by moving the arm. In this mode, the arm buttons can be used in place of the mouse buttons. The mouse mode allows you to select measure modes and perform measurements without going back and forth from the arm to the keyboard and mouse. So the large buttons make it easier to select measure modes when using the arm as a mouse. More information on setting and using the mouse mode can be found in the section titled “Customizing the Interface”. The SpatialAnalyzer CMM interface

provides an alternative method for selecting measure modes with the arm. When the arm is not in mouse mode, the arm buttons can be used to select the measure mode. Press and release button one (closest to probe on the Faro) on the arm to iterate through the measure modes. The measure mode button which appears to be pressed is the currently selected measure mode. Here is the main dialog showing the Circle



mode as the selected measure mode:



Press button two to iterate through the measure modes in the opposite direction. Press and hold button one to begin the selected measure mode. All measure mode details are described in the section titled “Measure Modes”.

So measure modes can be selected, and measurements can be started all from the arm buttons. When in the main dialog, short button hits select the measure mode, and a long hit (hold button down for ~1.5 seconds) begins the selected mode. By determining the duration, or length of time the user holds down a button, the interface effectively adds an extra function to each button. The SpatialAnalyzer CMM Interface allows you to program the action of each arm button for both short and long duration button hits. **IMPORTANT:** Since the interface uses each button to perform two functions by determining the duration which the button is held down, **all actions are taken when an arm button is released, not when pressed.** So, you must remember when acquiring data that the data point is taken when you let go of the button, not when you press it. See the next section for more information on programming the button modes.

Note regarding simulation mode: If we had not connected to an arm, the dialog title bar would contain the text “Sim.” as a reminder. When running simulation, the bottom of the dialog will include a row of buttons,



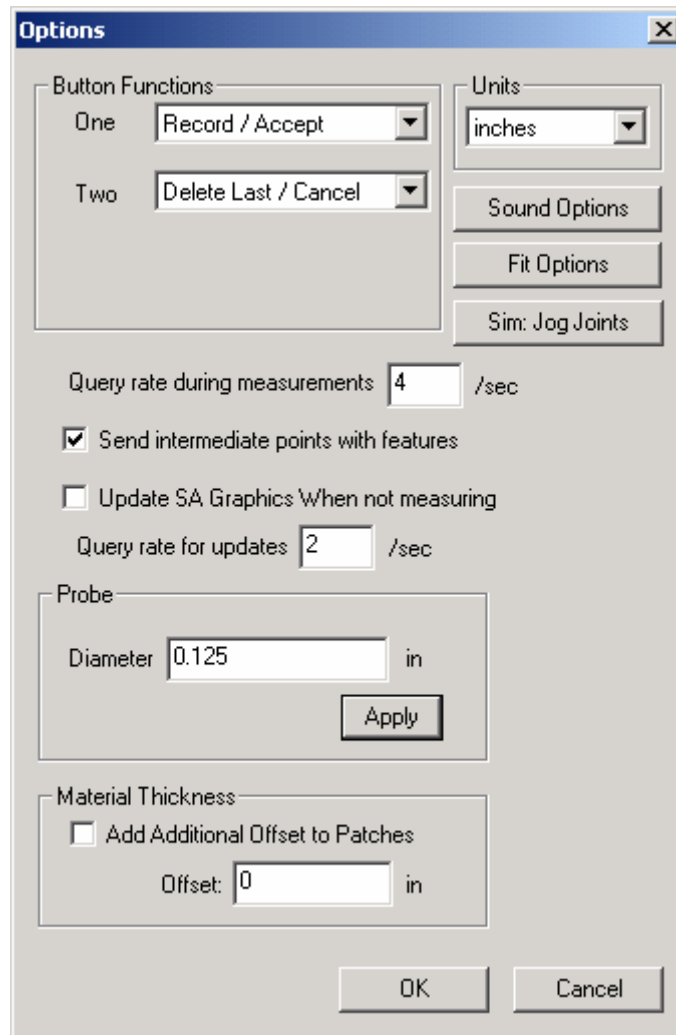
These buttons simulate the arm buttons. They are delineated by button number and duration of simulated button hit. The label meanings are: 1S = button one, short duration; 1L = button one, long duration; 2S = button two, short duration; 2L = button two, long duration.


Customizing the Interface

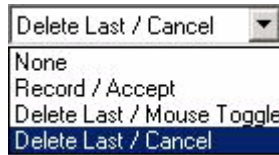
You can control many aspects of the behavior of the SpatialAnalyzer CMM Interface. To



change arm settings, press the Arm Settings Button in the main dialog. The CMM Interface Options Dialog will appear:



In the Button Functions section of the Options Dialog, you will be able to program the action of the arm buttons. In the case of the Romer arm, there will appear a program interface for button three. The default behavior for button one is “Record / Accept”, meaning that a short duration button hit (just press and let go) will result in a Record action, and a long duration (press and hold down for around 1.5 seconds) will result in an Accept action. For most measurement modes, Record will acquire a single point, and Accept will accept the entire measurement. The Accept function is the same as an OK button for an entire process. The default behavior for button two is “Delete Last / Cancel”. Again, the meaning is “Short duration action / Long duration action”. For most measurement modes, Delete Last will remove the last point acquired, and Cancel will cancel an entire measurement. To change the programming for button two, press the  in the corresponding button function drop list. You’ll see the available functions for button two in the drop list:



Just select a new program for the button by single left clicking the desired program in the list. In this example, we'll just keep the default behavior. At the time of this writing, Delete Last / Mouse Toggle is supported only in the Romer interface. When the Mouse mode is toggled on, the arm can be used in place of the mouse to move the Windows cursor around the screen, and the arm buttons can be used in place of the mouse buttons to perform general Windows actions.

Query rate during measurements /sec allows you to set the sampling rate of the arm while it is acquiring position data. In this case, the interface is set to query the arm 4 times per second during measurement operations.

☒ **Send intermediate points with features** allows you to tell the interface whether to send all measured points at the end of a geometry fit type of measurement. For example, if checked, then at the end of a sphere fit measurement, the interface will not only send the fitted sphere and the sphere center point, it will also send all points which were actually measured with the arm and used for the fit. If not checked, the interface will send only the sphere geometry and the sphere center point.

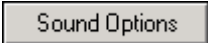
☐ **Update SA Graphics When not measuring** allows you to update the arm model in SpatialAnalyzer even when the arm is not measuring. If left unchecked, the interface will update the model in SA only when actually acquiring data. If you are working with an SA job which contains a large CAD file, you may want to leave this option unchecked in favor of better graphics performance.

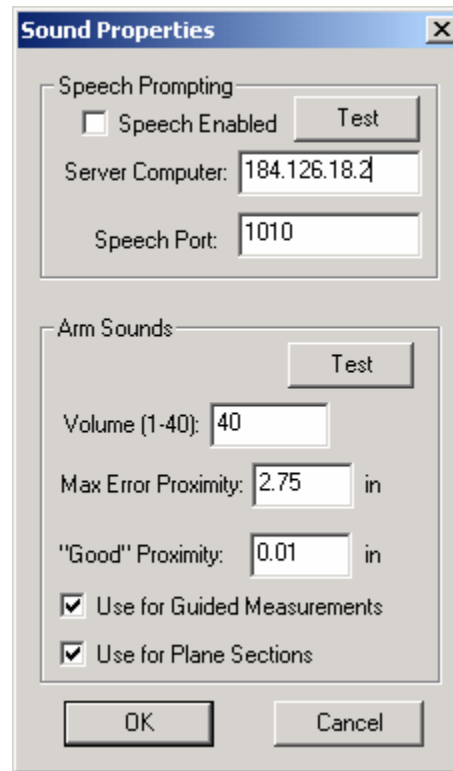
Query rate for updates /sec is the setting for the frequency of SA Graphics updates when not measuring. If ☐ **Update SA Graphics When not measuring** is not checked, then **Query rate for updates** /sec does not apply.

The Probe section of the Options Dialog allows you to select the diameter of the probe currently in use. In the case of the Romer interface, the probe diameter can be auto-detected through the interface. To enter a new probe diameter, just type in the new diameter and press **Apply**. Whenever the probe is changed, a probe calibration should be performed. More information on probe calibration is available in the section titled "Arm Calibration".

The Material Thickness section of the Options Dialog allows you to set a material thickness for the Patch Measure Mode. More information on material thickness will be presented in the section titled "Measure Modes".

The Units section allows you to set the linear units displayed throughout the interface. Just press the down arrow in the Units drop list to view/change the displayed units.

To view or change sound options, press  in the Options Dialog. The Sound Properties Dialog will appear:



The Sound Properties dialog box is titled "Sound Properties" and has a close button (X) in the top right corner. It contains two main sections: "Speech Prompting" and "Arm Sounds".

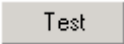
Speech Prompting section:

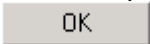
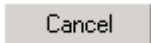
- There is a checkbox labeled "Speech Enabled" which is currently unchecked. To its right is a "Test" button.
- Below this is a text field labeled "Server Computer:" containing the IP address "184.126.18.2".
- Below that is a text field labeled "Speech Port:" containing the value "1010".

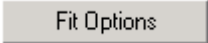
Arm Sounds section:

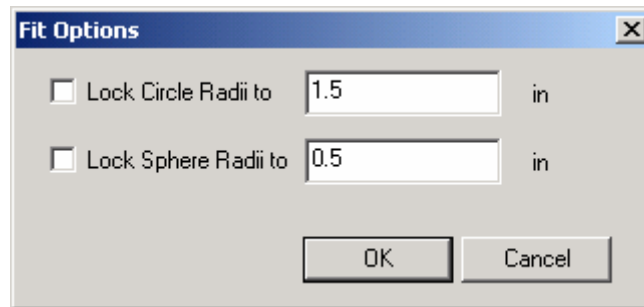
- There is a "Test" button at the top right of this section.
- Below it is a text field labeled "Volume (1-40):" containing the value "40".
- Below that is a text field labeled "Max Error Proximity:" containing the value "2.75", followed by the unit "in".
- Below that is a text field labeled "\"Good\" Proximity:" containing the value "0.01", followed by the unit "in".
- Below these are two checked checkboxes: "Use for Guided Measurements" and "Use for Plane Sections".

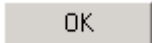
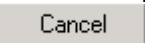
At the bottom of the dialog are two buttons: "OK" and "Cancel".


To enable Speech Prompting, you must be running the SASpeechServer. Please see the SpatialAnalyzer User's Manual for more information on the speech server. If you are running the speech server, and would like to receive voice prompts, just check Speech Enabled, and press . If your settings are correct, the SpatialAnalyzer voice will inform you that all is well.

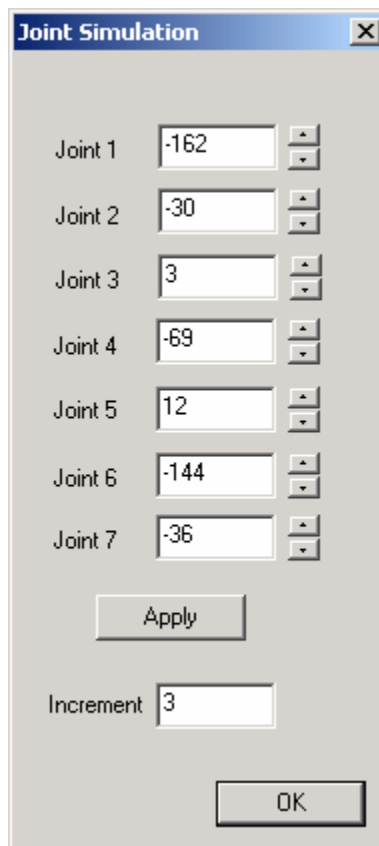
The Arm Sounds section in the Sound Properties dialog allows you to customize the sounds the arm makes during certain operating procedures. Max Error Proximity allows you to set the proximity at which an arm sound will be made. This sound is meant to let you know the arm probe tip is headed away from the desired position. "Good" Proximity is a range within which the arm will give another sound to let you know the arm probe tip is near a desired position. Use for Guided Measurements and Use for Plane Sections are the operating procedures to which the arm sounds are applied if checked. More information on guided measurements and plane sections are presented in the section titled "Measure Modes". Press  to apply changes and close the Sound Properties dialog, or  to ignore changes and close the dialog.


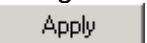
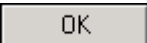

Back in the Options Dialog, press  to view or change fit options. The Fit Options Dialog will appear:

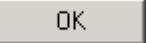



If, for example, Lock Sphere Radii is checked, then all fitted spheres will subsequently be locked to the radius you enter. This feature is commonly used when measuring tooling balls of known radii. The fit routine is locked to the known radius, so that the only variables to solve for are the location of the center of the sphere. Press  to accept and apply any changes you've made, or  to ignore changes. The Fit Options dialog will close either way.

When running in simulation mode, the Options Dialog shows the  button. Here is the Joint Simulation Dialog:



This dialog allows you to update the arm model in SpatialAnalyzer when you are not connected to a real arm. When running a 7 degree of freedom, there are 7 joints which can be manipulated. Just click on the up or down arrow in one of the spin buttons  to move the corresponding joint in the SA model. To change the angular Increment per spin button click, enter a new Increment, and press . Press  or  to close the Joint Simulation Dialog.

Finally, back in the Options Dialog one last time, press  to apply changes and close, or  to ignore changes and close.

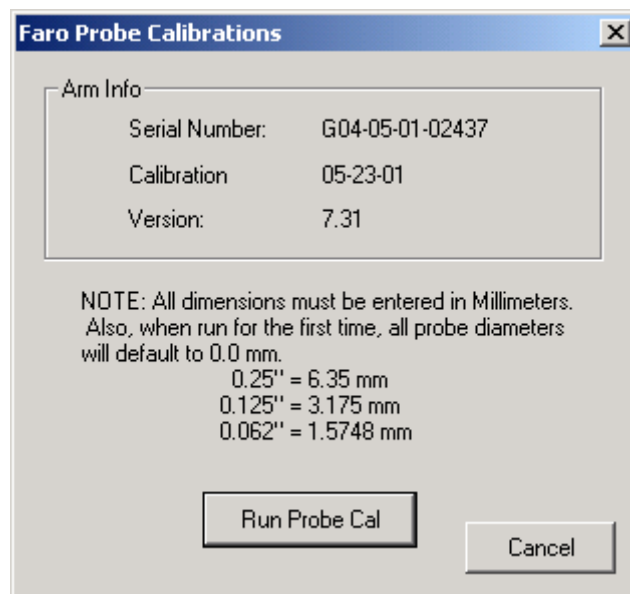
The next section will address the issue of arm calibration. Following that, we'll finally get to some measurements with the arm.

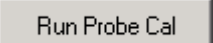
Arm Calibration

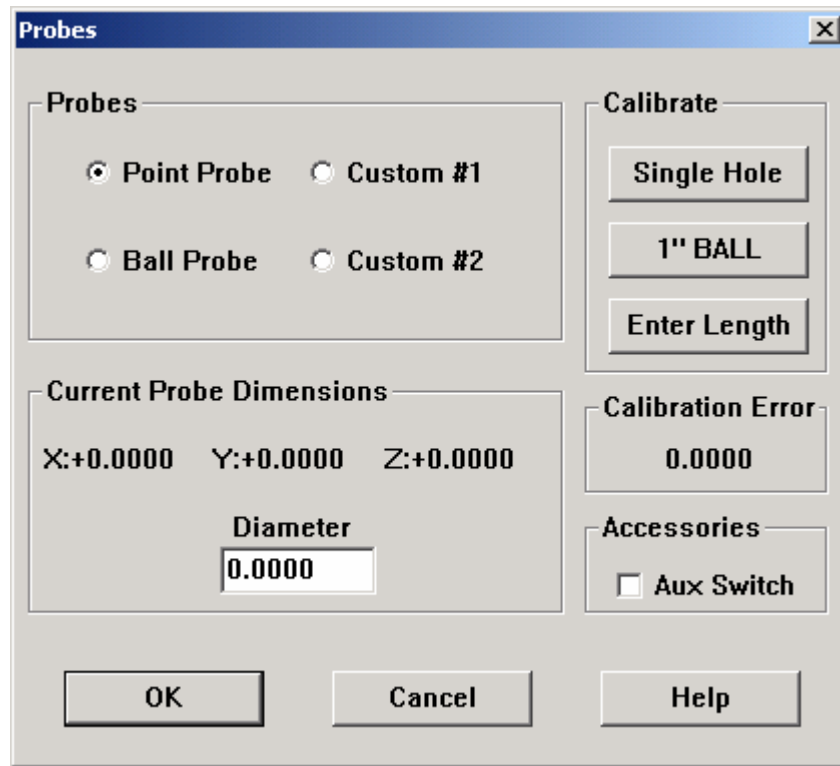
The SpatialAnalyzer CMM Interface allows for arm calibrations by calling the



manufacture's calibration routines. Press  in the main Interface Dialog for a look at the arm calibrations. In this case, the Faro Probe Calibrations dialog will appear:



The Arm Info section is populated with data specific to this arm, and we see some brief instructions for running the Faro Probe Calibration routines. Press  for a look at the Faro Probe Cal interface:

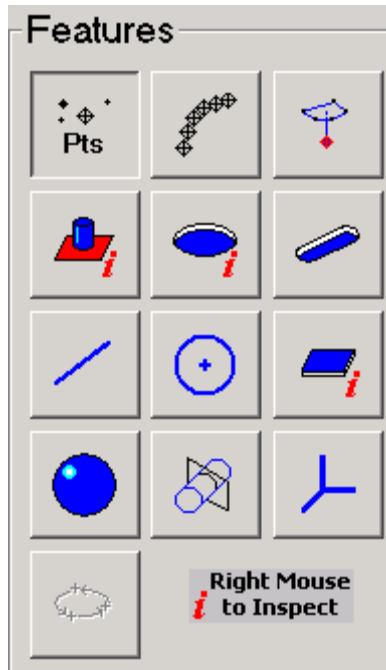


Just enter the appropriate probe diameter (in mm as instructed in the previous dialog), and select a calibration from the Calibrate section. The Faro interface will take you through a scripted process.

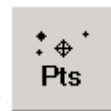
Once the arm is calibrated, you are ready to measure. The next section will address that subject.


Measure Modes

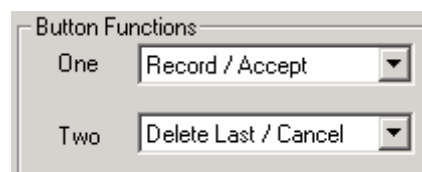
Now that you've seen how to customize the arm interface and calibrate the arm, you're ready to take some measurement data. The Features section of the main SA CMM Interface dialog contains all the available measure modes:



You can hover over a measure mode button to get a mouse prompt text description of



the mode. For example, if you place the mouse over , a small text box next to the mouse cursor will say “Measure Single Points”. To begin any measure mode, simply left click on the measure button. As described in the section titled “Starting the Interface”, you can alternatively iterate through to the desired measurement with the arm buttons, or operate the arm in mouse mode. And as described “Customizing the Interface”, the buttons will perform the actions you set in the Button Functions section of the Options Dialog. When running an arm, remember this **IMPORTANT NOTE**: Since the interface uses each button to perform two functions by determining the duration which the button is held down, **all actions are taken when an arm button is released, not when pressed**. So, you must remember when acquiring data that the data point is taken when you let go of the Record / Accept button, not when you press it. This note does not apply to simulation mode, the button action is taken when the corresponding simulation button is pressed. We’ll cover all the measure modes, corresponding arm button actions, and common usages next. For all the mode descriptions, we’ll assume the default programming for the simulated arm buttons:




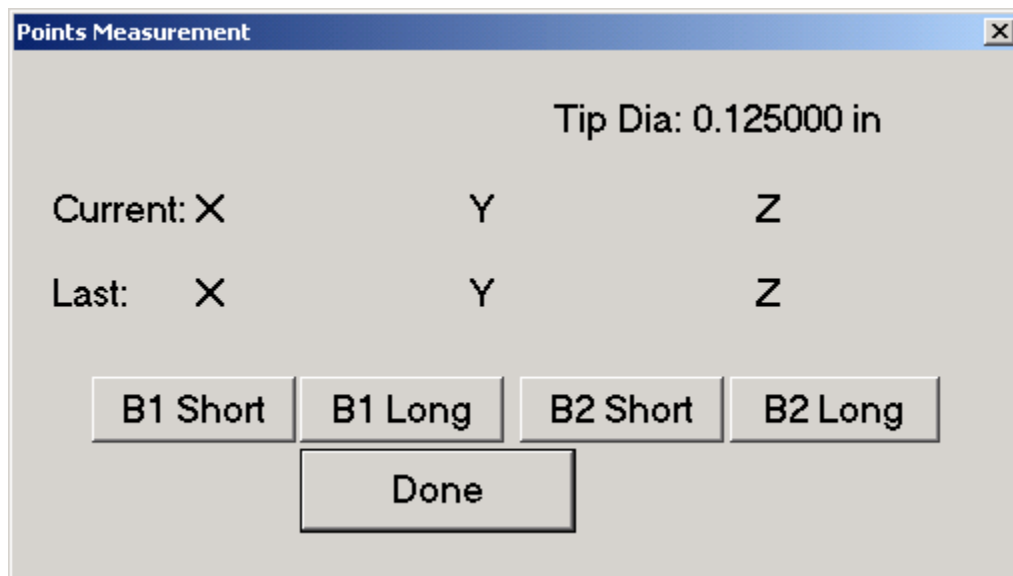
Note regarding simulation mode: All measure modes which provide a simulation will have a row of simulated arm buttons near the bottom of the measurement dialog:



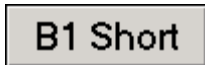
As described in the section titled “Starting the Interface”, these buttons simulate the arm buttons when in simulation mode. They are delineated by button number and duration of simulated button hit. The label meanings are: B1 Short = button one, short duration; B1 Long = button one, long duration; B2 Short = button two, short duration; B2 Long = button two, long duration.



Single Points Press . The Points Measurement Dialog will appear (shown here as if in simulation mode)



Note the current Tip Diameter, set in the Options Dialog, is displayed. Press and release a short duration (≤ 1 sec.) of the Record / Accept button, or press



if you're running in simulation mode. A data point will be acquired at the current position of the probe tip, and that point will be sent to SpatialAnalyzer. The Points Measurement Dialog will now look similar to:

Points Measurement

1 Pts Measured Tip Dia: 0.125000 in

Current: 122.87 2.04 -16.00

Last:

B1 Short B1 Long B2 Short B2 Long

Done

Press and release a short duration (≤ 1 sec.) of the Record / Accept button, or press **B1 Short** if in simulation, again to acquire another point. The dialog will now look like:

Points Measurement

2 Pts Measured Tip Dia: 0.125000 in

Current: 117.76 23.16 -4.13


Last: 122.87 2.04 -16.00

B1 Short B1 Long B2 Short B2 Long

Done

So the Points Measurement dialog always displays the Cartesian position of the two most recent points acquired. Note that with this measure mode, points are sent to SA as soon as they are acquired, so the Delete button function takes no action. To close the Points Measurement Dialog, press a long duration of the Record / Accept button, or **B1 Long** if in simulation. This is the equivalent an OK button. In this case, the Cancel button will perform the same function. Since there is no process to cancel, it will

A rectangular button with a light gray background and a thin black border. The word "Done" is centered in a bold, black, sans-serif font.

simply close the dialog also. Finally, the  button, which is available when connected to an arm, is a third equivalent way to close the Points Measurement Dialog. This button is for users who may be at the computer screen and wish to close the dialog without having to press an arm button. Most measure mode dialogs will have

A rectangular button with a light gray background and a thin black border. The word "Done" is centered in a bold, black, sans-serif font.

buttons similar to .

Button Functions:

Record – acquires a single point. The Point is immediately sent to SA.

Accept – closes the dialog and ends the measurement.


Delete Last – takes no action in this case. All data is sent to SA as it is acquired.

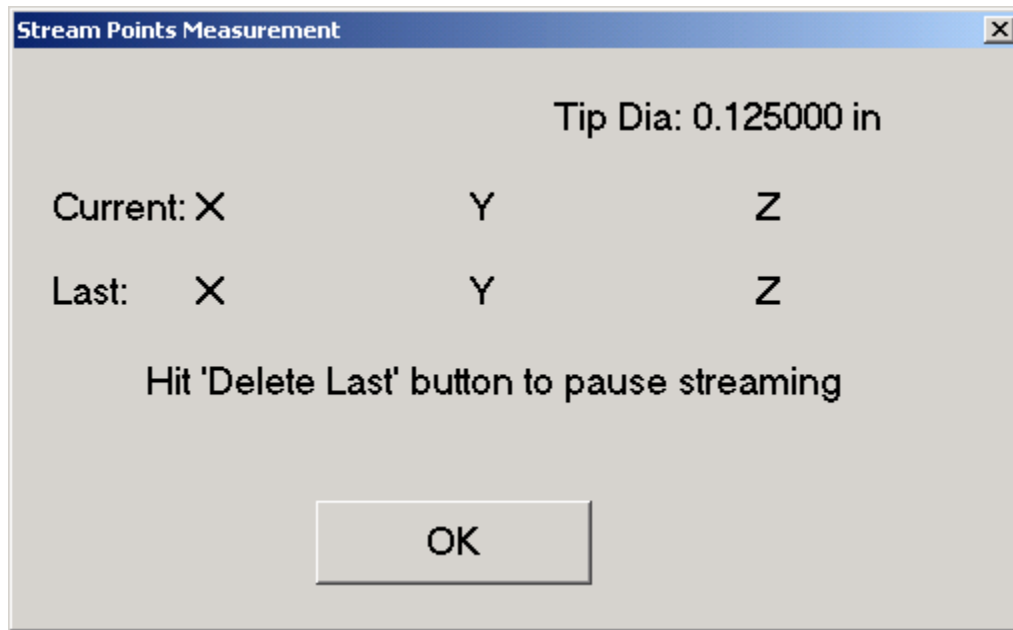
Cancel – closes the dialog and ends the measurement.

Common Uses:

Often measurement fixtures contain points which have relatively known positions. These points are often used to locate a metrology instrument in the frame of the fixture. The single Points Measurement mode is used to acquire individual points in discrete positions.



Stream Points: Now press  in the main interface dialog. The Stream Points Measurement Dialog will appear:



Press and release a short duration (≤ 1 sec.) of the Record / Accept button to begin streaming points to SpatialAnalyzer.

The interface will continue streaming points until you tell it to stop. You can send separate streams by pausing and restarting the stream. As instructed in the Stream Points Measurement Dialog, you pause with a short duration press of the Delete Last button. You can subsequently restart a new stream by once again pressing the Record button.

Button Functions:

Record – begins streaming points to SpatialAnalyzer. The points are sent to SA as they are acquired.

Accept – closes the dialog and ends the measurement.


Delete Last – takes no action in this case. All data is sent to SA as it is acquired.

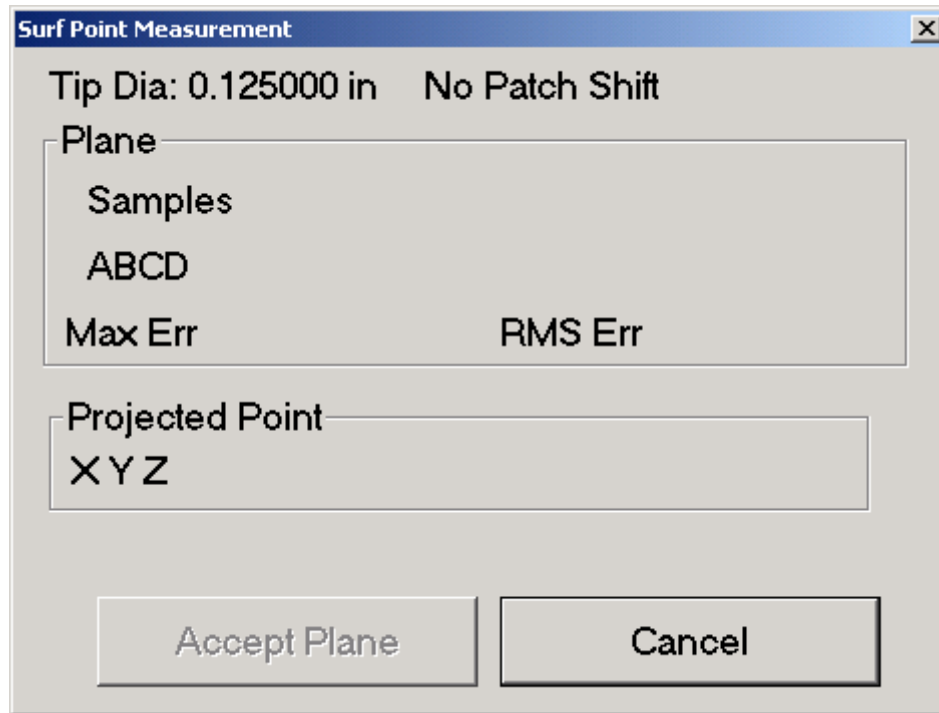
Cancel – closes the dialog and ends the measurement.

Common Uses:

Streamed points are most often used to fit a geometry or a surface in SA. The ability to pause and restart this mode is provided specifically for surface fits. Any time data is to be fit with a method available in SpatialAnalyzer, Streamed Points is likely the best measure mode to use.



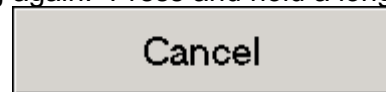
Patch Measurement: Press  to measure Patch Surface Points. The Surf Point Measurement Dialog will appear:



The dialog box is titled "Surf Point Measurement" and contains the following fields and buttons:

- Tip Dia: 0.125000 in No Patch Shift
- Plane
- Samples
- ABCD
- Max Err
- RMS Err
- Projected Point
- XYZ
- Accept Plane
- Cancel

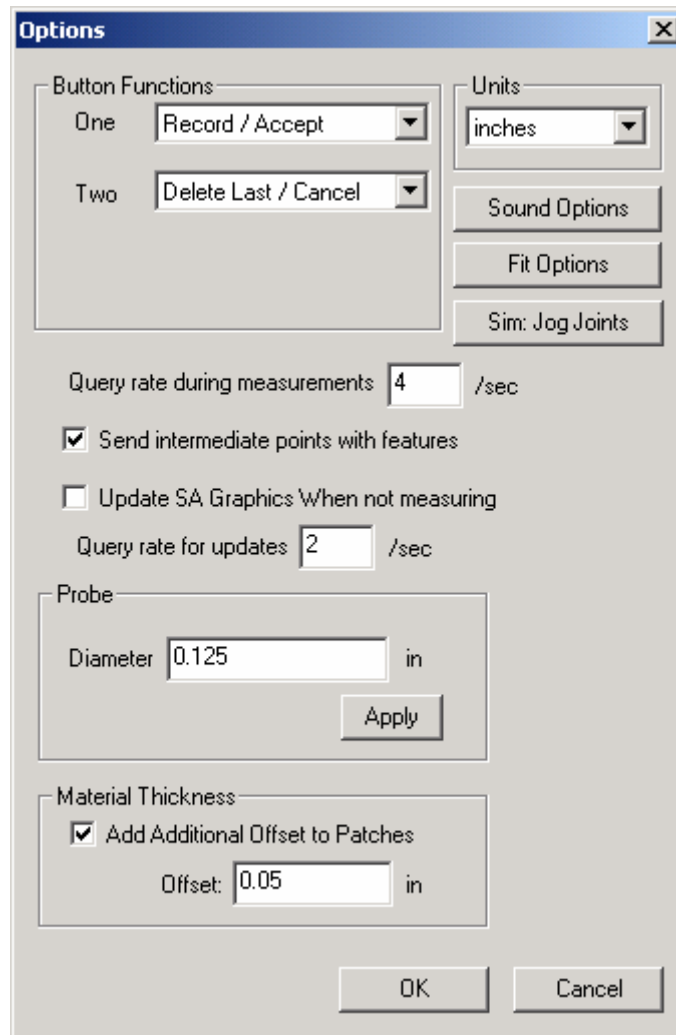
As usual, the Tip Diameter which we set earlier in the Options Dialog is shown. Note also that "No Patch Shift" indicates the current "Material Thickness" setting from the Options Dialog. We'll look at the material thickness setting again. Press and hold a long



duration of the Delete Last/Cancel arm button, or press with the mouse to close the Surf Point Measurement Dialog. Back in the main interface



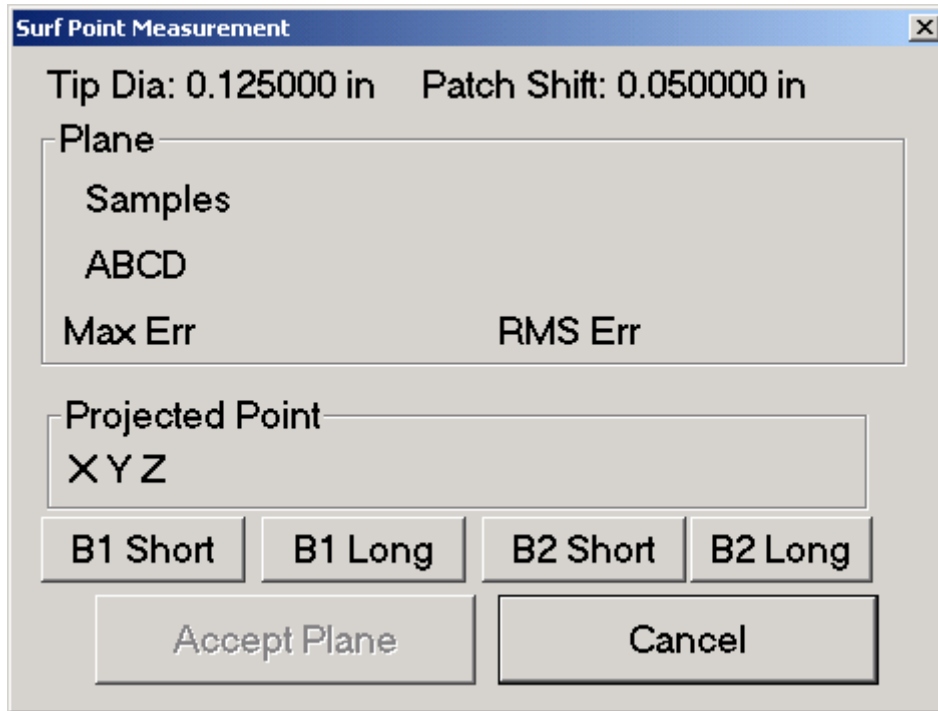
dialog, press  to go back to the Options Dialog and revisit the arm settings:



In the Material Thickness section, type in 0.05” (or an appropriate value), and check ☒ Add Additional Offset to Patches to apply the Material Thickness. Press to



accept the changes and close the Options Dialog. Now press in the main dialog again to go back into the Surf Point Measurement Dialog:



Now the dialog shows the 0.05" patch shift we just entered in the Options dialog. Press and release a short duration (≤ 1 sec.) of the Record / Accept button to acquire the first plane (patch) point. As with the single point mode, continue to press the Record button to acquire more plane points. After at least three plane points have been acquired, the interface will fit a plane to all the plane points with each newly acquired point. The results, ABCD of the fitted plane along with Maximum and RMS errors, will be displayed. When you are done acquiring plane points, press a long duration (hold until you hear a beep, ~ 1.5 sec.) of the Record / Accept button to accept the plane portion of the measurement. You can also press the Accept Plane button with the mouse, which becomes enabled as soon as at least 3 points are acquired. Now press and release a short duration (≤ 1 sec.) of the Record / Accept button if you're running an arm, to acquire the Projected Point. The Surf Point Measurement Dialog will display the Cartesian location of the measured point. If you are not satisfied with the measurement, just re-acquire. If you are satisfied with the measurement, press another long duration (hold until you hear a beep, ~ 1.5 sec.) of the Record / Accept button to accept the entire measurement. You'll also notice that the Accept Plane button has become the Accept Point button. It now has two functions – to accept the projected point, and indeed the entire measurement.

Button Functions:

Record – acquires a single point. If measuring the plane, the Record button also causes a plane fit to be performed so long as at least 3 plane points have been acquired. If the plane has been accepted, acquires the projected point.

Accept – If measuring the plane, accepts the fitted plane and makes the dialog ready for the Projected Point. If the Projected Point has been measured, accepts the entire measurement, and closes the dialog. The data is then sent to SA. Note: The long

duration press of the Record / Accept button is functionally identical to the Accept button in the dialog. The difference is that one button is located on the arm, the other is in the dialog.


Delete Last – If measuring the plane, deletes the last measured plane point and eliminates it from subsequent plane fits. If measuring the Projected Point, removes the last logged measurement. You can also just Record over the last logged Projected Point.

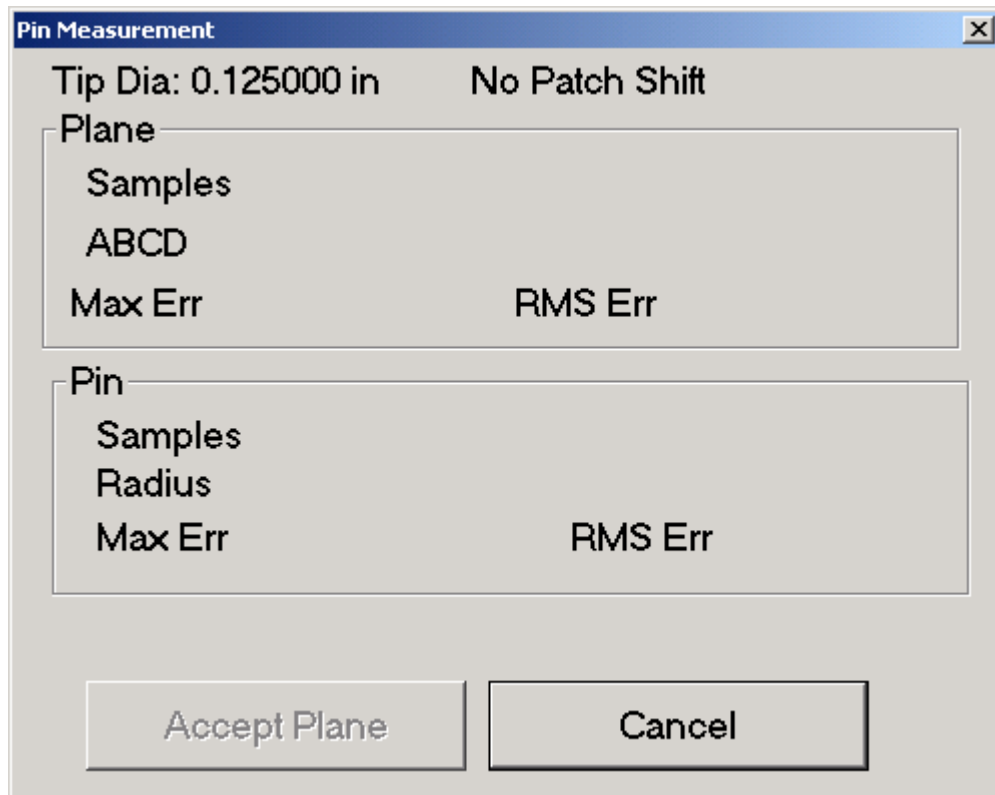
Cancel – closes the dialog and ends the measurement. No data is sent to SA. Note: The long duration press of the Delete Last / Cancel button is functionally identical to the Cancel button in the dialog. The difference is that one button is located on the arm, and the other is in the dialog.

Common Uses:

The Patch Measurement, or Surface Point Measurement, is often used to measure a point on the “hidden” side of a surface. For example, you may need to know the position of a point on the inside of the sheet metal skin of a machine part. So long as the material thickness of the skin is known, the Patch Measurement mode can be used to measure the point.



Pin Measurement: Press  to measure a pin. The Pin Measurement Dialog will appear:



Note the Patch Shift is shown in this dialog. A material thickness can be used in the same way as with the Patch Measurement. Please see the Patch Measurement description in this section for more information on setting and using the Patch Shift. This is another measure mode which contains more than one sub-procedure – a plane and a pin (circle) in this case. Press and release a short duration (≤ 1 sec.) of the Record / Accept button to acquire the first plane (patch) point. As with the single point mode, continue to press the Record button to acquire more plane points. After at least three plane points have been acquired, the interface will fit a plane to all the plane points with each newly acquired point. The results, ABCD of the fitted plane along with Maximum and RMS errors, will be displayed. When you are done acquiring plane points, press and release a long duration (hold until you hear a beep, ~ 1.5 sec.) of the Record / Accept button to accept the plane portion of the measurement. You can also press the Accept Plane button, which becomes enabled as soon as at least 3 points are acquired. Now press and release a short duration (≤ 1 sec.) of the Record / Accept button if you're running an arm, to acquire the first Pin point. After at least 3 Pin points have been acquired, the interface will fit a circle to all the pin points with each newly acquired point. The Radius, Maximum error, and RMS error of the Pin (circle) will be displayed. If you are satisfied with the measurement, press a long duration (hold until you hear a beep, ~ 1.5 sec.) of the Record / Accept button to accept the entire measurement. You'll also notice that the Accept Plane button has become the Accept Pin button. It now has the two functions – to accept the pin measurement, and close the dialog.

Button Functions:

Record – acquires a single point. If measuring the plane, the Record button also causes a plane fit to be performed so long as at least 3 plane points have been acquired. If measuring the pin, the Record button also causes a circle fit to be performed so long as at least 3 circle points have been acquired.

Accept – If measuring the plane, accepts the fitted plane and makes the dialog ready for the Pin (circle) points. If the Pin Points have been measured, projects the pin circle points to the plane, accepts the entire measurement, and closes the dialog. The data is then sent to SA. Note: The long duration press of the Record / Accept button is functionally identical to the Accept button in the dialog. The difference is that one button is located on the arm, the other is in the dialog.


Delete Last – If measuring the plane, deletes the last measured plane point and eliminates it from subsequent plane fits. If measuring the pin, deletes the last measured pin point and eliminates it from subsequent circle fits.

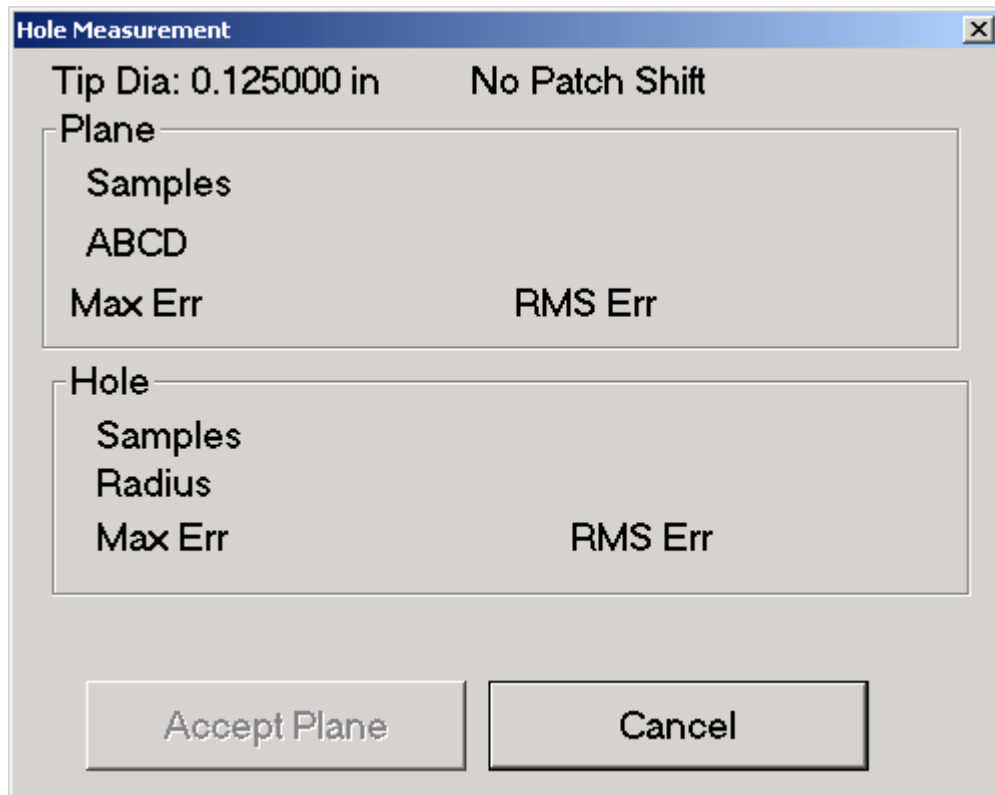
Cancel – closes the dialog and ends the measurement. No data is sent to SA. Note: The long duration press of the Delete Last / Cancel button is functionally identical to the Cancel button in the dialog. The difference is that one button is located on the arm, the other is in the dialog.

Common Uses:

Pins, like tooling balls, are often used to locate instruments with respect to CAD models. The Pin mode also serves as a cylinder measurement. Given that the resulting circle is projected to the plane, and can be shifted by the arm tool tip diameter plus a patch offset, the measurement result can be located at the true base of the cylinder, or even inside the planar surface.




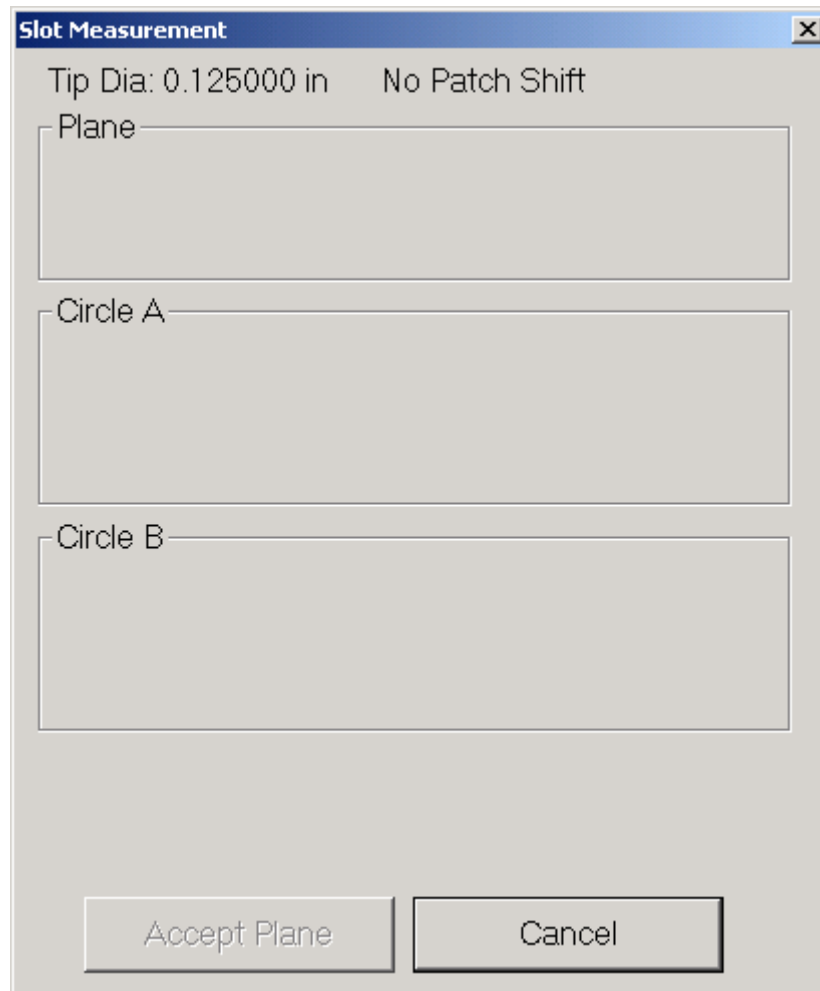
Hole Measurement: Press  to measure a hole. The Hole Measurement Dialog will appear:



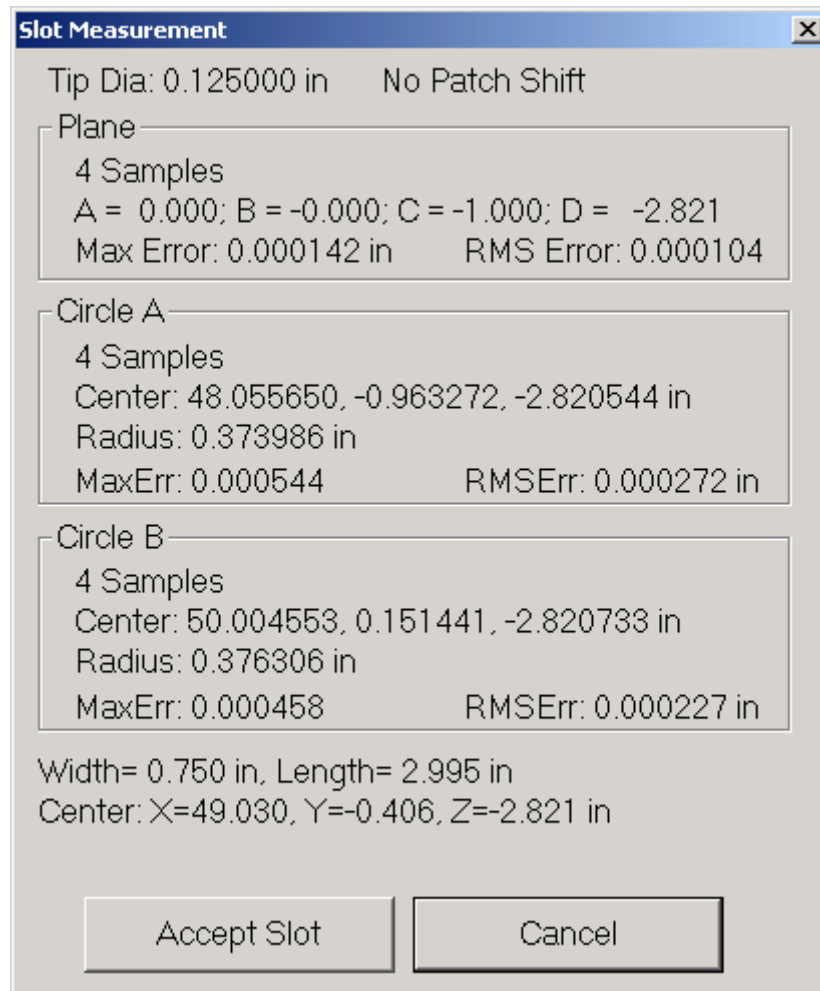
This measurement mode is functionally identical to the Pin Measurement. The only difference is that the pin measurement involves fitting a circle to the base of a cylindrical object and projecting the fitted circle to a previously measured plane, whereas the Hole Measurement involves measuring a circular hole and projecting that to a previously measured plane. Please see the description of the Pin Measurement in this section for details.




Slot Measurement: Press  to measure a slot. The Slot Measurement Dialog will appear:

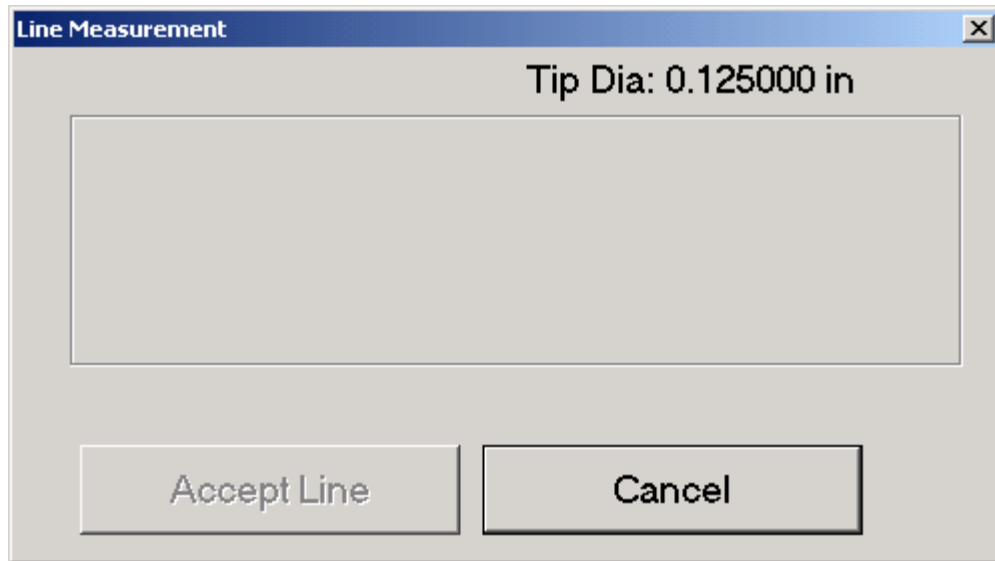


The Slot Measurement is procedurally quite similar to the Pin and Hole Measurements. The difference is the fact that the Slot Measurement involves two holes (circles) instead of one. Therefore, there is simply one extra step in the Slot Measurement. Please see the description of the Pin Measurement in this section for details. When the Slot Measurement is done and ready for acceptance, the Slot Measurement will look similar to:

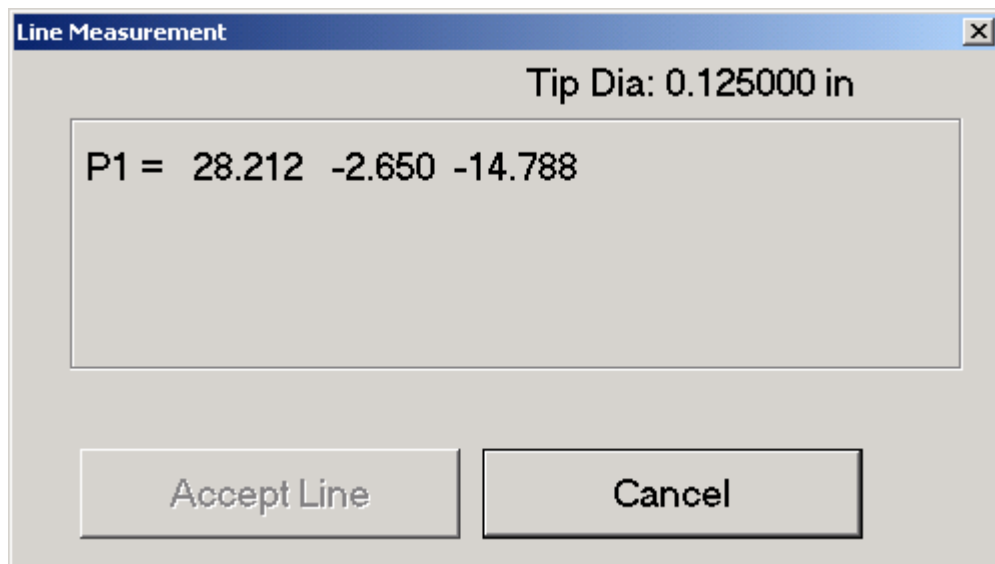


Note the extra results displayed near the bottom of the Slot Measurement Dialog. These report the width, length, and center of the slot after both holes (Circle A and Circle B) have been fitted and projected to the Plane.

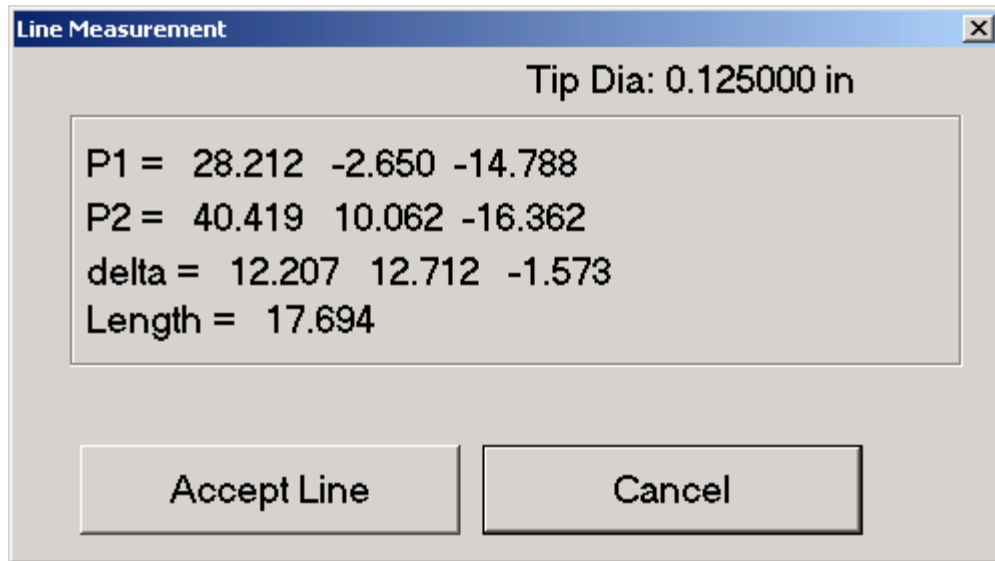
Line Measurement: Press  to measure a line. The Line Measurement Dialog will appear:



Press and release a short duration (≤ 1 sec.) of the Record / Accept button to acquire the first line point. The dialog will then look something like:



The Cartesian position of the first point, P1, is shown. Now press and release a short duration (≤ 1 sec.) of the Record / Accept once more to acquire the second line point. The dialog will then look something like:



Now both line points, P1 and P2, are displayed. Delta is the Cartesian difference in the point positions, and Length is the resulting length of the line.

Button Functions:

Record – acquires a line point.

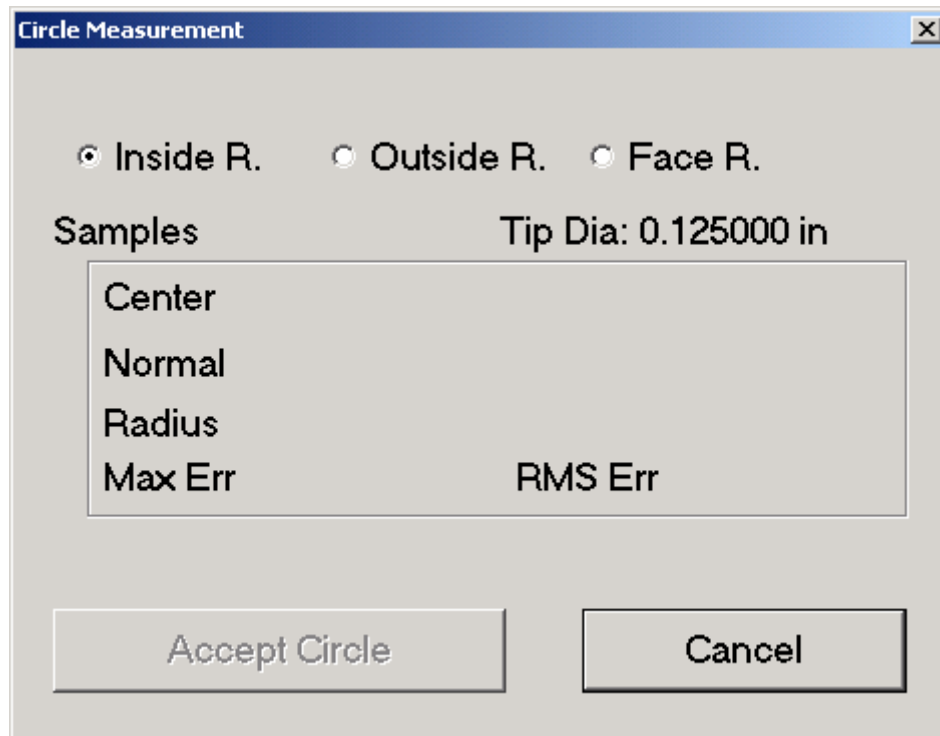
Accept – accepts the line, closes the dialog, and sends the data to SA. Does nothing if both line points have not been measured.

Delete Last – removes the last measured line point.

Cancel – closes the dialog and ends the measurement. No data is sent to SA.




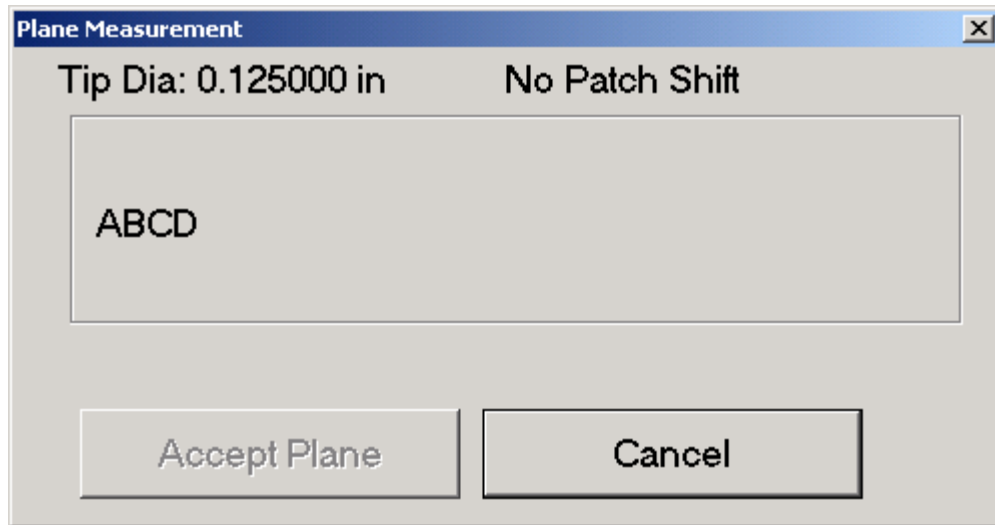
Circle Measurement: Press  to measure a circle. The Circle Measurement Dialog will appear:



The Circle Measurement procedure is nearly identical to the circle measurement steps in the Pin, Hole, and Slot Measurements. Note that you are required to set the radial measurement offset ☒ Inside R. ☐ Outside R. ☐ Face R.. This is because the general circle fit can offset the fitted circle with respect to the measured data, but must be told which direction to offset. In the Pin Measurement, the circle is always measured Outside the Radius. In the Hole and Slot Measurements, the circles are always measured Inside the Radius. Please see the description of the Pin Measurement in this section for all other details regarding the Circle Measurement procedure.



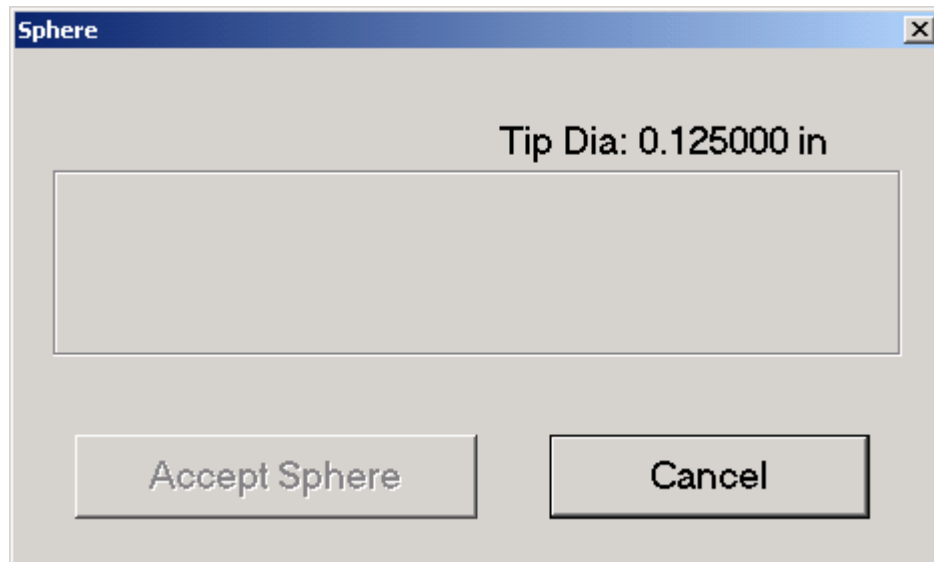
Plane Measurement: Press  to measure a plane. The Plane Measurement Dialog will appear:



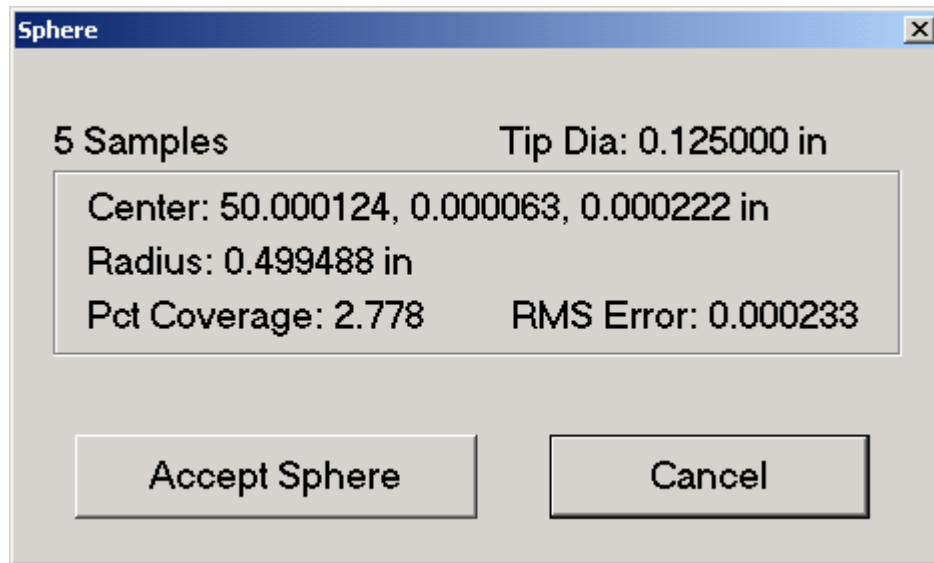
The Plane Measurement is functionally identical to the first step in the Pin, Hole, and Slot Measurements. Please see the description of the Pin Measurement in this section for details.



Sphere Measurement: Press  to measure a sphere. The Sphere Measurement Dialog will appear:



Press and release a short duration (≤ 1 sec.) of the Record / Accept button to acquire the first sphere point. As with the single point mode, continue to press the Record button to acquire more sphere points. After at least four sphere points have been acquired, the interface will fit a sphere to all the sphere points with each newly acquired point. The Sphere Measurement Dialog will then resemble:



The results displayed are the Cartesian location of the fitted Center of the sphere, the resulting Radius of the sphere, the Percent Coverage of the sphere given the fit and the locations of the measured points, and the RMS Error of the fit.

Button Functions:

Record – acquires a single point. If at least 4 points have already been acquired, the Record button also causes a sphere fit to be performed.

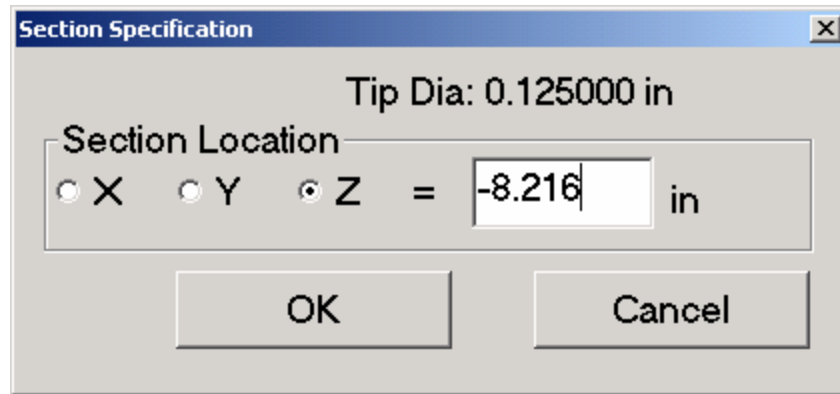
Accept – Accepts the entire measurement, and closes the dialog. The data is then sent to SA. Note: The long duration press of the Record / Accept button is functionally identical to the Accept Sphere button in the dialog. The difference is that one button is located on the arm, the other is in the dialog.

Delete Last – Deletes the last measured sphere point and eliminates it from subsequent sphere fits.

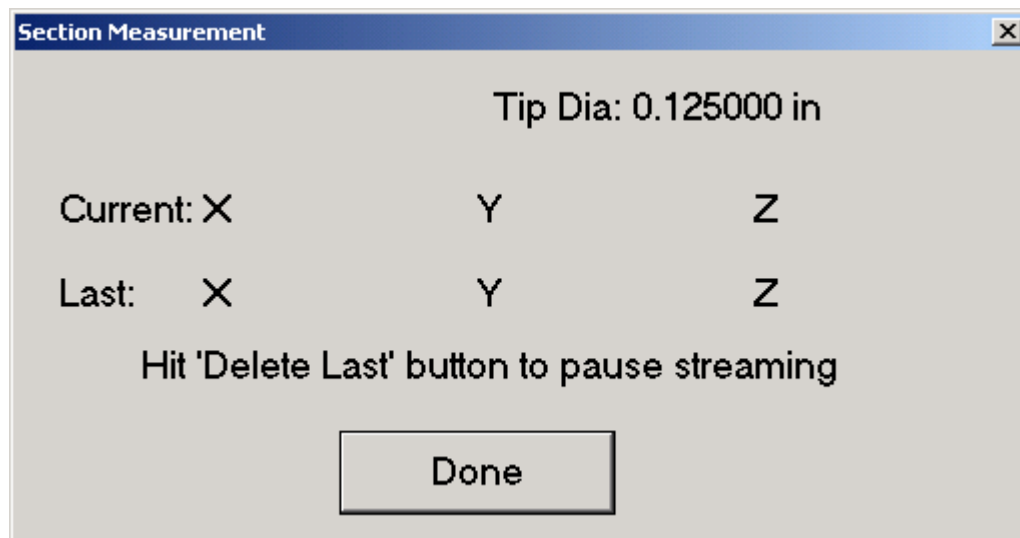
Cancel – closes the dialog and ends the measurement. No data is sent to SA. Note: The long duration press of the Delete Last / Cancel button is functionally identical to the Cancel button in the dialog. The difference is that one button is located on the arm, and the other is in the dialog.



Cross Section Measurement: Press  to measure a Cross Section. The Section Specification Dialog will appear:



The Section Location section tells the Cross Section Measurement where to collect data. To use this measure mode, you must be familiar with creating frames and setting the Working Frame in SpatialAnalyzer. Please see the SpatialAnalyzer User's Manual for more information. Be sure the appropriate frame in SA is the current Working Frame. Select the appropriate plane in the Section Location area. For example, if Z is selected, then a plane parallel to the XY plane will be defined. Type in the value for the plane. For example, -18.28 in. defines a plane parallel to the XY plane, and crossing the Z-axis at Z = -18.28 inches. Press OK to accept the Section Specification. The Section Measurement Dialog will appear:



Press and release a short duration (≤ 1 sec.) of the Record / Accept button to begin sampling data. A new point will be acquired and sent to SpatialAnalyzer whenever the center of the arm probe tip crosses the plane you defined in the Section Specification Dialog. The interface will continue sampling data until you tell it to stop. As instructed in the Section Measurement Dialog, you can pause sampling with a short duration press of the Delete Last button. You can subsequently restart a new sample stream by once again pressing the Record button.

Button Functions:

Record – begins sampling data. Points are acquired and sent to SA when the center of the arm probe tip crosses the plane defined in the Section Specification Dialog.

Accept – closes the dialog and ends the measurement.


Delete Last – takes no action in this case. All data is sent to SA as it is acquired.

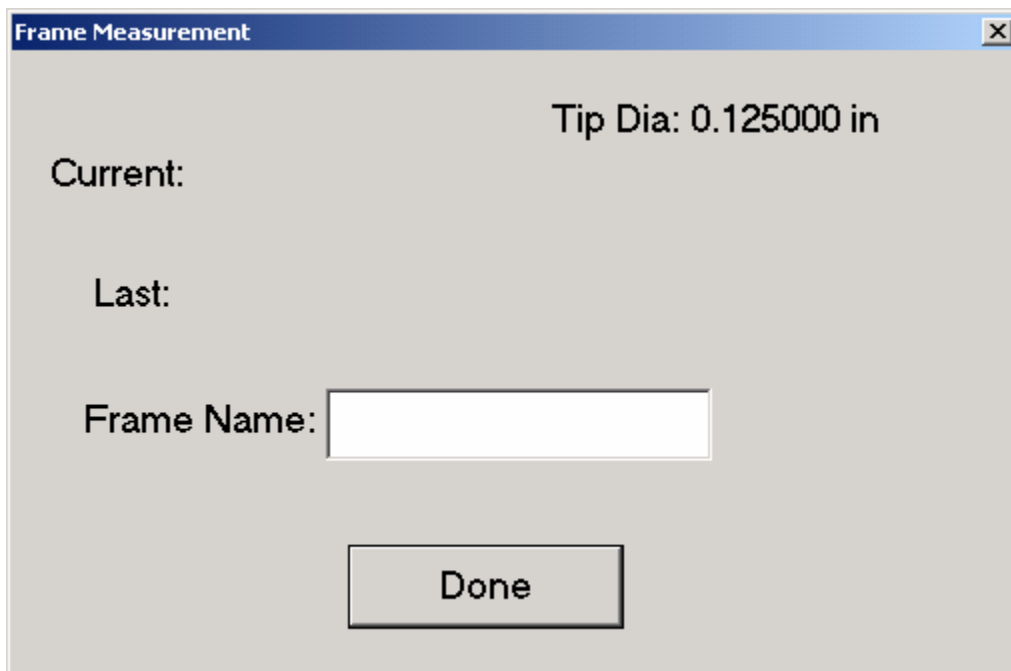
Cancel – closes the dialog and ends the measurement.

Common Uses:

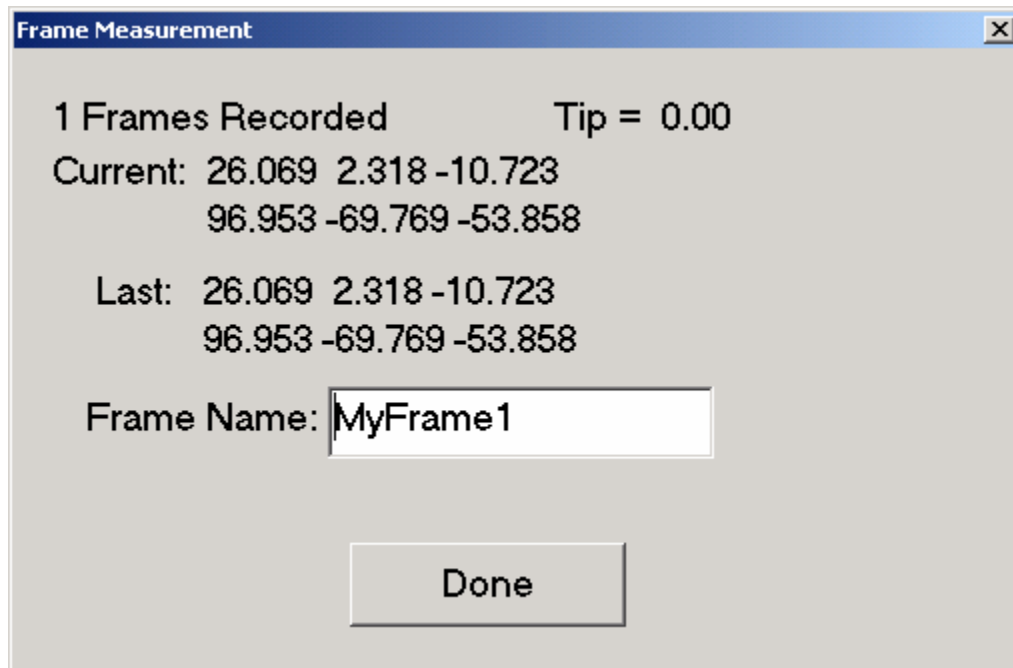
Section Measurement could be considered the inverse of a plane fit. Section Measurement allows you to define a plane in space, and then acquire data which lies on that plane.



Frame Measurement: Press  to measure frames. The Frame Measurement Dialog will appear:



Type in a name for the first frame, and press and release a short duration (≤ 1 sec.) of the Record / Accept button if you're running an arm. A frame will be acquired with the origin at the current position of the probe tip, and the orientation such that the Z-axis of the frame is parallel with the axis of the straight probe. The frame will be sent to SpatialAnalyzer, with the Name you have given it. The name in the dialog will be incremented, and the Frame Measurement Dialog will now look similar to:



So far only one frame has been recorded, so Current and Last show the same frame. The reported results are the translation (xyz in current display units), and rotation (degrees) of the measured frame with respect to the current Working Frame in SA.

Button Functions:

Record – acquires a Frame. The Frame is immediately sent to SA.

Accept – closes the dialog and ends the measurement.

Delete Last – takes no action in this case. All data is sent to SA as it is acquired.


Cancel – closes the dialog and ends the measurement.

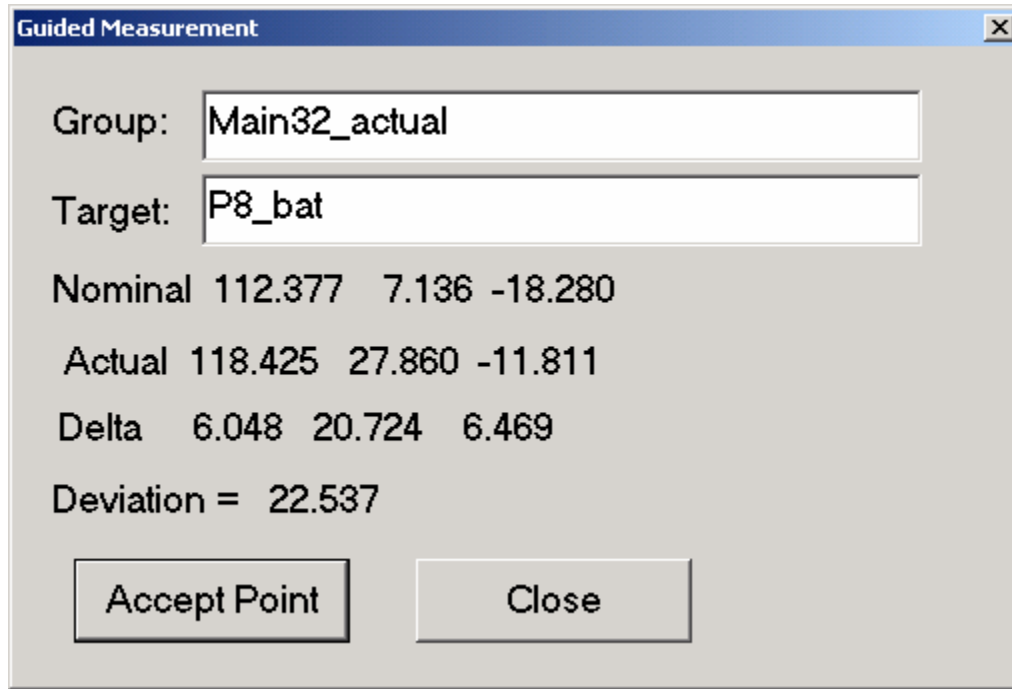
Common Uses:

Frame Measurement is the same as point measurement with the addition of orientation. Point Measurements result in three dimensional locations, whereas Frame Measurements result in six dimensional positions and orientations.

Guided Measurement: To use this mode, you must be familiar with Batch Point Measurement in SpatialAnalyzer. In SA, go to the Instrument menu, and expand Automatic Measurement. Select Measure Batch of Points. You'll be prompted to select the points you'd like to measure, and for a group and target suffix for the batch. You will then be prompted to select the instrument for transmitting the batch. When the batch instructions are transmitted, the Guided Measurement button will become enabled.



Press  to begin the Guided Measurement. The Guided Measurement Dialog will appear:



Guided Measurement

Group: Main32_actual

Target: P8_bat

Nominal 112.377 7.136 -18.280


Actual 118.425 27.860 -11.811

Delta 6.048 20.724 6.469

Deviation = 22.537

Accept Point Close

The dialog shows the identifying Group and Target names of the current Nominal point in the batch. The Cartesian position of the current Nominal point is shown. The real-time Actual Cartesian position of the arm probe tip is shown. The Cartesian Delta between the batch point and the probe tip is shown, and the magnitude Deviation

between the batch point and the probe tip is shown. Press , or press and release a short duration (≤ 1 sec.) of the Record / Accept button to accept the batch point and move on to the next one.

Button Functions:

Record – accepts a single point. Same as pressing the Accept Point button.



Accept – closes the dialog and ends the measurement. Same as pressing the Close button.

Delete Last – takes no action in this case. All data is sent to SA as it is acquired.


Cancel – closes the dialog and ends the measurement. . Same as pressing the Close button.

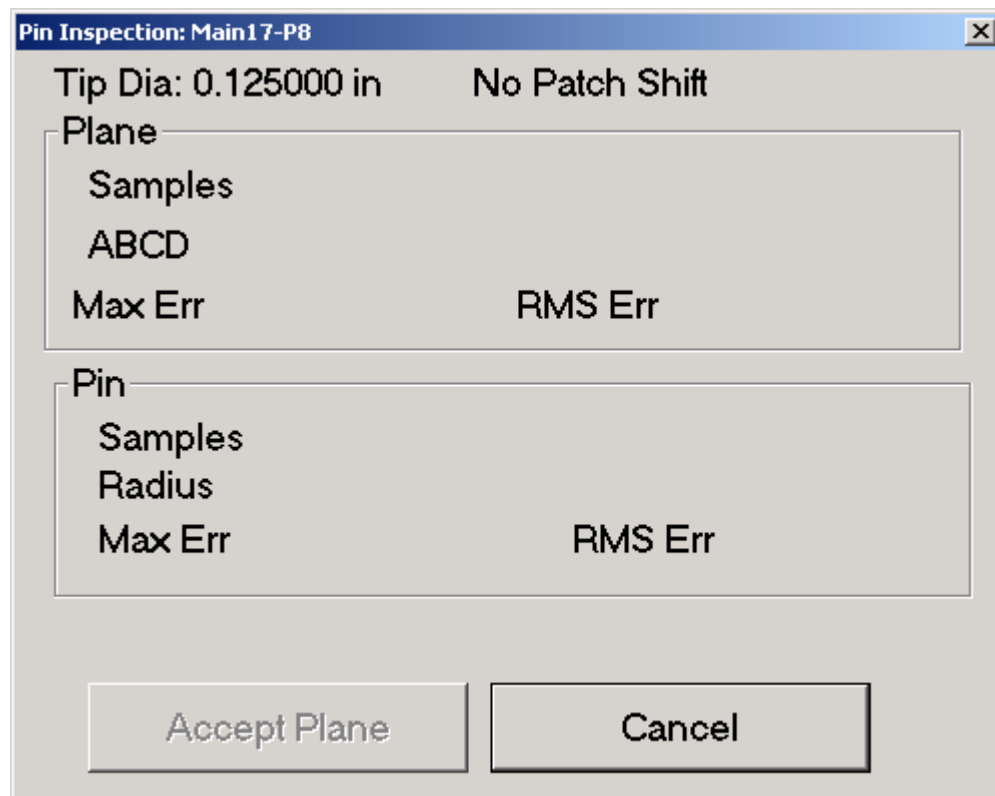
Common Usage:

The Batch Measurement can be used as a diagnostic to check the drift of a fixture with respect to the instrument.

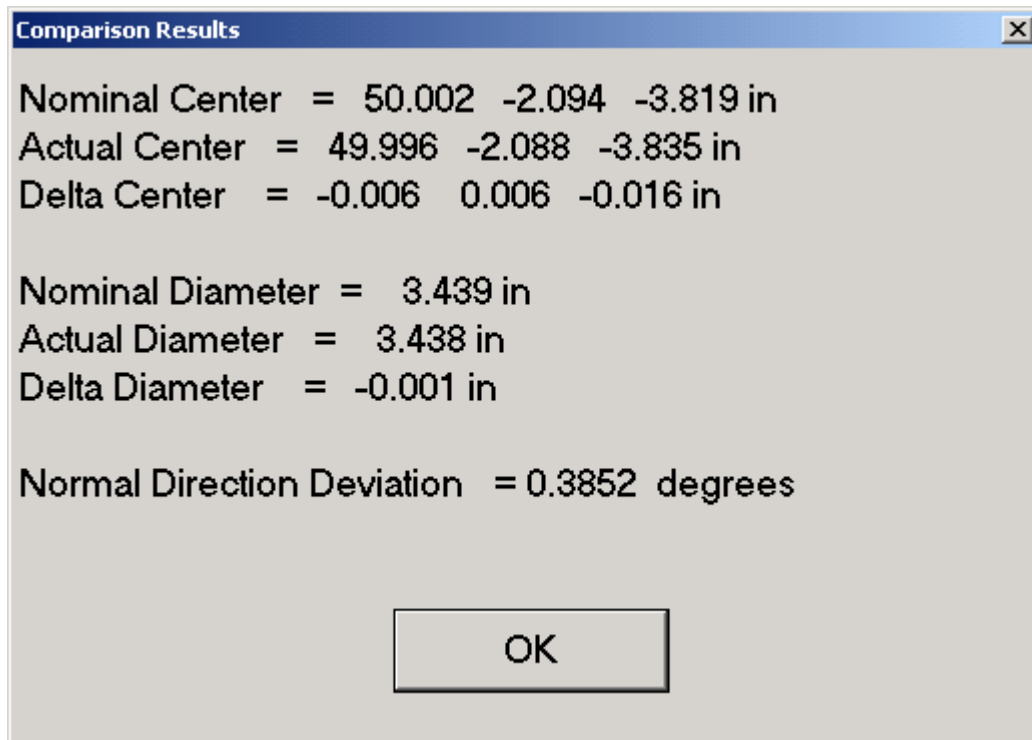
Inspections: Notice  in the main dialog. Any measure button which includes the  in its icon can be used to inspect a previously measured entity of the



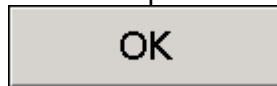
same type. For example, right press  to inspect a previously measured pin. The Pin Measurement Dialog will not appear immediately. First, drag the mouse over SpatialAnalyzer. You'll see a mouse prompt instructing you to "Select Circle to Inspect". Select the Pin in question by left double clicking directly on the representative circle in SA. Now the Pin Inspection Dialog will appear:



Note the dialog title bar indicates the name of the Pin Center in SA. Measure the pin exactly as described in the Pin Measurement description earlier in this section. When the pin is accepted, the Comparison Results Dialog will appear:



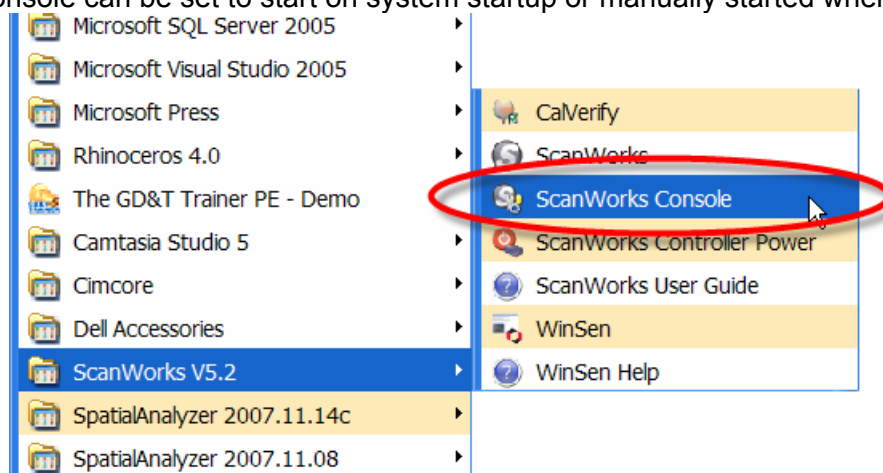
The Comparison Results Dialog presents the Pin Inspection results. Press



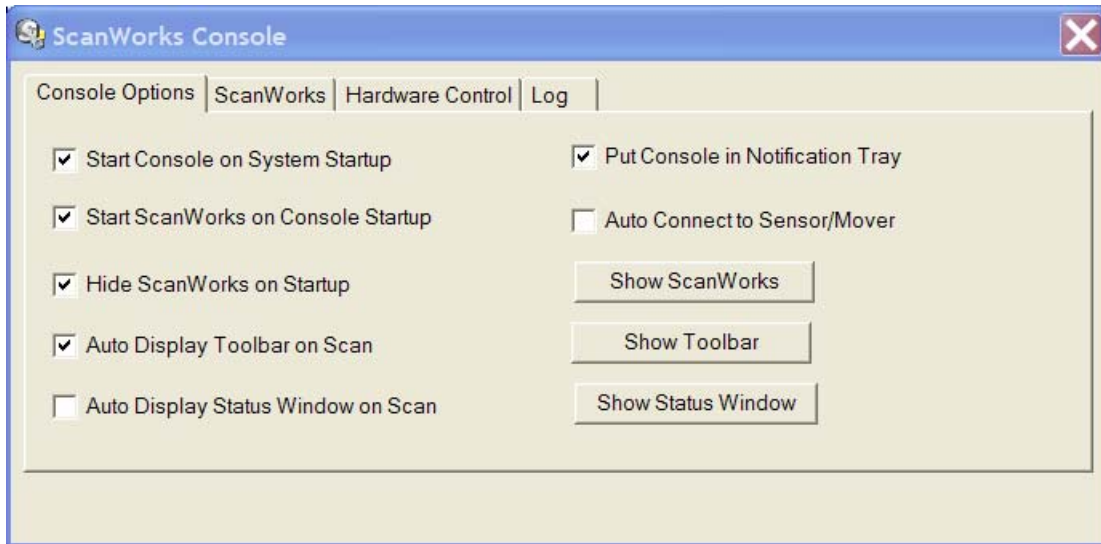
to close the Comparison Results and Pin Inspection Dialogs. The Hole and Plane inspections are performed in the same way as the Pin Inspection.

Running Perceptron V5 Scanner with Cimcore Arm in Spatial Analyzer

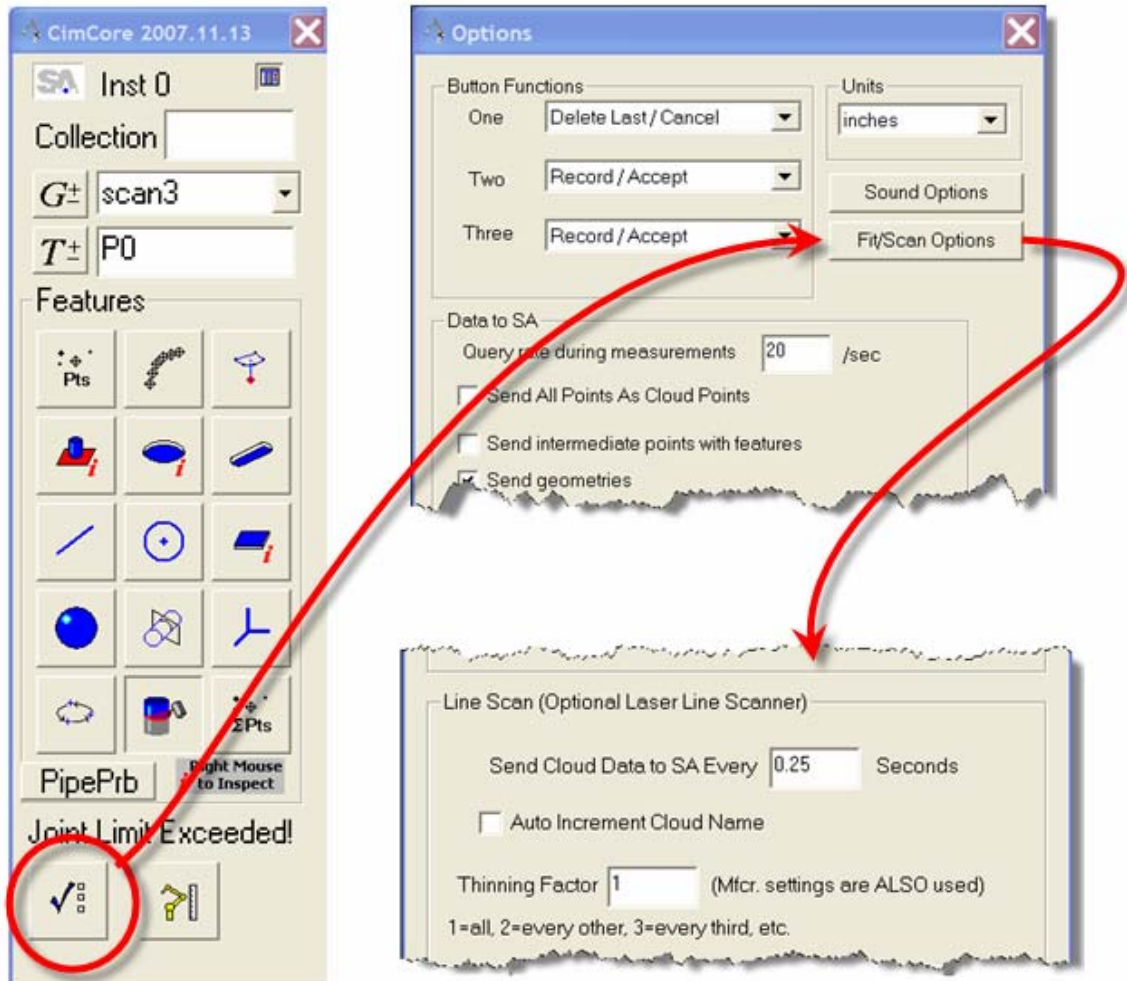
- Running the Perceptron Scanner requires Scanworks (version 5.21.116 or later) to be installed and running. Please refer to Scanworks documentation for details and settings. Scanworks interface is controlled by a console application. The console can be set to start on system startup or manually started when needed.




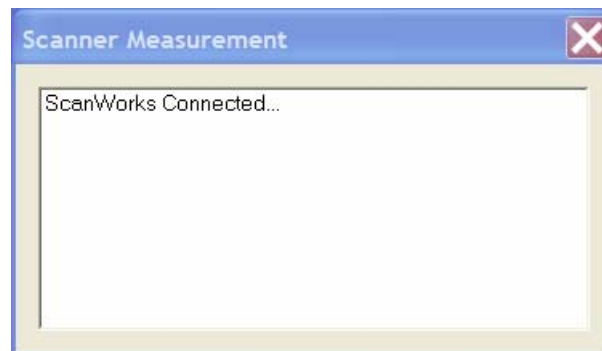
- The below options in the console must be set accordingly to allow SA to communicate with the arm and scanner. If settings are not set correctly communication errors will occur.



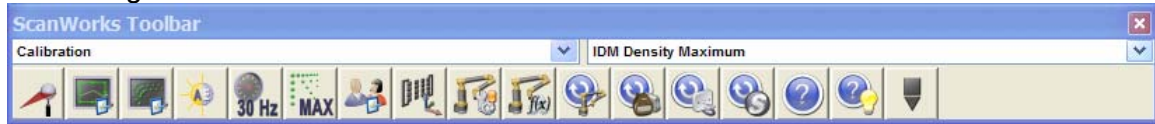
- Once the console options have been set, now operation of the scanner can begin. Open SA and add your instrument.
- Start the instrument interface. Best practice is to have the console running before you start the instrument interface but it is not required. The console must be running before the scanning operation is started.
- In the instrument interface there are several settings that can affect the outputs of the contour scanner. These settings can be optimized to best fit your application and the capacity of your computer.



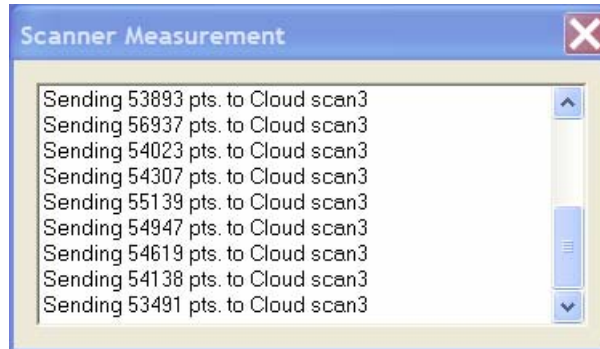
- Once Line Scan settings are configured, simply press  to start scanning. You will see a status window notifying the connection between SA and Scanworks.



- You will notice the below toolbar display when connected to the scanner. This will allow access to settings such as scan rate, scan density and exposure settings.



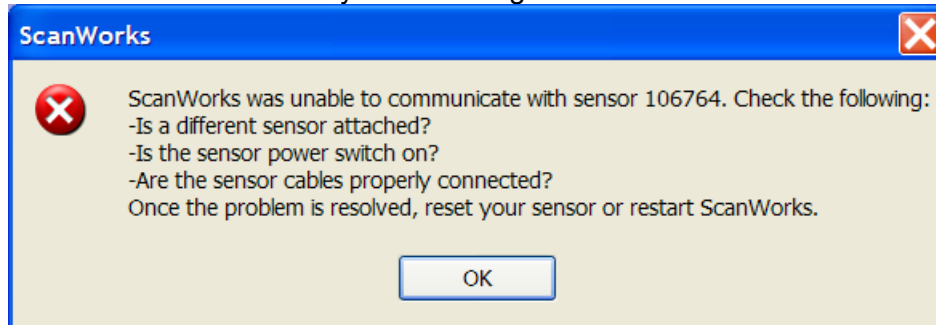
- Now press the Red button to start collecting data. The status window will show data being sent to SA.



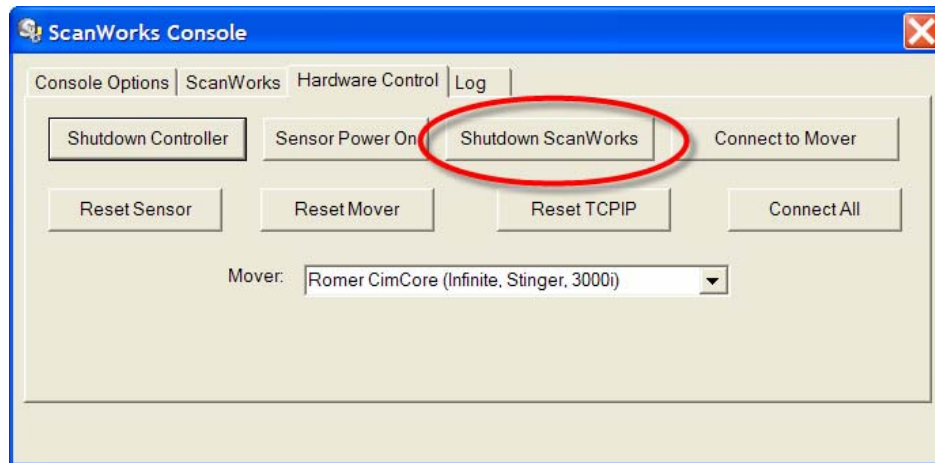
- To stop data collection, press the Red Button again. To exit the scanning operation press one of the White Buttons and SA will disconnect from Scanworks.

Troubleshooting

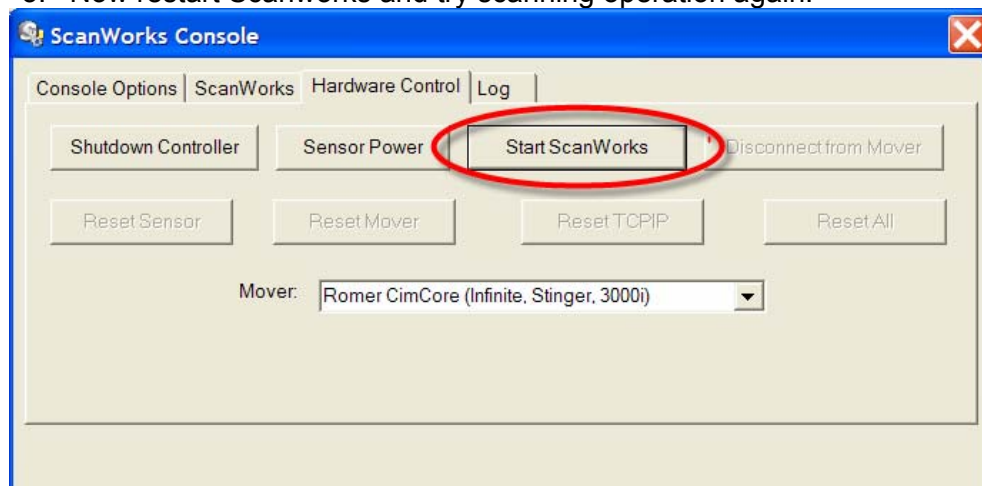
- If the below error occurs try the following:



1. Check to be sure sensor is attached and all connections are correct
2. Go to the Scanworks Console and close Scanworks



3. Now restart Scanworks and try scanning operation again.



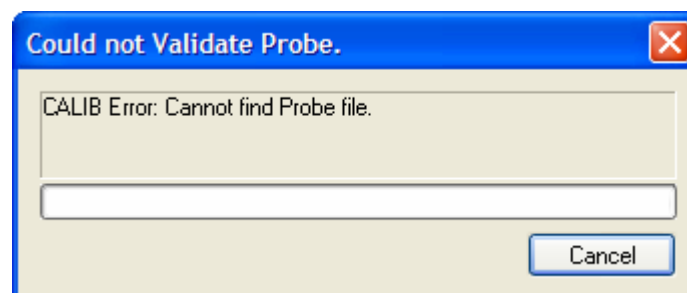
Scanner Settings:

You can edit the scan data handling and naming in the Arm Settings dialog. Press



to edit arm settings.

Note that if you still have the Perceptron scanner connected, you will briefly see an error message from CimCore:



This is expected, since the CimCore Validate Probe function is called, and CimCore has no knowledge of the Contour Scanner as a valid probe definition. Indeed, this error will be generated any time a measurement other than scanning is attempted while the scanner is connected to the arm.

The arm Options dialog will appear.

Options

Button Functions

One: Delete Last / Cancel

Two: Record / Accept

Three: Delete Last / Cancel

Units: inches

Sound Options

Fit/Scan Options

Query rate during measurements: 20 /sec

☐ Send intermediate points with features

☒ Send geometries

☒ Increment Group After Measurements

Increment Point Names by: 1

☒ Update SA Graphics When not measuring

Query rate for updates: 16 /sec

Probe

☒ Auto Detect Probe

Probe Number: 8 Apply

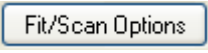
Material Thickness

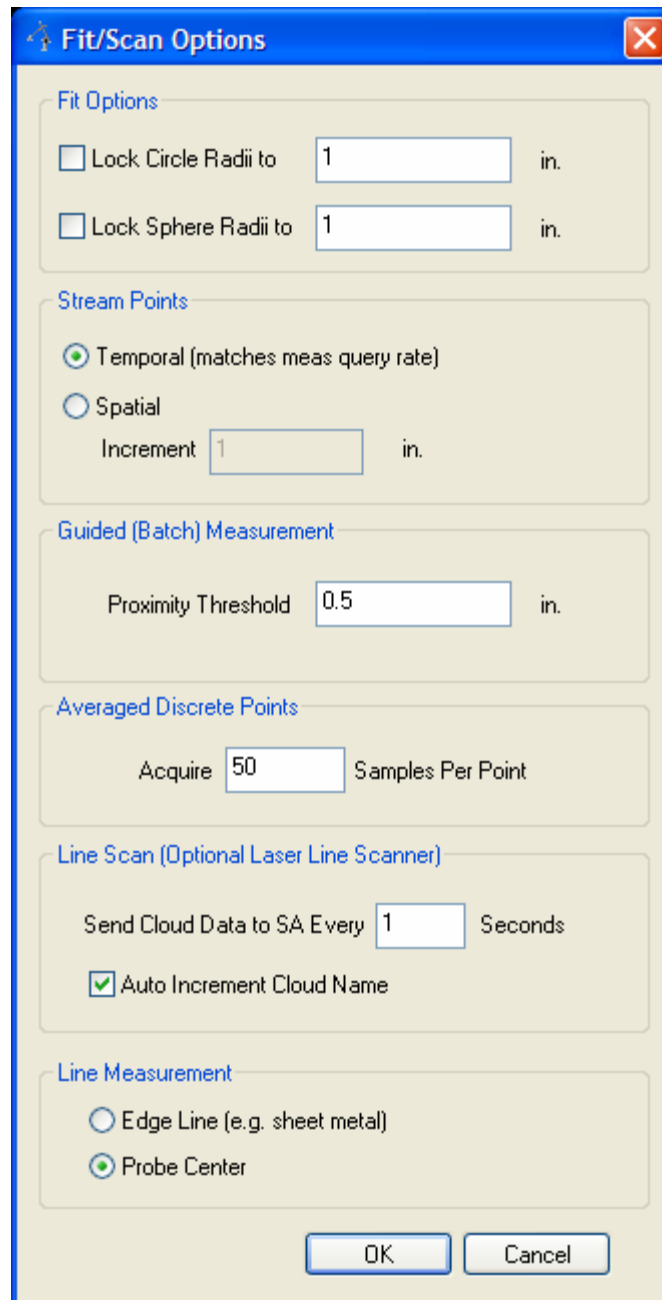
☐ Add Additional Offset to Patches

Offset: 0 in.

Change Language

OK Cancel

Press  to show the Fit/Scan Options dialog



The image shows a software dialog box titled "Fit/Scan Options" with a blue header bar and a red close button. The dialog is organized into several sections, each with a blue title:

- Fit Options:** Contains two checkboxes. "Lock Circle Radii to" is unchecked, with a text box containing "1" and "in." to its right. "Lock Sphere Radii to" is also unchecked, with a text box containing "1" and "in." to its right.
- Stream Points:** Contains two radio buttons. "Temporal (matches meas query rate)" is selected. "Spatial" is unselected. Below "Spatial" is a text box containing "1" and "in." labeled "Increment".
- Guided (Batch) Measurement:** Contains a text box containing "0.5" and "in." labeled "Proximity Threshold".
- Averaged Discrete Points:** Contains a text box containing "50" and "Samples Per Point" labeled "Acquire".
- Line Scan (Optional Laser Line Scanner):** Contains a text box containing "1" and "Seconds" labeled "Send Cloud Data to SA Every". Below this is a checked checkbox labeled "Auto Increment Cloud Name".
- Line Measurement:** Contains two radio buttons. "Edge Line (e.g. sheet metal)" is unselected. "Probe Center" is selected.

At the bottom of the dialog are two buttons: "OK" and "Cancel".

Under 'Line Scan (Optional Laser Line Scanner)', you can control the interval at which scan clouds are sent to SA (Perceptron Contour can acquire up to ~23kPts/sec), by editing the field labeled 'Send Cloud Data to SA Every ____ Seconds'. The Cloud Name is taken from the current Group Name in the main dialog. If 'Auto Increment Cloud Name' is checked, then the name will automatically increment after each cloud data packet is sent to SA. NOTE: The Scanner Measurement dialog is modeless, meaning the Group Name (and therefore Cloud Name) in the main dialog can be edited any time, even during a scan session. The scanner measurement progress shown earlier in this section was performed with the settings as shown here. To apply settings, simply press OK.

Scanner Calibration:

The ScanWorks User Guide includes detailed instructions for calibrating the scanner to the arm. Read the instructions through before attempting to calibrate the scanner.

If you need to change computers, but will be using the same scanner and arm, you can transfer the existing calibration to the new computer. The calibration file is stored in the ScanWorks install on the application pc. For example, if you used the default location for the ScanWorks install, the calibration file will be stored in

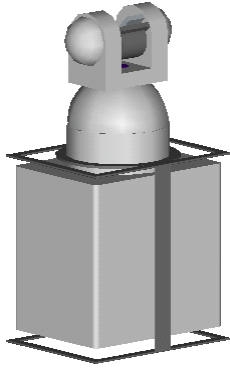
C:\Program Files\Perceptron\ScanWorks\CimCore\data\

The name of the file is

xform_[scanner serial number].dat

where [scanner serial number] will match the number printed directly on the scanner. Simply copy this file to the same location in the ScanWorks install folder on the new pc, and you are ready to run using the existing scanner calibration.

LASER SCANNER: METRIS CLR



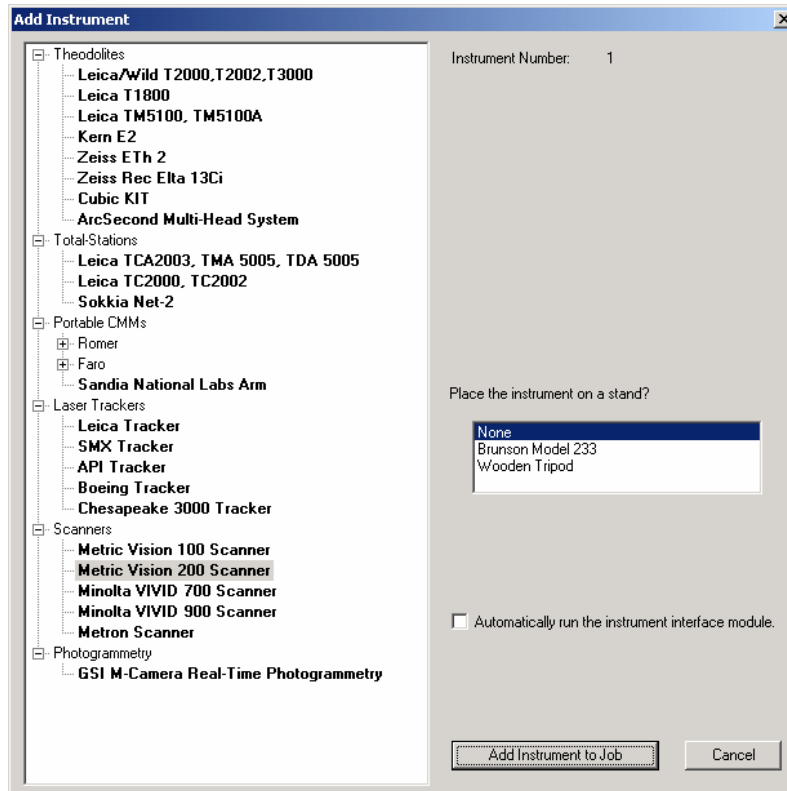
SAMVx.exe is the SpatialAnalyzer application used to interface to the Metris CLR family of scanners. SAMVx communicates to SpatialAnalyzer using New River Kinematics' TCP/IP socket protocol and communicated to the Metris scanner using Metris's CLRICx, ActiveX protocol.


To run SAMVx, you must first install the CLRICx control from Metris. This handles the low-level communication with the scanner controller. This is available at www.Metris.com.

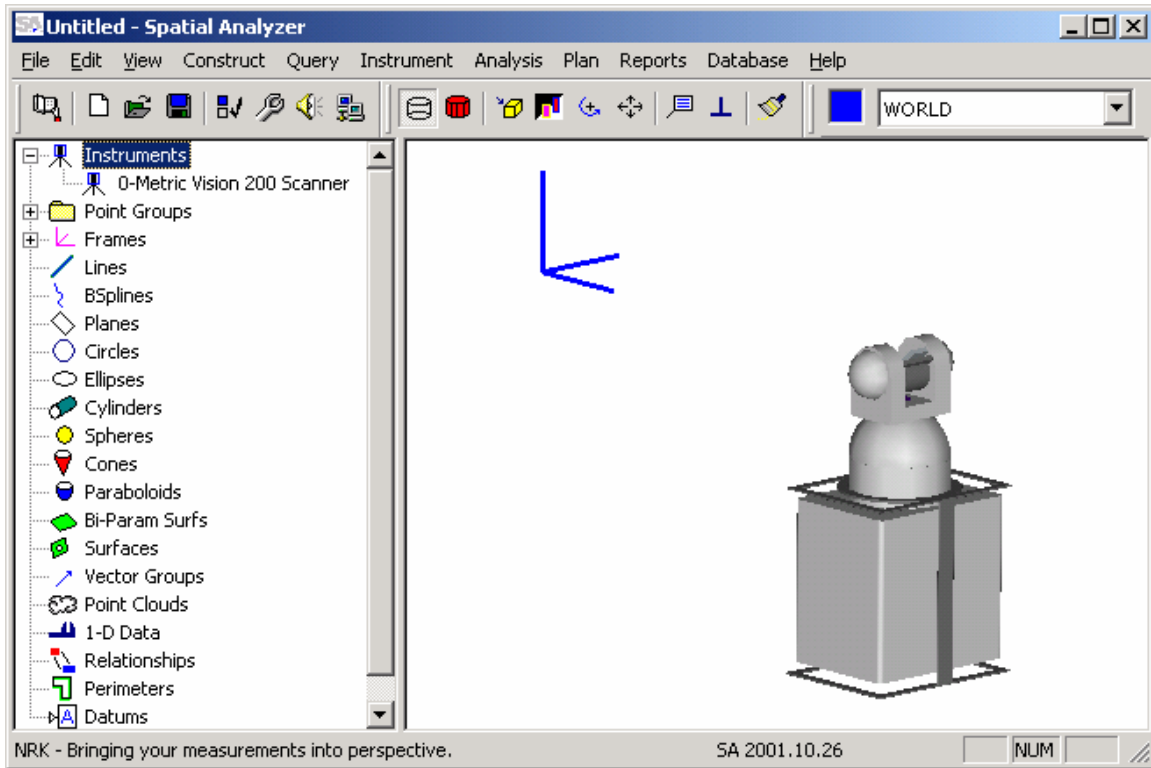
You must also have SpatialAnalyzer installed. The SA install includes the SAMVx application, so only one install is required.

Getting Started

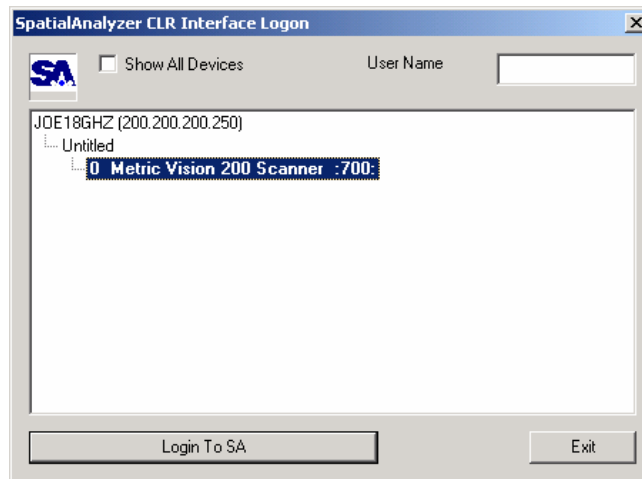
First, you will need to add the CLR instrument to SA. From the SA menu, select Instrument->Add. This will allow you to pick from the instrument list:



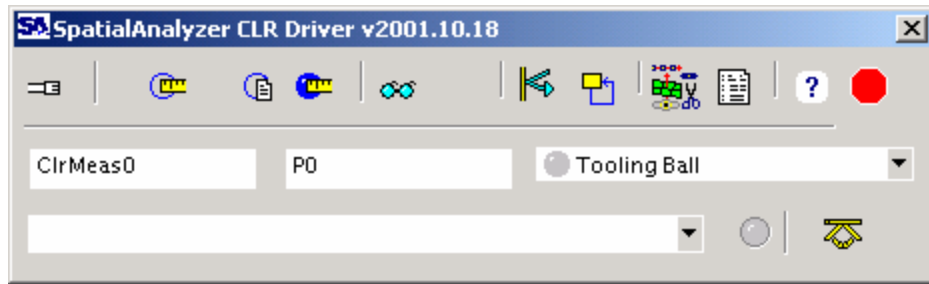
Select the “Metric 200 Scanner” and press . This will add the instrument to the job and the graphical model:




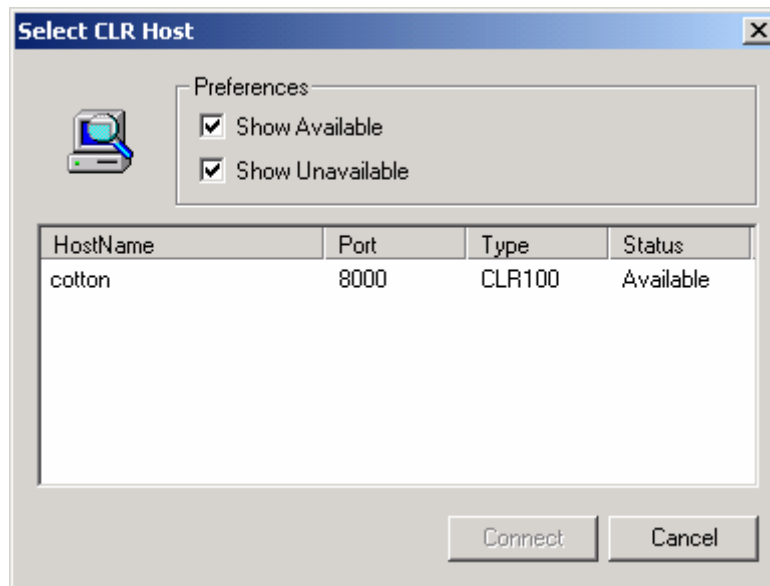
Next, run the Metris interface (SAMVx.exe) from the SA area of the Windows start bar. This will prompt you with the logon screen. Here, you select which SA job and which instrument in that job you want to connect to:




After making that selection, the SAMVx.exe interface dialog will appear:



The instrument interface is now connected to SA. The next step is to connect to the scanner itself. To do this, press  and the connection dialog will appear:



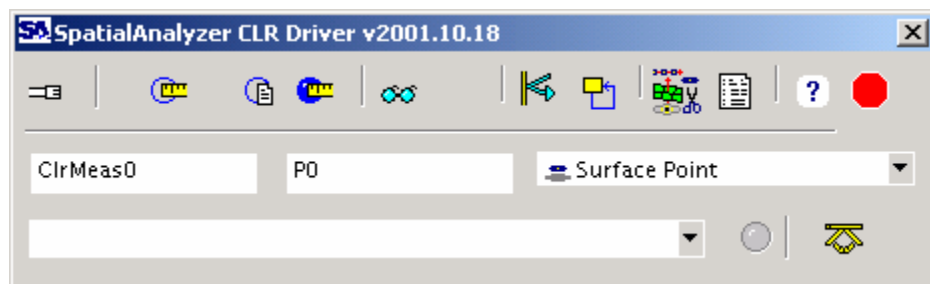
Select the scanner you wish to connect to, and press connect. The  button on the main interface will appear in a depressed state to indicate a connection has been made. You can also set a default connection in the CLRICx properties. This will automatically connect when you start the interface.


Taking Your First Measurement

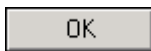
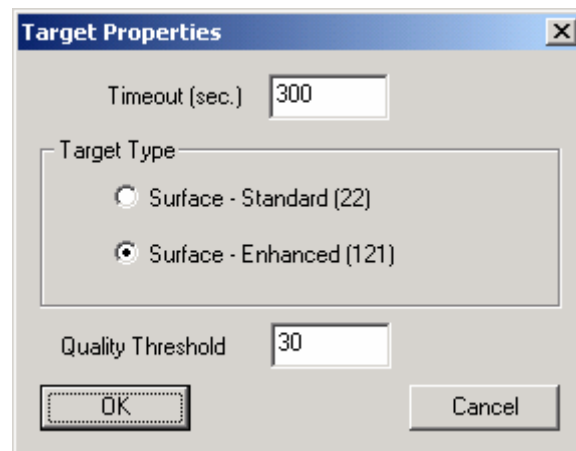
Select the target type you wish to measure from the combo-box on the right hand side of the dialog:

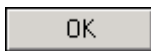



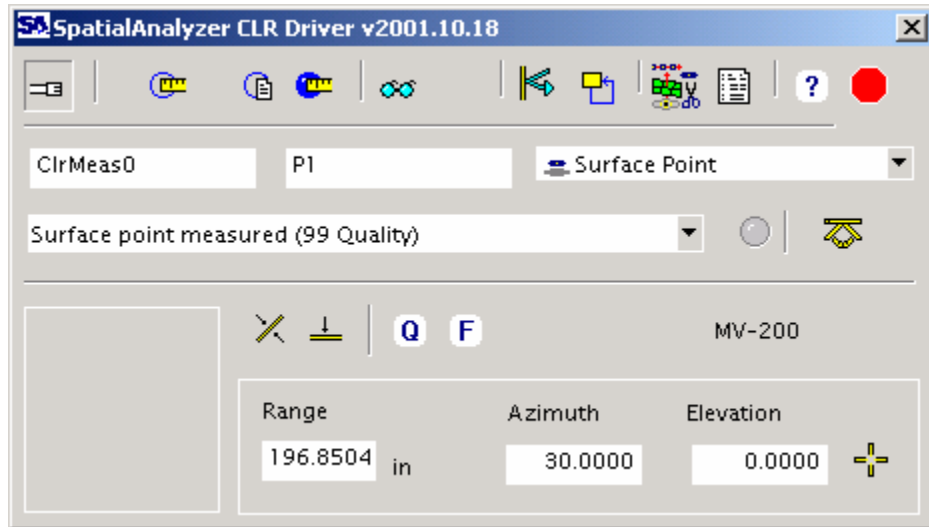
For this example, select surface point, . This will display the selected target in the combo back on the main dialog:



The next step is to verify the properties of the surface point target. To do this, press . This will bring up the target properties dialog for the current target:




Verify or change the settings and hit . At this point we are ready to measure. Now, we need to point to the surface area we are interested in. Press the dialog expansion button,  from the main dialog. This will show additional controls at the bottom of the main dialog:



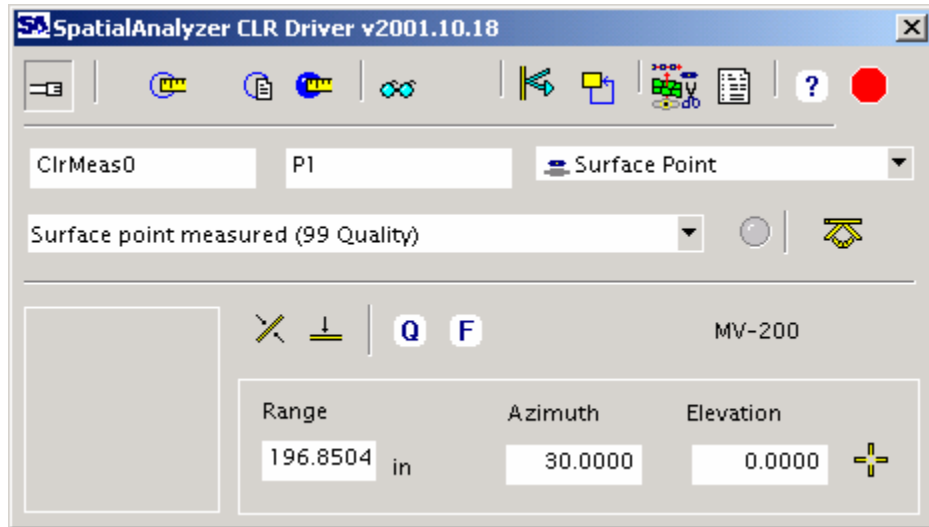
Move the mouse over the “NAV” window in the lower left side of the dialog. If you hold down the left button, you can steer the scanner around. You can also right click and access the CLRICx scanner menu. From there, select Video->Display. This will activate the video display interface. You will see the view from the camera mounted inside the scanner. Click on the video image to point at a certain location. You can also use a combination of SHIFT, CTRL, or SHIFT+CTRL and the arrow keys to steer the beam around.

Once you are located where you wish to measure, press F8 to autofocus the scanner.

Once complete, press the measure button,  to measure the surface point. The resulting confidence will be displayed in the status combo, and the measurement sent to SpatialAnalyzer.

Interface Description

In this section, we will describe the components of the SAMVx interface. This will begin with the main interface screen and then dive into the functional areas accordingly. First, let's begin with a graphical description of the sections of the main interface screen. The following figure will present the function of the various controls. In addition, you can hover the cursor over each button to see a description of its function.



Connection

This button displays the connections dialog (see Getting Started), and allows you to connect to the CLR. When it displays in the depressed state, you know that you are connected to the scanner.



Measure

Measure the current target type. Remember that you will need to autofocus before measuring for many of the target types.



Target Properties

Shows a dialog allowing you to set the properties for the current measurement mode.



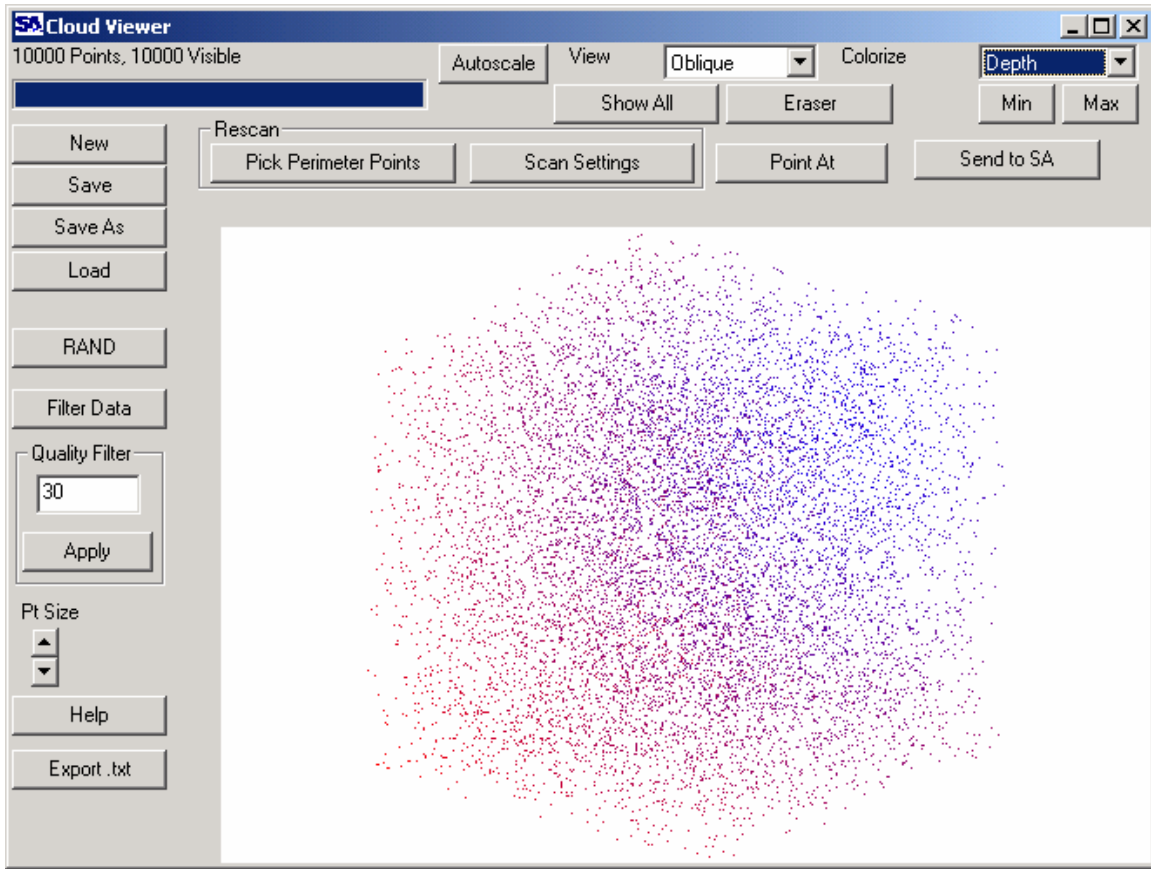
Inspect Target

For some target types (specifically Hole), you can use this option to prompt you to select an object in SA that you wish to inspect. For the case of Hole, SA will prompt you to pick a circle. Once you make the selection, SAMVx will automatically measure the hole (using the nominal geometry from SA), and compare the two results.



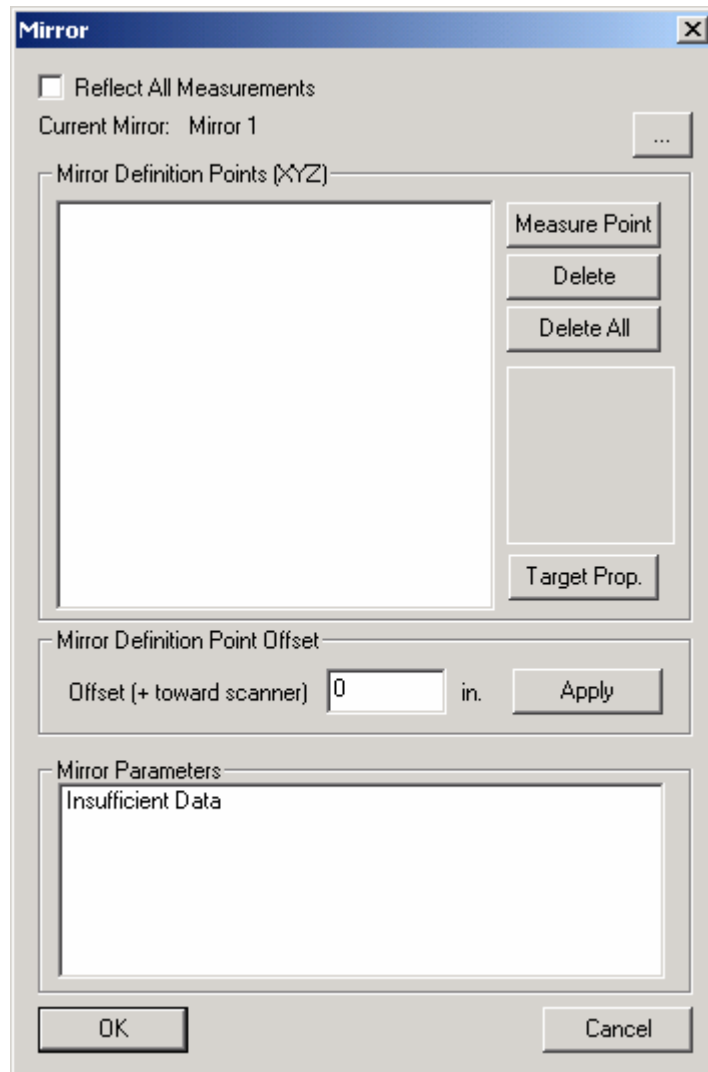
Cloud Viewer

Displays the interface for dealing with large clouds of data. The interface is shown below and will be discussed in more detail in later sections:



Reflections (Mirror)

The mirror function allows you to define a mirror that the CLR can use to measure through. This allows you to measure the backside of a part without moving the scanner. The first step in this process is to define the mirror by characterizing its transformation in space. Once this is determined, all measurements can be reflected through the mirror by turning the reflection mode on using the checkbox.



You can define many mirrors using the “...” button on the upper right corner of the window. Once you switch mirrors in that sub dialog, the current mirror will display in the Mirror dialog.

To define a mirror, you have 2 choices:


Use 2 points: Measure a tooling ball directly, and then measure the same tooling ball through the mirror. These 2 points will be used to back calculate the transformation of the mirror.

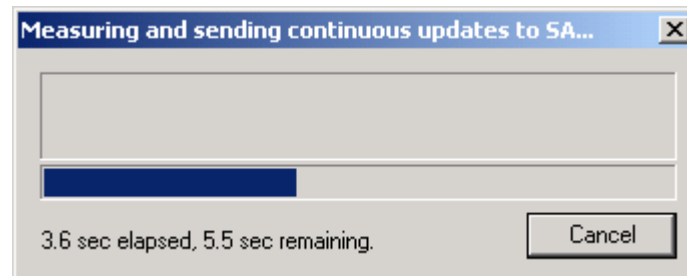
Use 3 or more points: In this method, you measure 3 or more points on (or offset from) the mirror. A plane fit is used to define the transformation of the mirror.



Measure Continuously (SA Watch)

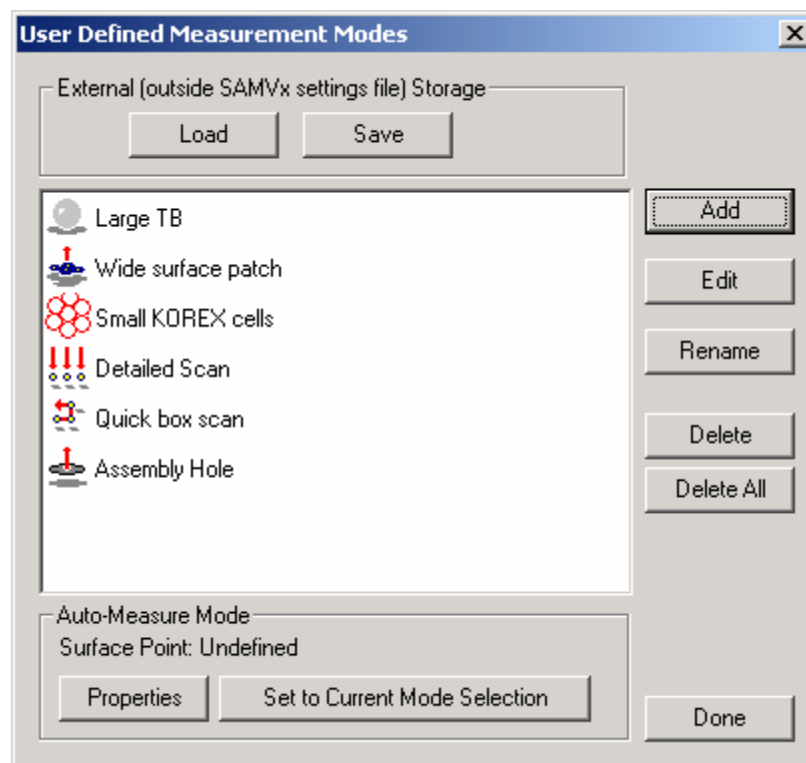
This mode repeatedly measures the current target type and sends the results as an update (not measurement) to SA. This allows you to use watch windows in SA to perform real-time analysis between points, or points and surfaces. You could, for example, track a tooling ball (if you move it slowly) using this option. While measuring, you will see a progress dialog that will update with each measurement. Press

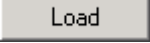

 to stop the measurement mode.



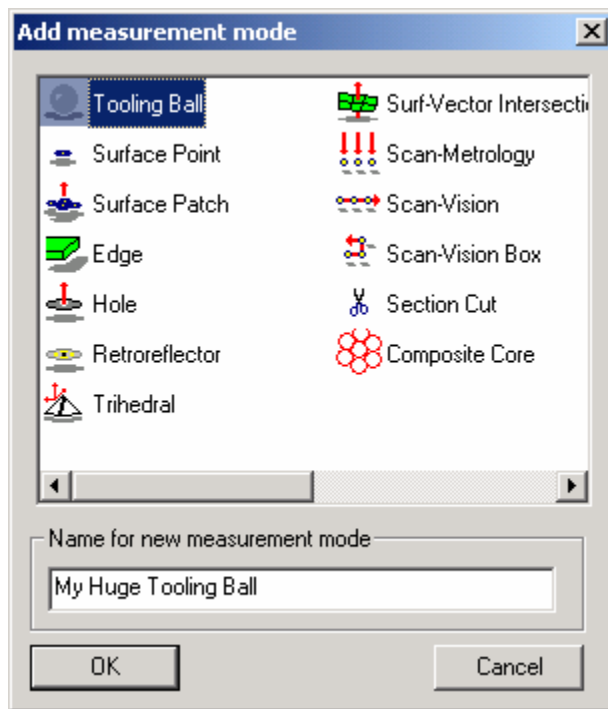
User Defined Measurement Modes

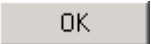
Provides access to the custom target database. This allows the user to configure measurement parameters that can be accessed easily in SA. Suppose you want to write a measurement plan to measure some tooling balls, then perform a quick box scan, then measure some large tooling balls. You will need to change the tooling ball target between scans, and also select parameters for the scan. This database allows you to configure all the measurement parameters ahead of time, then recall them by name later on.

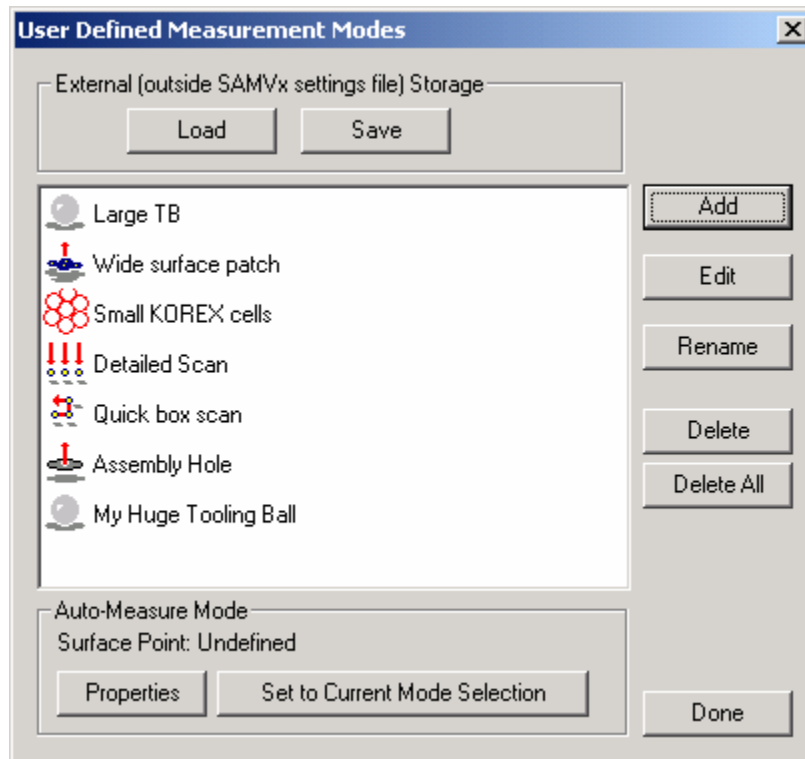


The  and  buttons allow you to manage the target type list independently of the other settings in the SAMVx settings file. This allows you to transport them with measurement plans.

To add a new custom target, press . This will prompt you to pick the target type and assign a name:



After pressing , you will be prompted to set the target properties for the selected target. In this case, a tooling ball. Once you do that, it will be added to the list:

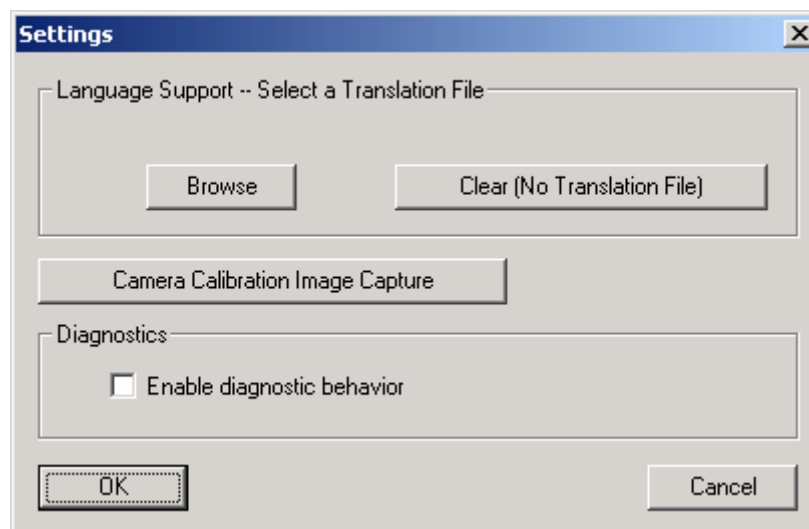


If you want to set the target that will be used by “Auto-Measure” mode in SA, highlight it, and hit **Set to Current Mode Selection**. This will change the default automeasure target. You can also use the SA measurement plan command “Set Instrument Measurement Mode” to specify the mode by name. This has the same effect.



System Settings

Allows you to set certain parameters for the interface:



Language Support: This option lets you specify a file containing the translation table for your language. Contact Metris for available translation files, or for information on how to create a new translation file. After changing the translation file, you will need to restart the instrument interface.

Camera Calibration Image Capture

This is a function performed at the factory to calibrate the video image to the scanner head.

Enable Diagnostic Behavior: If checked, certain functions perform differently, and provide more diagnostic information. Scans, for example, send lines to SA to indicate the proposed scanning pattern.



Stop

Sends a stop command to the CLR.

Surface Point

Target Type Selector

This combo allows you to pick the current target type from a list:



Once you select it, the combo in the main interface will change to reflect the selection.

Surface point measured (99 Quality)

Status Indicator

The status combo box provides a listing of status indicators from the SAMVx application. These messages will automatically roll off the list when a large quantity accumulate in the list.



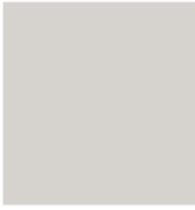
Status Light

Indicates Yellow when the CLR is busy, and Red when an error condition occurs. Click on this light in order to stop a pending process. This issues a stop command to the CLR immediately.



Dialog Expansion Control

Hides or shows the bottom portion of the dialog containing additional functionality.



Navigation Window

This window is the CLRICx interface provided by Metris. When you click and drag in this window, the scanner moves. If you right click, you will see Metris's menu for scanner functions. See the CLRICx documentation for a detailed description of this menu and its commands.



Flip Test

Initiates the CLRICx Flip Test using the following dialog. See the CLIRCx documentation for more information.

A screenshot of the 'Flip Test' dialog box. The dialog has a title bar 'Flip Test' with a close button. It contains a 'Navigation Window' on the left, which is a large gray rectangle. To the right of the navigation window are input fields for 'Front' and 'Back' parameters. The 'Front' section has fields for Range (5.00 m), Azimuth (0.35 deg), Elevation (-0.46 deg), and Confidence (0 dB). The 'Back' section has fields for Range (0.00 m), Azimuth (0.00 deg), Elevation (0.00 deg), and Confidence (0 dB). Below the 'Front' section is a 'Tooling ball radius' field set to 0.00 m. At the bottom right is a 'Difference' section with fields for Range (0.00 m), Azimuth (0.00 deg), and Elevation (0.00 deg). At the bottom center are 'Run Test' and 'Close' buttons.

Self Test

Initiates the CLRICx Self Test using the following dialog. See the CLIRCx documentation for more information.

Self Test

Status

Reference arm measurement: Complete

Mirror measurements:

Reference arm length

Delta of measurement and nominal value (m): 5.00

Quality (dB): 99

✓ Threshold: >= 98 (dB)

Mirror at -90 degrees elevation

Mirror offset delta (m): 0.00

Mirror offset standard deviation (m): 0.00

Mean quality (dB): 0

Threshold: < 10.00 (m)

Threshold: < 0.00 (m)

Threshold: >= 98 (dB)

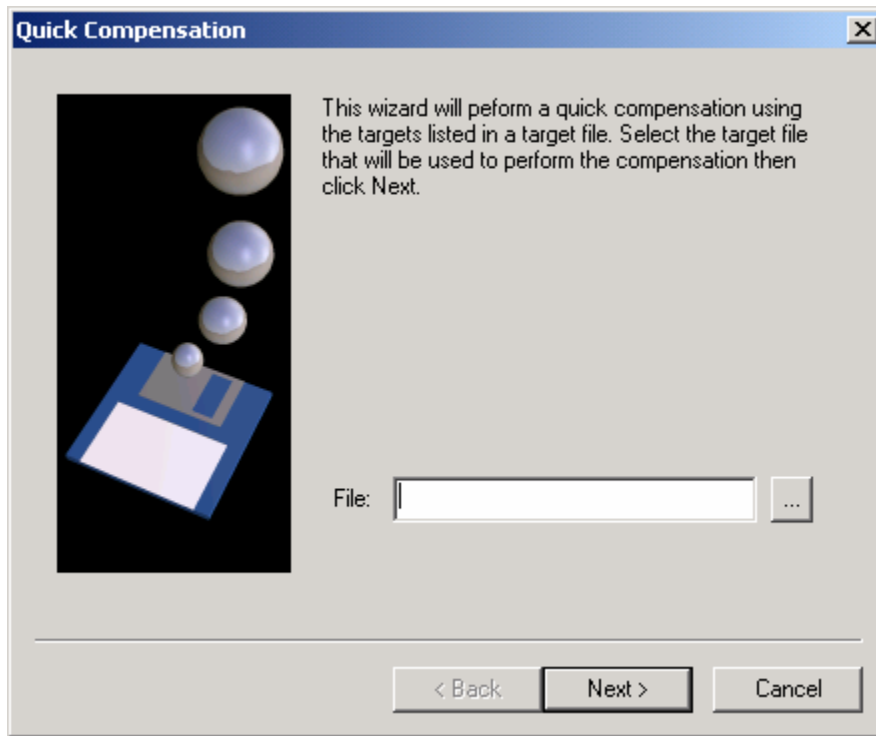
View Log

Cancel

Q

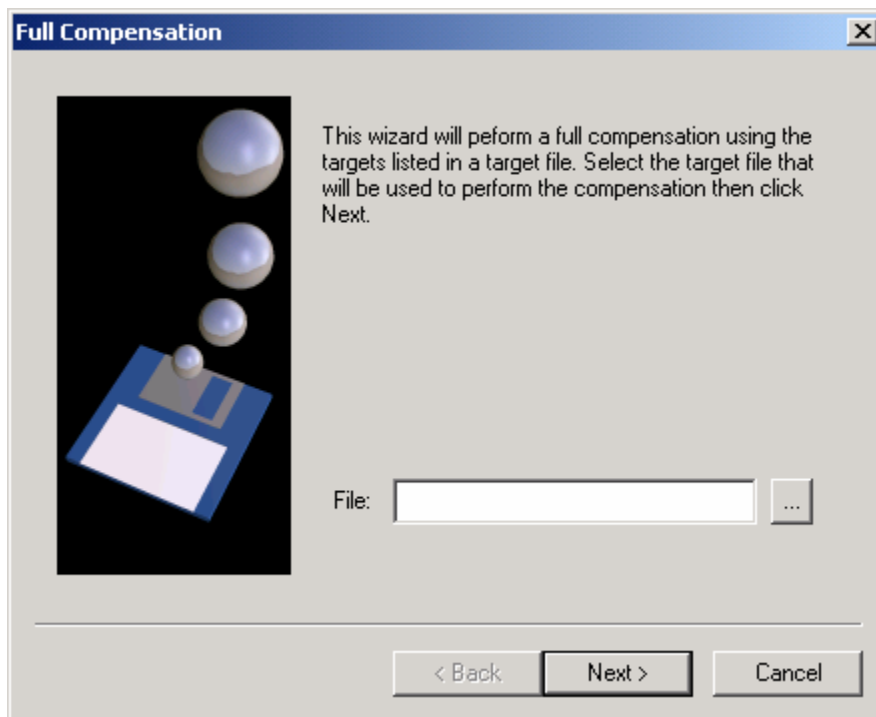
Quick Comp


Initiates the CLRICx Quick Comp using the following dialog. See the CLIRCx documentation for more information.




F Full Comp

Initiates the CLRICx Full Comp using the following dialog. See the CLIRCx documentation for more information.



Range	Azimuth	Elevation	
196.8504 in	30.0000	0.0000	

Current Range, Az, El

Displays the current values from the instrument. You can also edit these values then press  (Aim) to drive the scanner to the specified location.

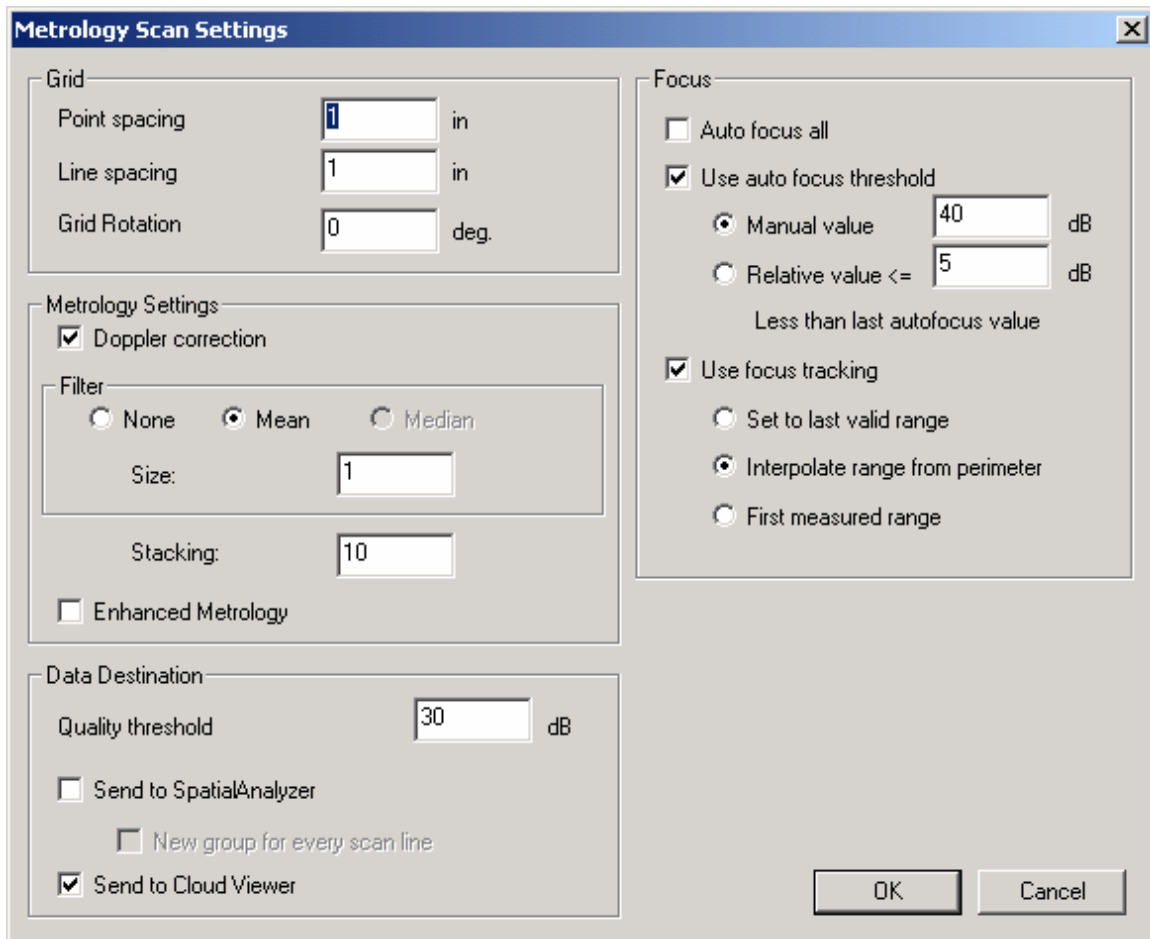
Scanning

There are two main scanning modes in the SAMVx interface: Metrology and Vision. The difference between the two is that the Metrology mode stops at each point and the Vision mode moves continuously through each scan line.

For both modes, there are several components necessary to perform a scan. The first is a perimeter to bound the measurement, and the second is a parameter set to define how the scanner operates within the boundary. In this section, we will present how you would run the scanner and initiate the scan manually from SAMVx. In the Measurement Plan section, we will explain how to do this in an automated fashion through MP.

In order to create a scan perimeter, measure several bounding points with the surface point target type. Then, go into SA, and select Construct->Perimeter. This will prompt you to pick the points that form the perimeter. Once you do this, a polygon will appear around the scan area. You will also be able to assign a name to the perimeter.

To create the parameter set, edit the properties of the scan modes in SAMVx. If you select Metrology scan as the current target type from the combo box, and press properties, you will see the metrology scan properties dialog:



The image shows a 'Metrology Scan Settings' dialog box with a blue title bar and a close button. It is divided into four main sections: Grid, Metrology Settings, Data Destination, and Focus.

- Grid:** Contains three input fields: 'Point spacing' (1 in), 'Line spacing' (1 in), and 'Grid Rotation' (0 deg.).
- Metrology Settings:** Includes a checked 'Doppler correction' checkbox. A 'Filter' section has radio buttons for 'None', 'Mean' (selected), and 'Median', with a 'Size' input field set to 1. Below is a 'Stacking' input field set to 10 and an unchecked 'Enhanced Metrology' checkbox.
- Data Destination:** Features a 'Quality threshold' input field set to 30 dB. It has two unchecked checkboxes: 'Send to SpatialAnalyzer' and 'New group for every scan line'. It also has a checked 'Send to Cloud Viewer' checkbox.
- Focus:** Contains an unchecked 'Auto focus all' checkbox. A checked 'Use auto focus threshold' checkbox is followed by radio buttons for 'Manual value' (40 dB, selected) and 'Relative value <= 5 dB'. Below this is the text 'Less than last autofocus value'. A checked 'Use focus tracking' checkbox is followed by radio buttons for 'Set to last valid range', 'Interpolate range from perimeter' (selected), and 'First measured range'.

At the bottom right are 'OK' and 'Cancel' buttons.

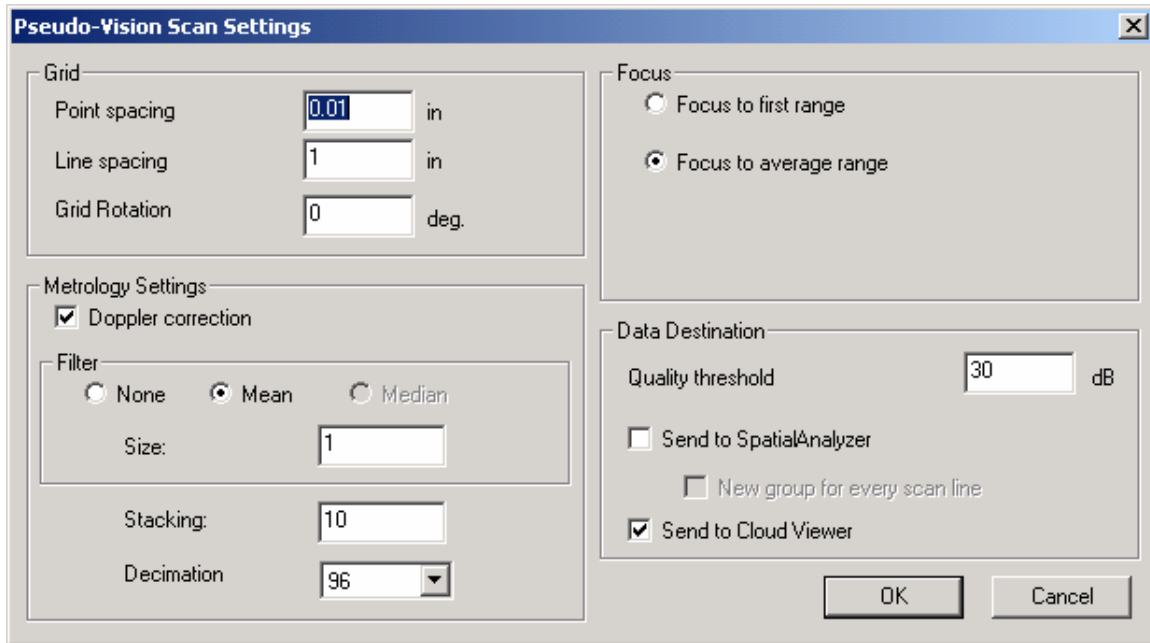
Grid – this area defines the layout of the scan points within the scan boundary.

Metrology Settings – settings specific to the CLR measurement mode.

Data Destination – allows you to control where the measured data is sent and what level of quality is required to keep a point.

Focus – controls the focus operation of the scanner.

For Vision scans, a similar dialog will appear:



Pseudo-Vision Scan Settings

Grid

Point spacing: 0.01 in

Line spacing: 1 in

Grid Rotation: 0 deg.

Metrology Settings

☒ Doppler correction

Filter

☐ None ☒ Mean ☐ Median

Size: 1

Stacking: 10

Decimation: 96

Focus

☐ Focus to first range

☒ Focus to average range

Data Destination

Quality threshold: 30 dB

☐ Send to SpatialAnalyzer


☐ New group for every scan line

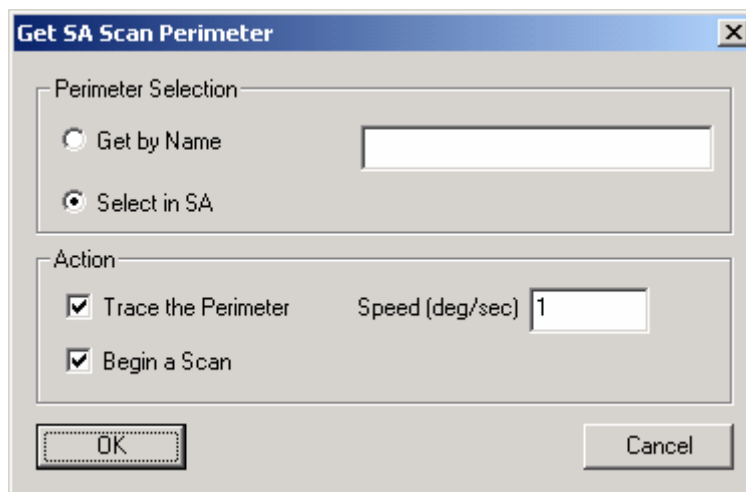
☒ Send to Cloud Viewer

OK Cancel

The primary difference is the focus control which is more limited since the scan is fluid as opposed to discrete.

Once you have the parameters setup and a perimeter created, you are ready to scan.

Press  to begin the scan. You will be asked to select the perimeter you want to use:



Get SA Scan Perimeter

Perimeter Selection

☐ Get by Name

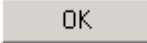
☒ Select in SA

Action

☒ Trace the Perimeter Speed (deg/sec) 1

☒ Begin a Scan

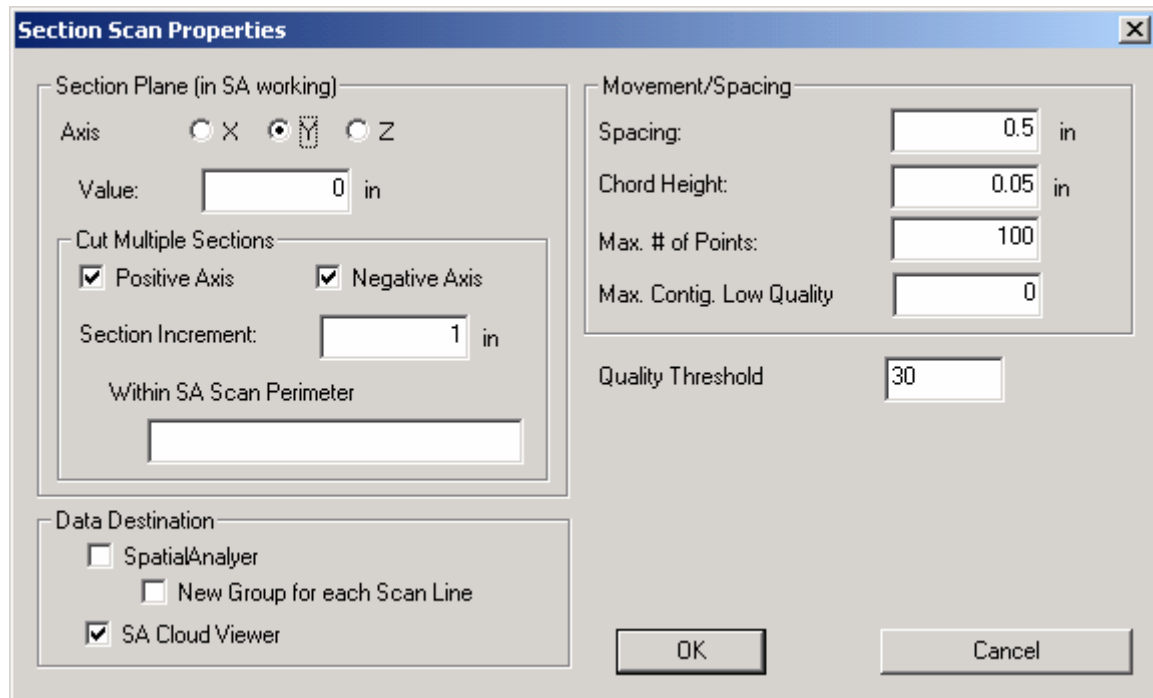
OK Cancel

You can either select the perimeter in SA, or specify it by name. Also, you can tell the CLR to trace out the perimeter to verify it before the scan. Once you press , the scan will begin, and a progress bar will appear tracking the total progress of the scan.

Special Scan modes

The **Vision Box Scan** mode does not require a perimeter. It simply uses the current scanner position (or a point in SA) to seed a rectangle (normal to the scanner) for a simple scan. Its parameters are set similar to those for a normal bounded vision scan. You also need to input the width and height of the box to be scanned.

The **Section Cut** scan uses a special scanning mode to hunt for the location where the measured data corresponds with section planes in the CAD model. Its properties are as follows:



The screenshot shows the 'Section Scan Properties' dialog box with the following settings:

- Section Plane (in SA working)**
 - Axis: ☐ X, ☒ Y, ☐ Z
 - Value: in
- Cut Multiple Sections**
 - ☒ Positive Axis, ☒ Negative Axis
 - Section Increment: in
 - Within SA Scan Perimeter:
- Data Destination**
 - ☐ SpatialAnalyzer
 - ☐ New Group for each Scan Line
 - ☒ SA Cloud Viewer
- Movement/Spacing**
 - Spacing: in
 - Chord Height: in
 - Max. # of Points:
 - Max. Contig. Low Quality:
 - Quality Threshold:

Buttons: OK, Cancel

Section Plane – in this section, you specify the plane (in the SA working frame) that you wish to cut. You also specify the starting plane value for the first cut. There are also options for cutting multiple sections given an increment. Notice you need to supply a scan perimeter as well. This provides the bounds for the section cuts.

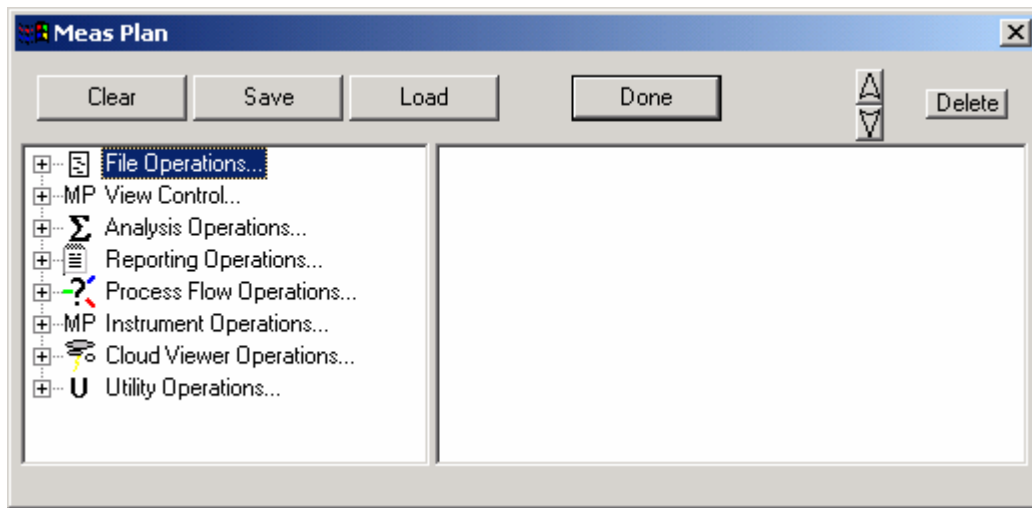
Data Destination – allows you to determine where to send the data.

Movement / Spacing – parameters specific to the CLR section cut scan mode.

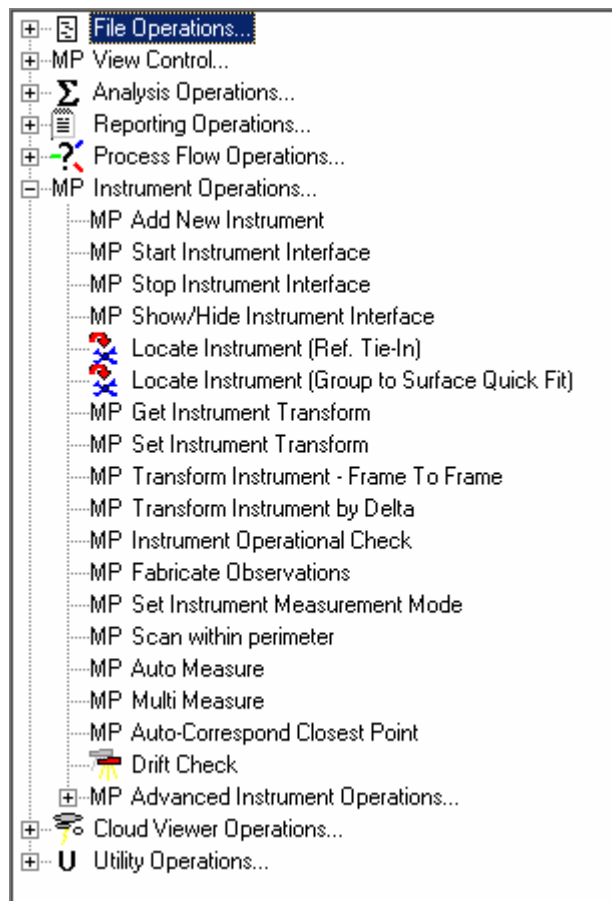
SpatialAnalyzer Measurement Plan Functionality

In this section, we will present the components of SA's measurement plan that apply specifically to the MV scanner. For basic information on creating, executing, and editing measurement plans, see the Measurement Plan chapter of the SA manual. This section will assume you know how to create measurement plans and will present the most pertinent MP commands for the CLR scanner.

In SA, select File->Create/Edit Measurement Plan. This will pull up the measurement plan window with the available commands on the left side:



Instrument Operations



All of the commands in this area apply to the CLR scanner. Many of them perform basic functions that are self explanatory. There are several, however, that are most useful with the CLR:

Scan Within Perimeter

The screenshot shows the 'Step Properties' dialog box with the title bar 'Step Properties' and a close button. The 'Step Type' is 'Scan within perimeter'. The 'Step Title' field contains 'Scan within perimeter'. Below this is a table titled 'Arguments' with the following data:

Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument to scan	Enter Value	0
1	Perimeter Name	Scan perimeter name	Enter Value	
2	String	Parameter set name	Enter Value	
3	Group Name	Group name	Enter Value	

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Indicate the instrument to scan, then specify a perimeter name. Next, pick the parameter set name. This is the name of the user defined target type in SAMVx. You can also specify a group name to contain the results of the scan. Executing this step will begin a scan immediately and move to the next step only when the scan is completed.

Set Instrument Measurement Mode

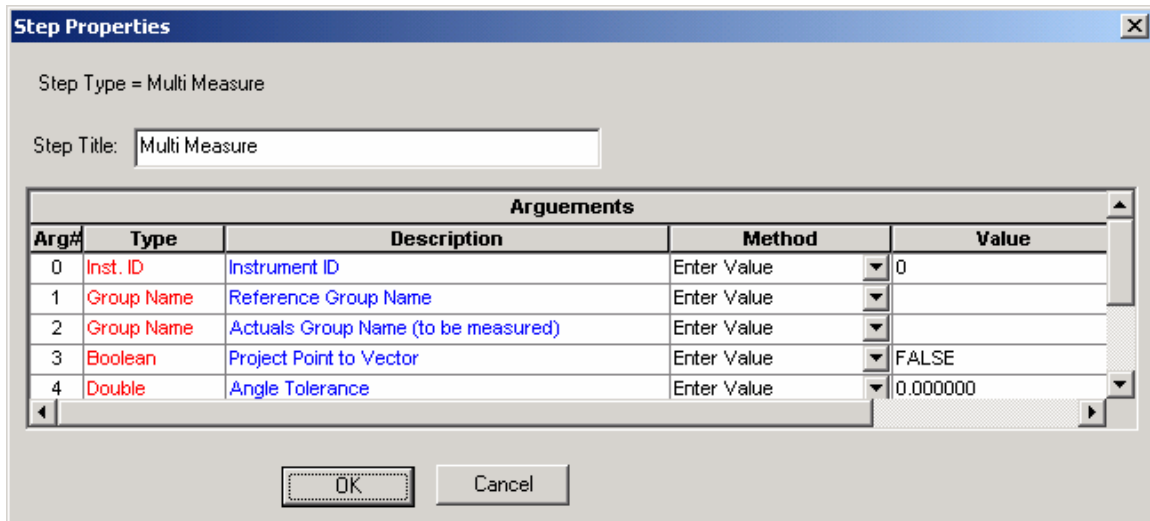
The screenshot shows the 'Step Properties' dialog box with the title bar 'Step Properties' and a close button. The 'Step Type' is 'Set Instrument Measurement Mode'. The 'Step Title' field contains 'Set Instrument Measurement Mode'. Below this is a table titled 'Arguments' with the following data:

Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument to set	Enter Value	0
1	String	Mode	Enter Value	

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Sets the “automeasure target type” in SAMVx. This is done by entering a name for the parameter set previously defined in the custom target type area of SAMVx.

Auto-Measure Vectors (multi-measure)



The dialog box titled "Step Properties" shows the configuration for a "Multi Measure" step. It includes a "Step Title" field and a table of arguments.

Step Type = Multi Measure

Step Title: Multi Measure

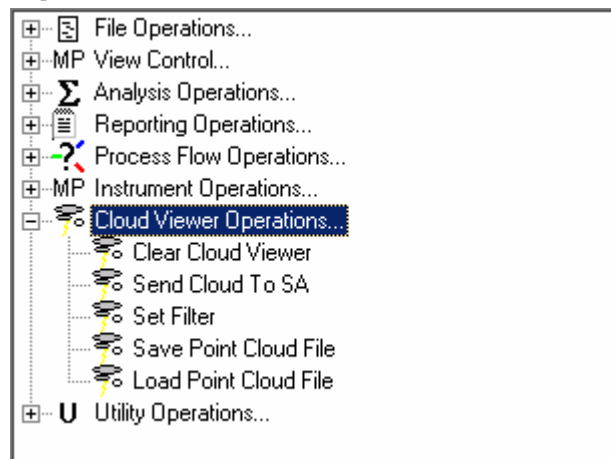
Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument ID	Enter Value	0
1	Group Name	Reference Group Name	Enter Value	
2	Group Name	Actuals Group Name (to be measured)	Enter Value	
3	Boolean	Project Point to Vector	Enter Value	FALSE
4	Double	Angle Tolerance	Enter Value	0.000000

Buttons: OK, Cancel

This initiates the vector measurement mode in SAMVx, or Surface Vector Intersection. You supply a Vector Group and SAMVx automatically uses the default SVI target (or a specific one if you set the measurement mode first).

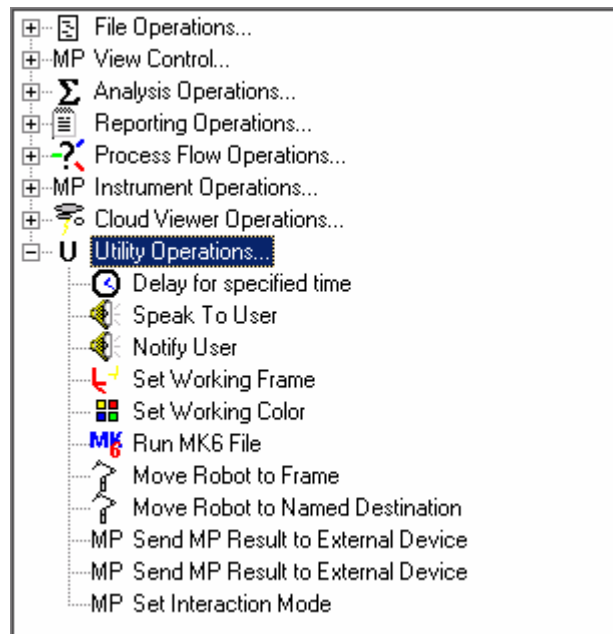
There are many other useful functions in the Instrument Operations area. These include Drift Check, Instrument Location, etc.

Cloud Viewer Operations



These modes allow you to control the cloud view component of the SAMVx interface from the SA measurement plan. This can be useful for clearing information, saving clouds, and sending them to SA.

Utility Operations

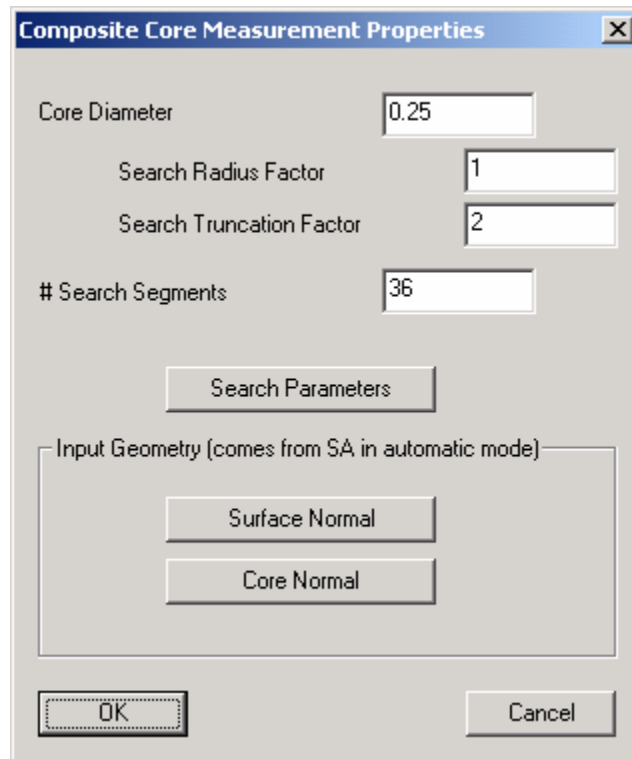


The robot control commands may be used with the Fanuc robot (or other robot) interface developed for positioning the scanner in a workcell. The details of that interface and the MP commands are described in more detail in the robot control interface manual.

Set interaction mode allows you to control the level of user prompting given when an MP encounters errors. It also controls whether measurement modes like drift check start measuring immediately when the MP command executes.

Composite Core Measurement

The composite core target type is designed for measuring the “honeycomb” structure of composite material. To use this mode, select the Composite Core target type. Then edit its properties:



Core Diameter is the largest enclosing diameter for an individual cell. Often, these cells are hexagons so this diameter is the diameter around the vertices.

Search Radius Factor is multiplied by the diameter to determine the search radius for the algorithm. A factor of 1.0 results in a search that covers from the center of the cell outward a distance of the diameter. This means it would theoretically cross the core edge halfway along that path.

Search Truncation Factor is also multiplied by the diameter and is used to limit the length of the search segments. The search pattern is based on the relationship between the core normal direction and the surface normal direction. Depending on this relationship, there could be a significant elongation of the scan pattern. This factor limits the search so that extreme elongation will not cause false edge detection. A value of 2 indicates that truncation will begin when the angle between the two vectors reaches 60 degrees.

Search Segments is the number of searches that will be made for the cell extents.

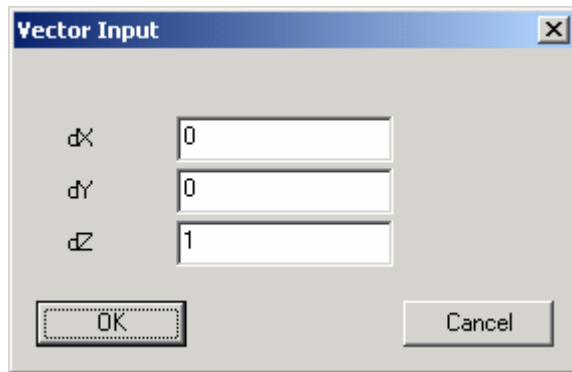
Search Parameters Allows you to set the parameters of the vision scan used in the search process. These parameters are presented in the standard vision scan dialog:

The line spacing and grid rotation fields are ignored, since these settings are used for the individual line scans only.

The send data options are normally off since the measurement mode will send the analyzed points only to SA. If you turn these on, you will see all the data used in the

analysis. This may be useful for determining how much noise is in the data and changing the settings to reduce that noise.

Surface Normal allows you to manually set the normal representing the surface cut on the composite core structure. This will bring up another dialog where you can enter the vector. The vector is relative to the **SA WORKING FRAME**.

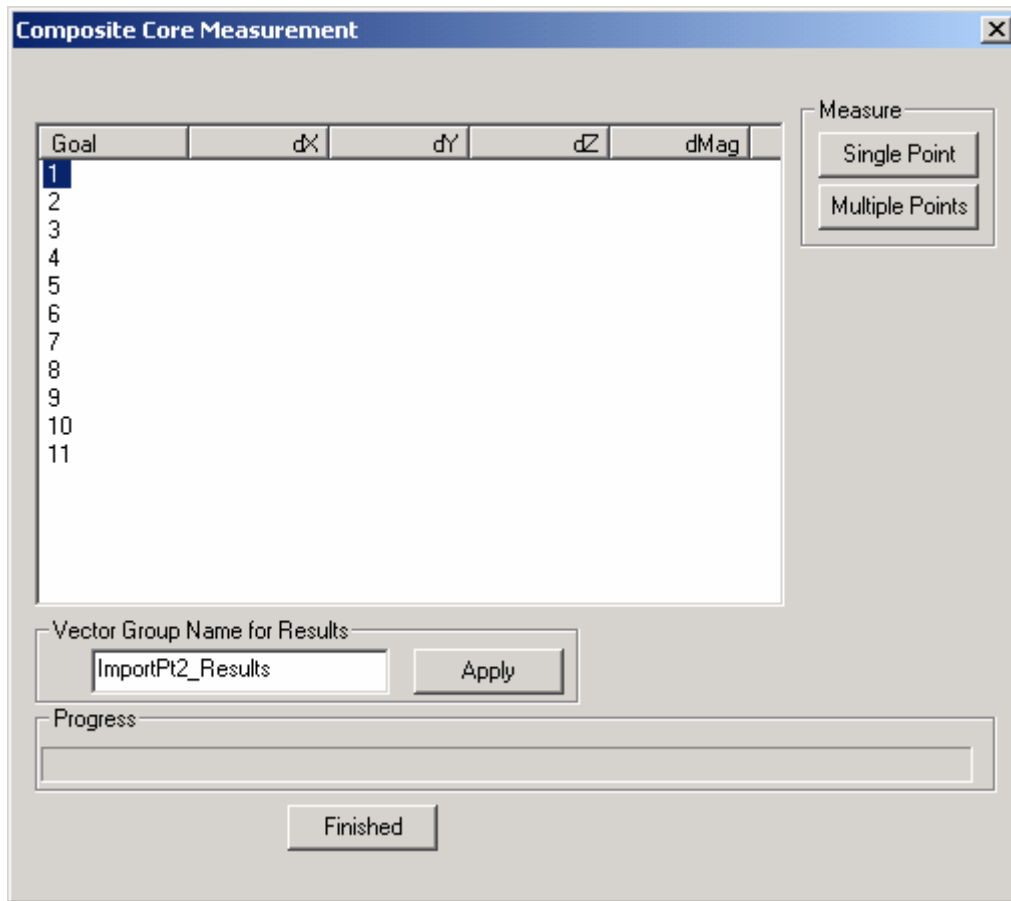


Core Normal allows you to set the normal vector representing the orientation of the core cell. Its input dialog is identical to that of the Surface Normal.

Auto-Measure Composite Core Mode in SA

To start the routine from the menu select Instrument->Automatic Measurement->Auto Measure Composite Core Material. You will be prompted to select an instrument (if more than 1 is present), a vector group and multiple surfaces. The vectorgroup should be generated such that the beginning points of the vectors lie exactly on the design surface to be inspected with the vector directions indicating the core axis orientation. As a practical note, in many cases the Core material axis is nearly coincident with the surface normal. For these cases you can easily construct the vector group by constructing a vector group by clicking on surfaces.

After starting the mode you will see the Core Measurement dialog shown below:



The goal points listed indicate the vectors (targets) to be measured. You can highlight any point to measure and hit the Measure Single or Multiple buttons to begin sampling. Depending on the target settings in SAMVx the measurement of a single target will take between 20 seconds and 3 minutes. When data is returned to SA after each measurement a value is computed that represents the standard deviation of the sampled data relative to the design surface. This standard deviation is then multiplied by the sigma value specified in the SA User Options, analysis tab to represent the confidence interval of the sampled data. For example if a sigma value of 2 is specified, the resulting dMag value represents a volume enclosing 95.46% of the core material surface samples. As you measure successive points the values will be updated.

Composite Core Measurement

Goal	dX	dY	dZ	dMag
1	-0.0006	-0.0042	0.0003	0.0043
2	-0.0000	-0.0040	0.0003	0.0040
3				
4	0.0006	-0.0035	0.0003	0.0036
5				
6				
7				
8				
9				
10				
11				

Measure

Single Point

Multiple Points

Vector Group Name for Results

ImportPt2_Results

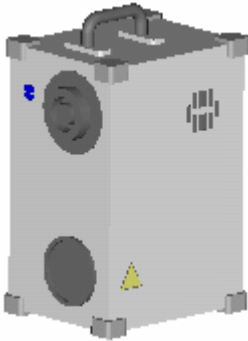
Apply

Progress

Finished

At any time you may change the name for the resulting vector group. This essentially resets the interface to take another set of samples and store results in a different location.

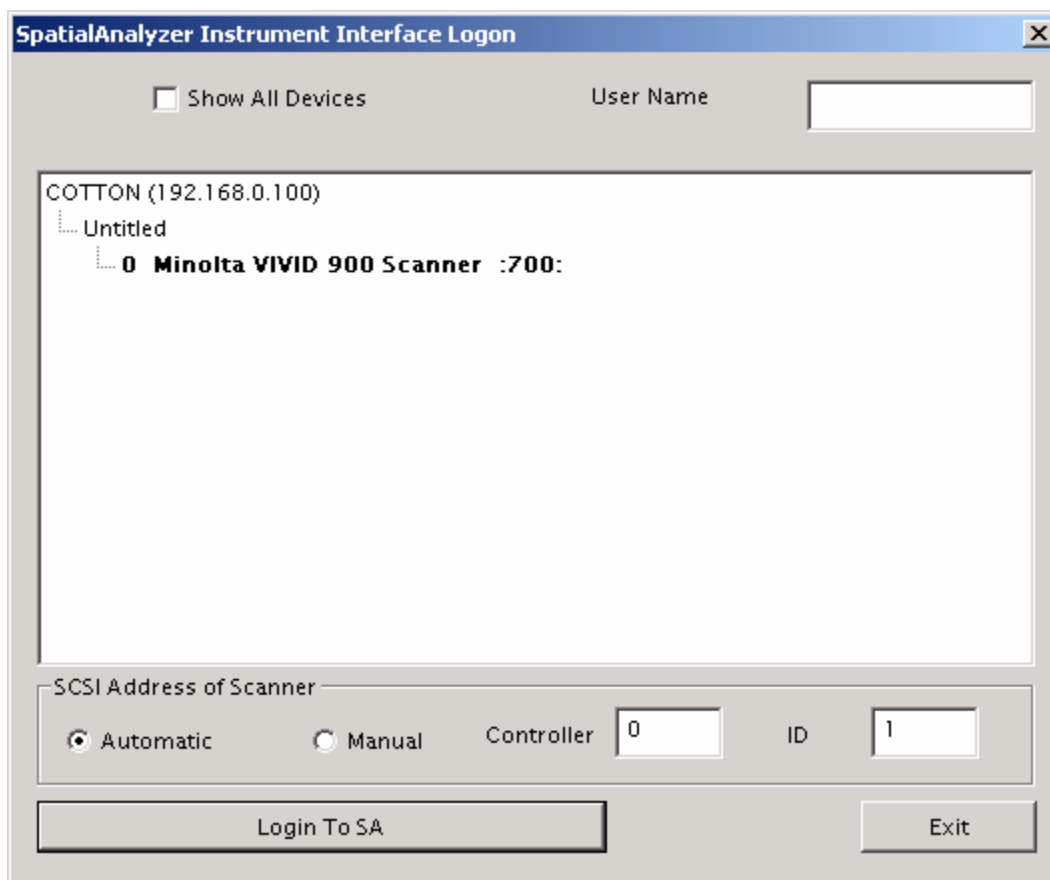
LASER SCANNER: MINOLTA VIVID



The program, Minolta 900.exe is used to interface SA to the VIVID 900 scanner. This module communicates with the scanner directly using a SCSI interface. In order to use the VIVID on your computer, you must have the SCSI layer properly installed. This is usually accomplished automatically when you add a SCSI card to your machine, but it may be necessary to modify the drivers or the system SCSI layer depending on your configuration.

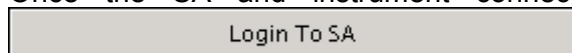
Once the SCSI communication is properly installed, connect the SCSI cable from your computer to the scanner. Then, turn the scanner on. It will go through a brief power-on phase and after approximately 30 seconds it will be ready to measure.

Next, add an instrument to SA. In this case, the VIVID 900. Then, run the Minolta interface from the SA start bar. It will show you the connection screen:

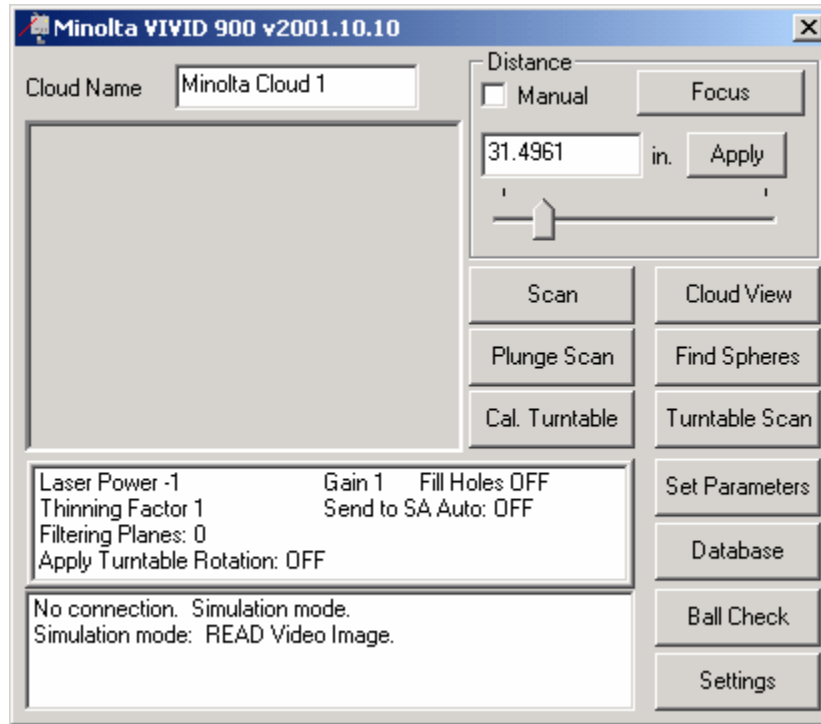


As with other SA interfaces, you can select the scanner you wish to connect to. You can also specify the SCSI addressing mode you want to use. If you just have one scanner on the SCSI chain, select automatic. If, however, you have several scanners on the same SCSI chain, or spread across several SCSI adapters, you will need to select the Manual addressing mode. For manual, you must specify the Controller number and the SCSI ID number. For more information on these parameters, see the Minolta scanner documentation. Basically, the controller identifies the card in the computer, and the ID is the SCSI id that you set from the back panel of the VIVID 900. The proper value for controller can vary according to machine configuration and operating system.

Once the SA and instrument connection selections have been made, press



This will connect to the instrument, then start the interface:



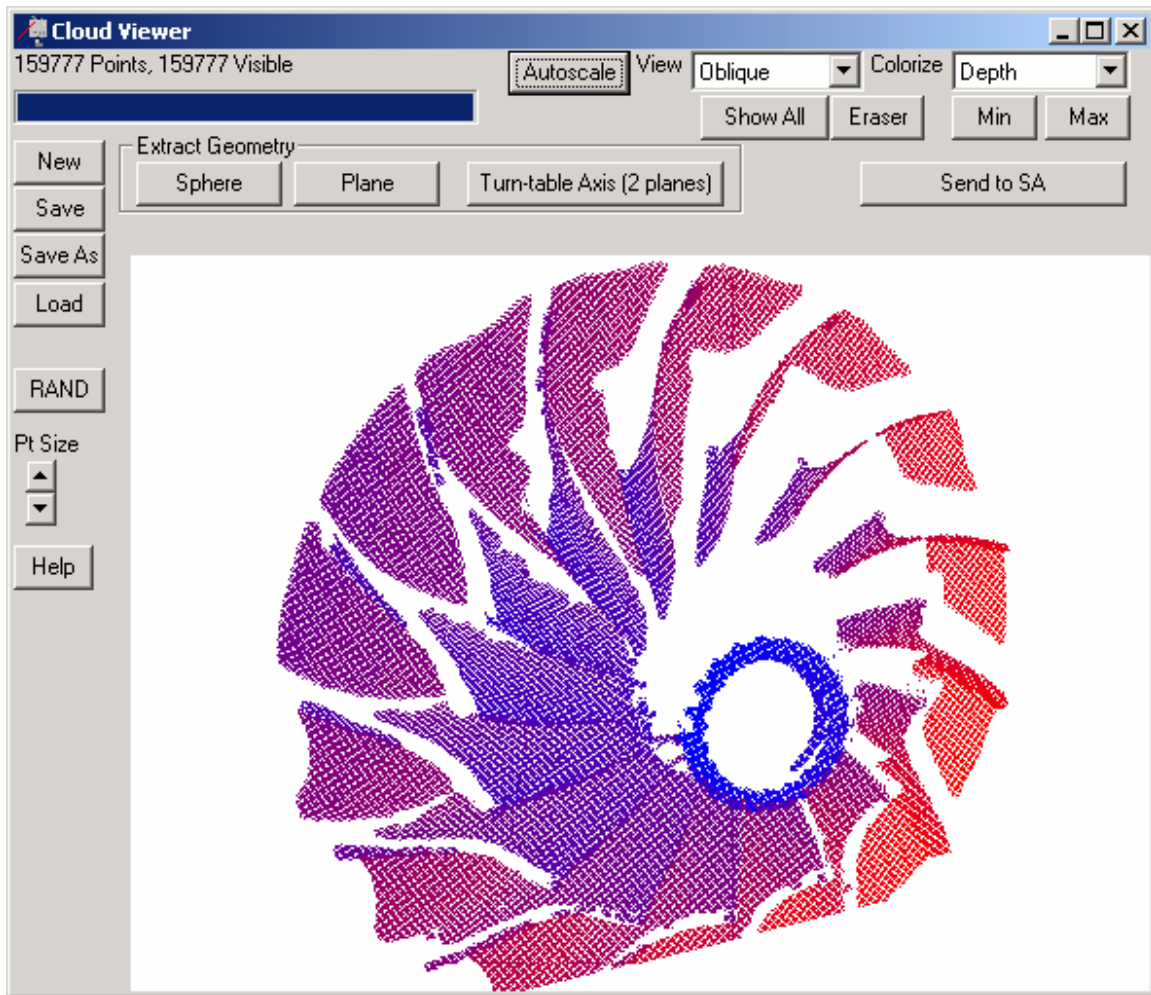
Notice that in this case, we are in simulation mode. This is because no instrument was found for a connection. If we were actually connected, the large window on the left side of the interface would show a view from the scanner camera.

Taking Your First Scan

The first step is to position the scanner properly relative to the part. Then, press the **Focus** button. This will attempt to focus on the object and set the laser intensity settings. Sometimes, you will need to enter a focus value manually, then autofocus. This is because the autofocus is performed based on starting at the current focus value. If this is completely invalid, the autofocus will fail.

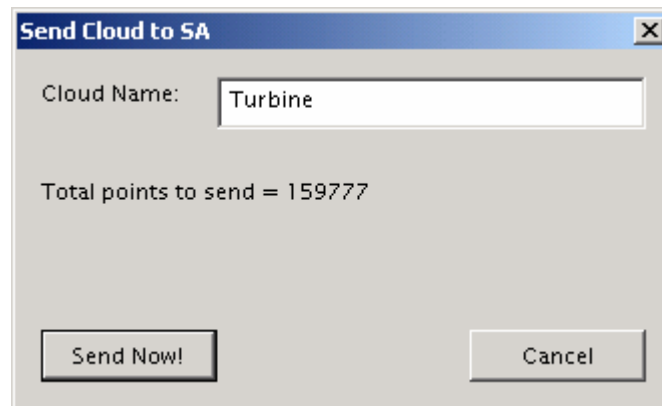
Once you are focused, press **Scan**. The scanner will beep, and the laser will sweep across the part. After the scan, you will see a notice in the status window telling you how many points were acquired in the scan.

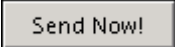
Next, take a look at the scan data you took. Press **Cloud View** to open the data viewer:



You can use the right mouse button to rotate the view and the left to zoom. For more detail on view control, press **Help**.

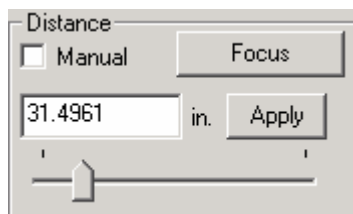
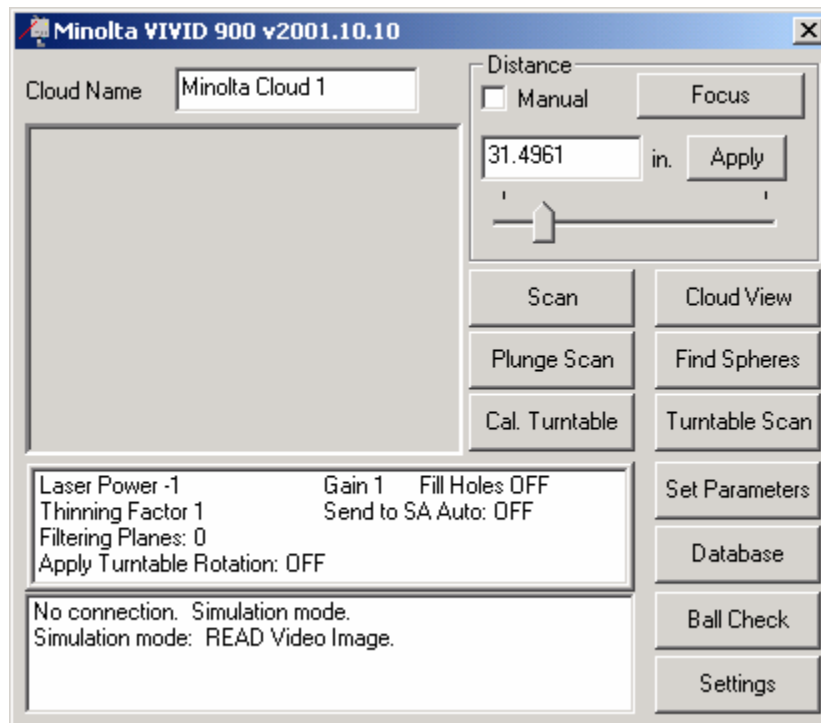
The next step is to send the data to SA as a cloud. To do this, press **Send to SA**. You will be asked for the point cloud name:



When you press , all the data will be sent to SA immediately.

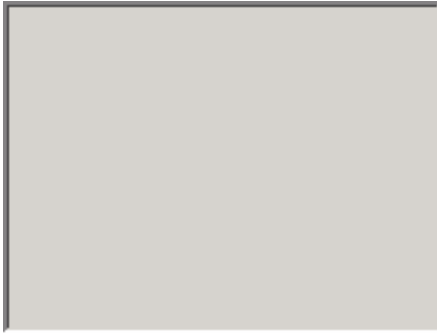
Interface Features

In this section, we will discuss each of the functions in the interface.



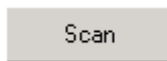
Focus Control

The focus setting is important to the successful operation of the VIVID scanner. There is a limited depth of field, so this value must be set properly so you can get the data you are interested in. The Manual checkbox lets you set a focus (either by entering a value, autofocusing, or using the slider), and leave it locked when you scan. If this is NOT checked, the scanner will automatically focus before taking a scan. When you change the focus, you will see the video image update as well.

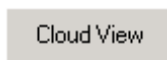


Camera Image View

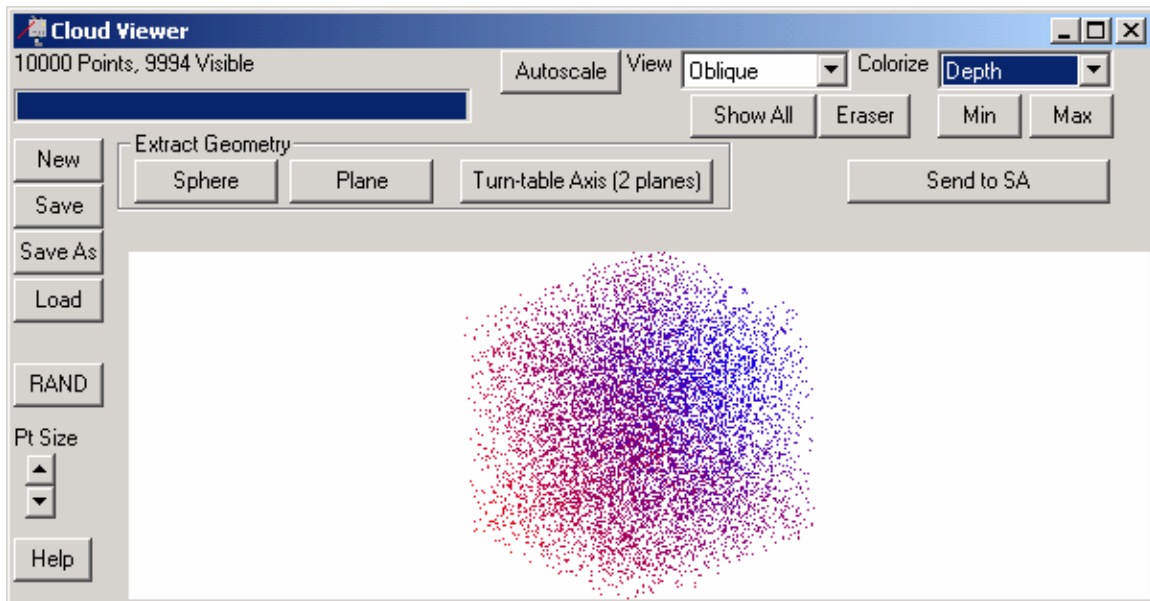
When connected to a scanner, this portion of the window contains the view through the scanner camera. It will be refreshed when you change the focus, but not when you scan (for speed reasons). If you want to refresh it at any time, simply click on it.



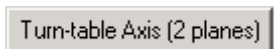
When you press this button, the scanner will scan using the current parameters.



Opens the viewer portion of the interface. This allows you to easily view the results of your scan, erase data, and send the remainder to SA:



Many of the basic functions of the cloud viewer are self-explanatory, but there are several that are worth mentioning:

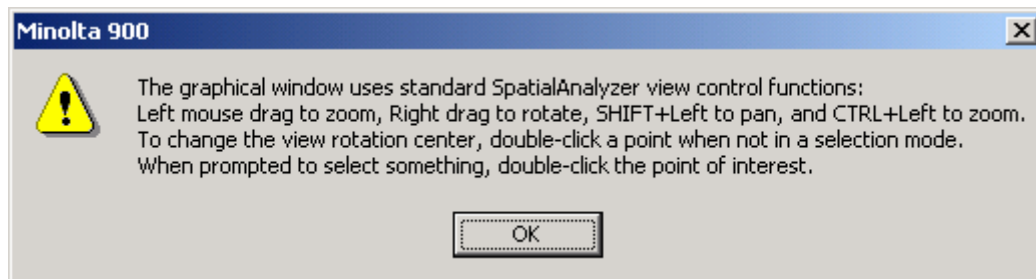


May be used to define the turntable axis. This is not as accurate as the sphere-based calibration available on the main interface screen, but it does

provide a quick method for characterizing the table axis. Pick 4 points on each of the planes to indicate the bounds for each plane fit. Make sure to go around the perimeter of the plane you wish to fit. Once the axis is computed, it will replace the current axis setting.

Eraser Allows you to trim unwanted data. When you press the button, the eraser block will appear. Use the arrow keys to change the size of the eraser and click-drag it around to wipe out points. Erasing simply sets a visibility flag so the points are not completely deleted until you send them to SA. When that happens, the non-visible ones are ignored. If you wish to undo an erasing option, press **Show All**.

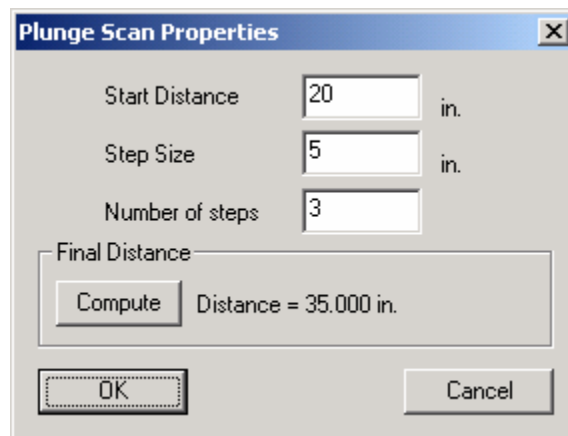
Help Provides information on view control operations:



Send to SA Transfers all the non-erased points to SA as a single cloud. If the cloud already exists in SA, it is appended. Otherwise, a new cloud is created.

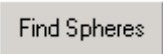
Plunge Scan

Since the depth of field of the scanner is limited, sometimes it will be necessary to scan an object from a certain vantage point several times in order to "plunge" through the focus range. This option let you do this. When you select it, you will be given options for the scan:

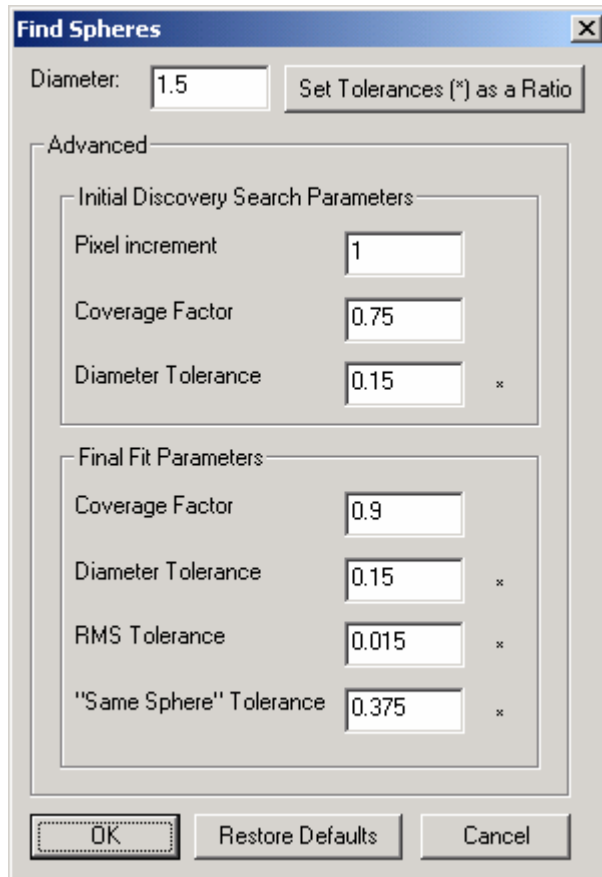


Indicate the starting focus value, the amount to step the focus by, and the number of steps. The Compute button allows you to calculate the final focus that will be used.

When you have configured the scan, press  and the scan will begin.

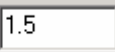


This function automatically searches the scan for spheres of a specified diameter. When they are found, a full fit is performed with all available points on the sphere. There are several options that control the performance of the algorithm. These are best explained by walking through the options page:



The 'Find Spheres' dialog box is shown. It has a title bar with a close button. Inside, there is a 'Diameter' input field with the value '1.5' and a 'Set Tolerances (*) as a Ratio' button. Below this is an 'Advanced' section with two sub-sections: 'Initial Discovery Search Parameters' and 'Final Fit Parameters'. The 'Initial Discovery Search Parameters' section contains 'Pixel increment' (1), 'Coverage Factor' (0.75), and 'Diameter Tolerance' (0.15). The 'Final Fit Parameters' section contains 'Coverage Factor' (0.9), 'Diameter Tolerance' (0.15), 'RMS Tolerance' (0.015), and '"Same Sphere" Tolerance' (0.375). At the bottom are 'OK', 'Restore Defaults', and 'Cancel' buttons.

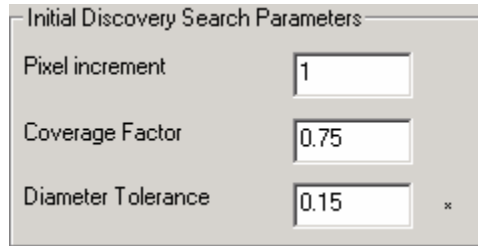
Section	Parameter	Value	Unit
Initial Discovery Search Parameters	Pixel increment	1	
	Coverage Factor	0.75	
	Diameter Tolerance	0.15	*
Final Fit Parameters	Coverage Factor	0.9	
	Diameter Tolerance	0.15	*
	RMS Tolerance	0.015	*
	"Same Sphere" Tolerance	0.375	*

Diameter: 

The nominal diameter of the spheres you are looking for.



Sets all other parameters based on default ratios of the diameter. These include all non-unitized values.



Initial Discovery Search Parameters

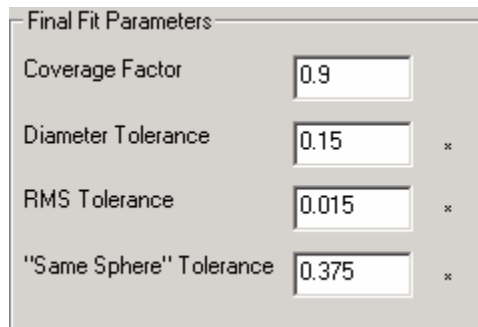
Pixel increment	<input type="text" value="1"/>
Coverage Factor	<input type="text" value="0.75"/>
Diameter Tolerance	<input type="text" value="0.15"/> *

The sphere extraction operation occurs in 2 phases. The first, and initial discovery, quickly passes through the data and identifies candidate spheres which are worth further investigation. These parameters control this scan.

Pixel Increment – the amount the test pattern is incremented as it moves across the point grid. 1 is the lowest value and provides the most thorough search.

Coverage Factor – the size of the test pattern as a factor of the nominal diameter. The test pattern consists of a center point and 4 radial points as 0, 90, 180, and 270 degrees. This factor controls the radial spread of these points. In some cases, you can reduce this value to 0.5 in order to avoid catching points near the edge of the sphere.

Diameter Tolerance – Once the test pattern has successfully passed a series of filters, 4 of the points are used to compute a sphere. If the diameter matches the nominal diameter by this tolerance, the sphere is considered further. You can increase this value to provide more liberal acceptance of test spheres.



Final Fit Parameters

Coverage Factor	<input type="text" value="0.9"/>
Diameter Tolerance	<input type="text" value="0.15"/> *
RMS Tolerance	<input type="text" value="0.015"/> *
"Same Sphere" Tolerance	<input type="text" value="0.375"/> *

Once a sphere has passed all the tests in the discovery portion, it moves on to the final fit. In this operation, a complete circle of points are used for a least-squares sphere fit. This is also done in 2 phases. First, a less rigorous fit is used to see if this sphere is still worthy of a complete fit. If that is true, a full, robust optimization is performed.

Coverage Factor – the diameter of the circle (as a fraction of the nominal diameter) used to filter the points for the fit. 0.9 covers 90% of the nominal diameter. This is a good value if you are getting reliable data near the edges of the sphere. If not, consider reducing this value to avoid the noise.

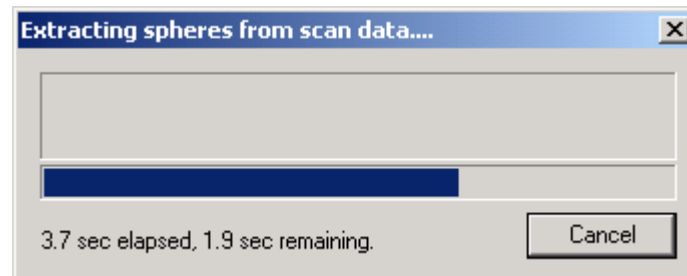
Diameter Tolerance – this nominal and fit diameters must match within this tolerance.

RMS Tolerance – The RMS from the sphere fit must meet this criteria or the sphere will be discarded.

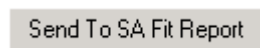
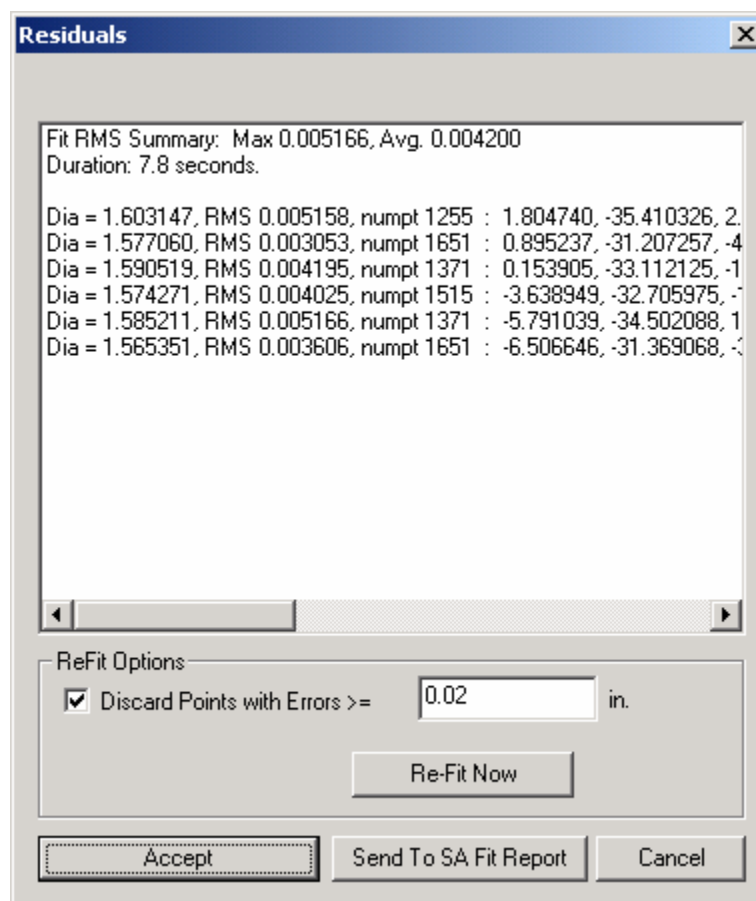
Same Sphere Tolerance – Since we are moving across the grid and locating candidate spheres, it is possible (and probable) that we will “discover” the same sphere more than once. If the centers point of a sphere matches a previously accepted sphere within this tolerance, it is ignored.



Press this to perform the sphere extraction. A progress bar will appear while the search takes place:



At the end, you will see the results:



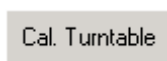
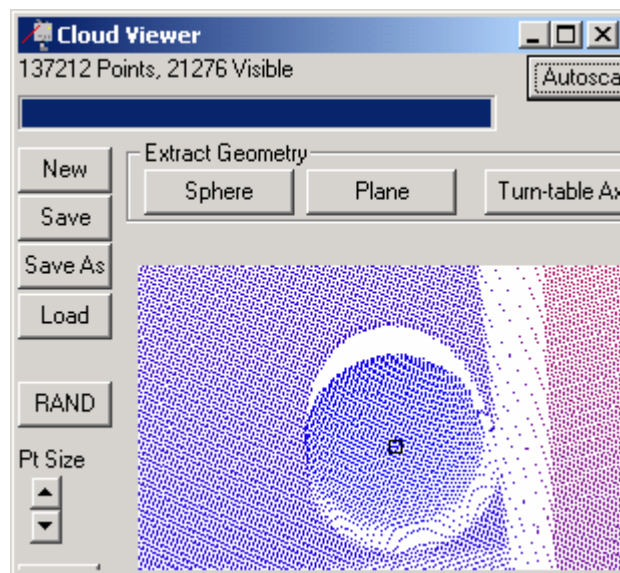
will allow you to send the text in this window to the SA Fit/Query report for later reference.



will prompt you for the group name in SA that will contain the points.

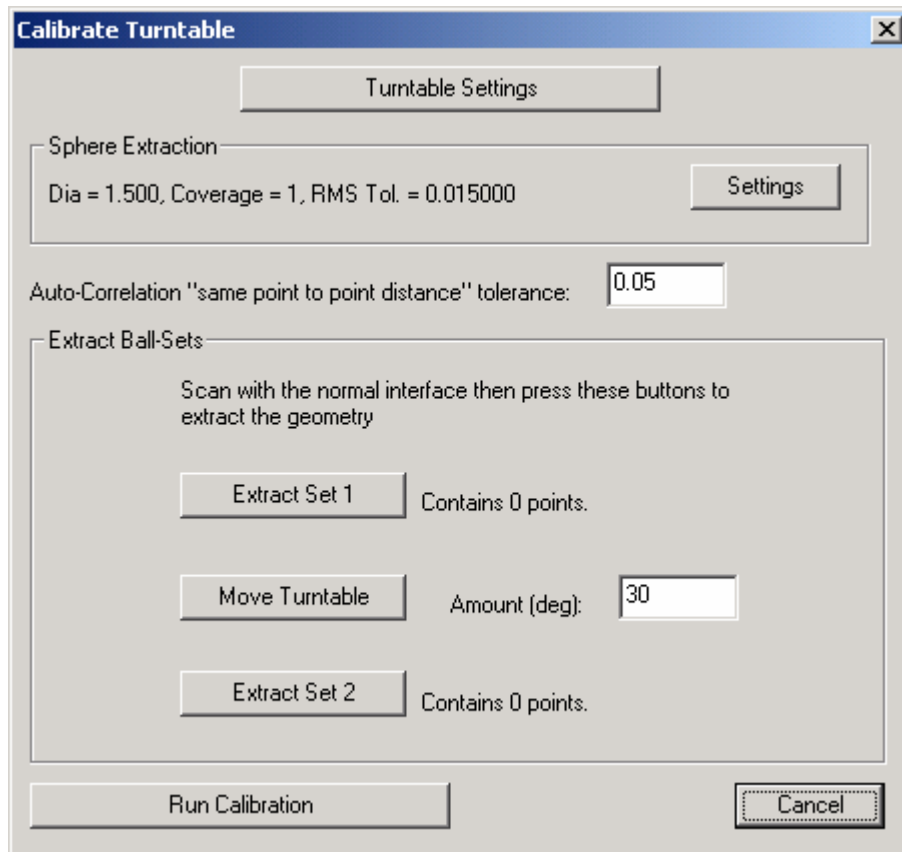


When you press OK, the points will be sent to SA immediately as measurements, not as clouds. Also, the centers of the spheres will be added to the cloud viewer and highlighted with boxes:



Calibrate the Turntable

In order to scan a part completely, it is often necessary to rotate it and take several scans. The turntable makes this easy and provides a way to automatically register the data sets. In order to make it useful, the turntable axis of rotation must be known. You can do this using the “2 plane” fixture and the cloud view plane fit option, or by measuring a ball-plate from 2 positions on the turntable. The latter method is the most precise, and will be discussed in this section:



Turntable Settings

Allows you to set the parameters of the turntable (COM port, rotation speed, position database). This will be presented in more detail in the Settings section.

Sphere Extraction

Dia = 1.500, Coverage = 1, RMS Tol. = 0.015000

Settings

The parameters controlling the sphere extraction component of the calibration. These settings are identical to those used in the “Find Sphere” function from the main interface screen.

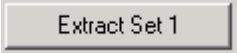
Auto-Correlation “same point to point distance” tolerance:

0.05

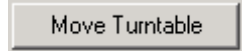
In order to calibrate the turntable axis, it is necessary to match up the spheres from the 2 calibration scans. To do this, SA’s VIVID interface uses a point set “Auto-Correlation” feature. This is similar to the options for this available in the standard SA interface. Basically, two point sets are matched together using the combinatorial inter-point distances in the 2 sets. This means you should not setup a calibration plate with the balls on a grid pattern. A random arrangement (with varying inter-point distances) is much preferred. This tolerance controls the point at which the algorithm says it is safe to assume that 2 points are the same based on inter-point distance.

Extraction Process:

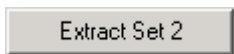
First, configure the scan settings in the main interface properly. Make sure the “Apply Turntable Rotation” option is OFF, and the focus is set properly. Then, take a scan. Use cloud viewer to verify that you are getting a good shot of all the balls on your calibration plate.


Then, Press . This will process the first scan. Make sure you get the correct number of points.

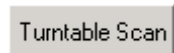
Next, enter the amount of turntable rotation. 90 is often a good value since it rotates a square calibration plate to a location where all of the balls should be in focus. Press

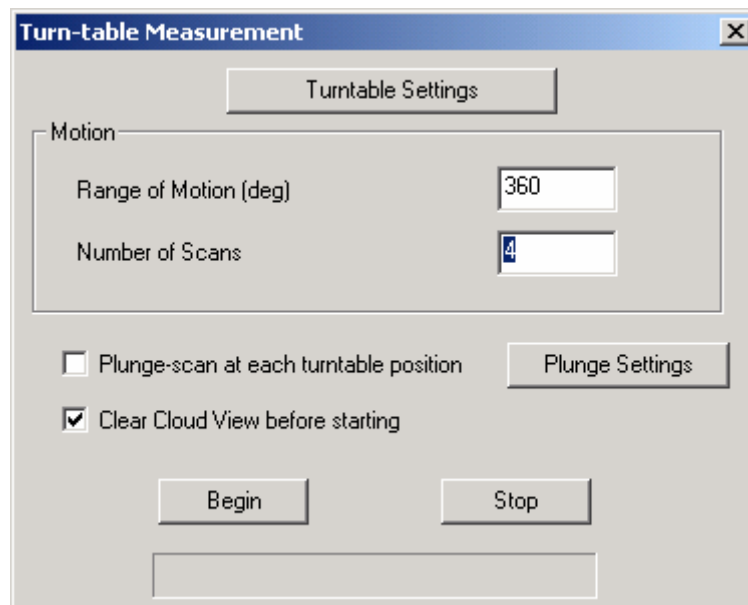
.

From the main interface screen, take another scan and verify the data with cloud viewer.

Press  and make sure you get the right quantity of points.

 Will match up the 2 points sets, perform an optimization to determine the axis of the table, and present results.

 Initiates an automated turntable scan. First, enter your options in the following dialog, then press begin to scan:



The image shows a software dialog box titled "Turn-table Measurement". At the top, there is a button labeled "Turntable Settings". Below this, a section titled "Motion" contains two input fields: "Range of Motion (deg)" with the value "360" and "Number of Scans" with the value "4". Below the "Motion" section, there are two checkboxes: "Plunge-scan at each turntable position" (unchecked) and "Clear Cloud View before starting" (checked). To the right of these checkboxes is a button labeled "Plunge Settings". At the bottom of the dialog, there are two buttons: "Begin" and "Stop". Below these buttons is a large, empty rectangular text area.

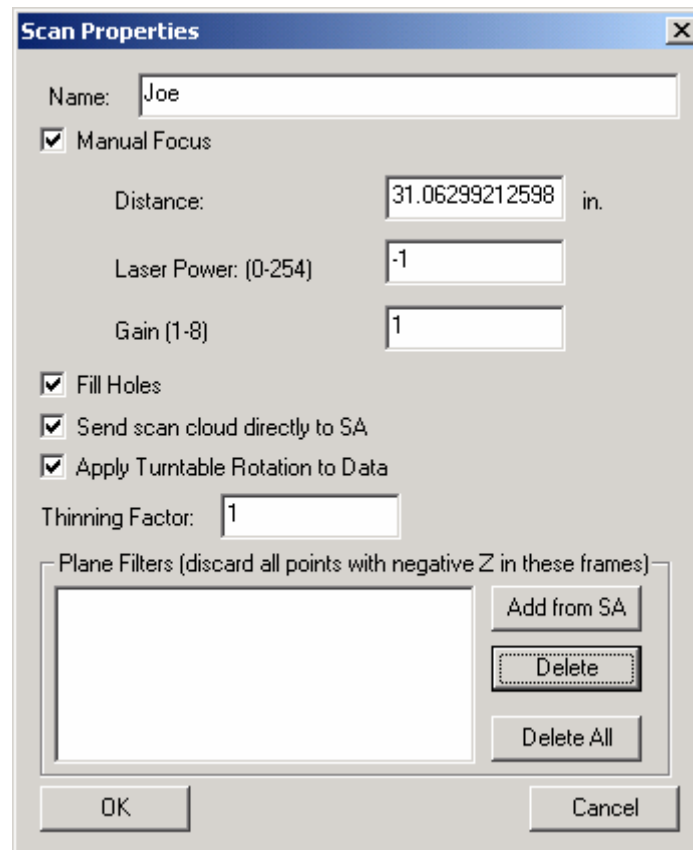
The turntable settings option lets you control the details of the turntable settings. Then, you pick the range of motion and the number of scans. The next option, Plunge-Scan, allows you to plunge at each location to maximize the coverage area for objects that do not fit within the standard focus depth.

Laser Power -1	Gain 1	Fill Holes ON
Thinning Factor 1	Send to SA Auto: ON	
Filtering Planes: 2		
Apply Turntable Rotation: ON		

Current scan parameters. Double-clicking on this control will pull up the current parameters just as if you just the Set Parameters Button.

Set Parameters

Set the current scan parameters. This shows the following dialog:



The image shows a 'Scan Properties' dialog box with the following fields and controls:

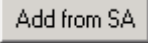
- Name:** Joe
- ☒ **Manual Focus**
 - Distance:** 31.06299212598 in.
 - Laser Power: (0-254)** -1
 - Gain (1-8)** 1
- ☒ **Fill Holes**
- ☒ **Send scan cloud directly to SA**
- ☒ **Apply Turntable Rotation to Data**
- Thinning Factor:** 1
- Plane Filters (discard all points with negative Z in these frames)**
 - Empty list box
 - Add from SA** button
 - Delete** button
 - Delete All** button
- OK** and **Cancel** buttons at the bottom.


The focus and fill holes options are directly analogous to the VIVID 900 settings for those values. See the Minolta documentation for more detailed information.

Send scan cloud directly to SA – This option causes the interface to send the scan data to SA immediately after it receives it from the scanner. The group name in the main interface screen will be used for the cloud, and the name will be incremented each time a cloud is sent.

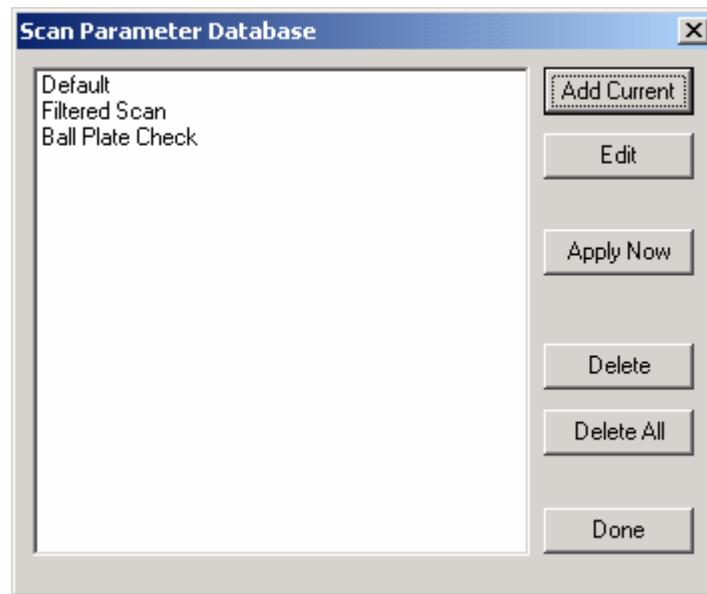
Apply Turntable Rotation to Data – If checked, the current turntable rotation value and the axis definition are used to transform the scan data to the proper position.

Thinning Factor – Used to reduce the data set density. 1 uses all the points – no reduction. 2 cuts the data in half.

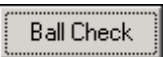
Plane Filters – allows you to apply a series of plane filters to the data. All data on the negative side of each plane is discarded. To add a plane filter, press . This will cause a prompt to appear in the SA interface. Select a frame to indicate the filter plane. The filter plane will be set to the X-Y plane of the selected frame.

 Accesses the scan parameter database

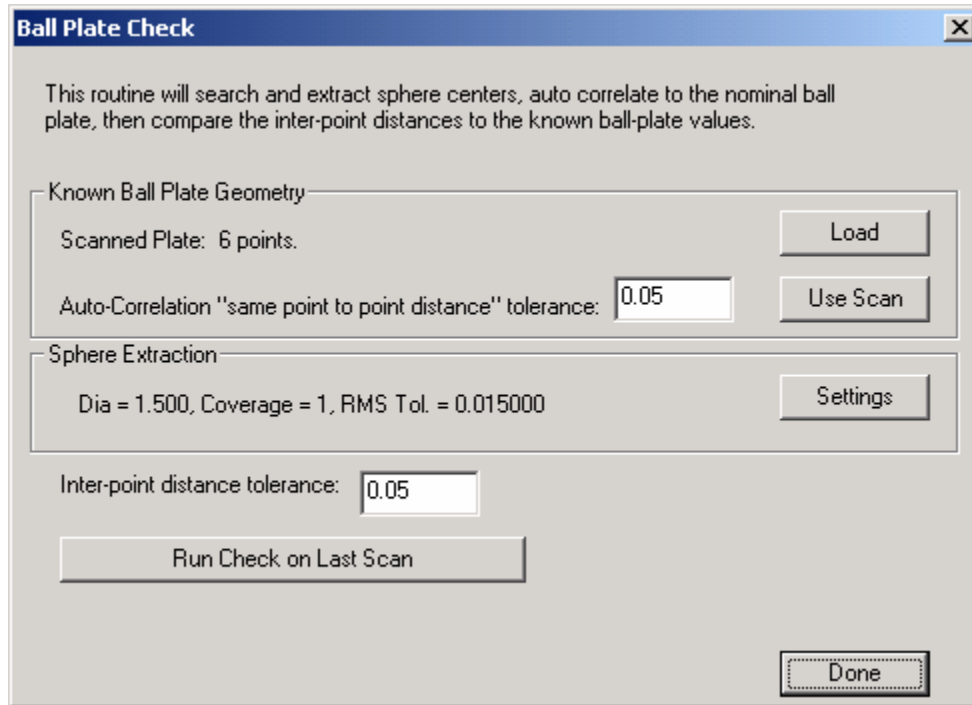
The database allows you to store a series of scan parameter sets and assign names to each set. Then, you can use the SA measurement plan to call up a specific set for a scan. The individual scan parameter sets have properties identical to those in the “Set Parameters” option on the main interface screen. The database window looks like this:



A new set is added using the Add current option. You can also select an item and press edit. Apply now sets the current interface scan settings to the selected set. These sets are stored with the scanner interface settings file in the SA persistence directory.

 Perform a ball-plate operational check of the scanner data.

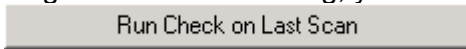
This function compares a measured ball-plate to a known artifact. The comparison is performed in 2 ways. First, a comparison of the inter-point distances is performed. This is analogous to a series of scale-bar checks. Second, a best-fit transformation is performed on the two data sets. This is done both scale-fixed and scale-free. In both cases, the results of the fit are presented in order to determine the health of the scanner data. When you run the ball-plate check, you will see this dialog:



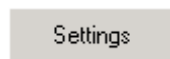
Known Ball Plate Geometry – This is the nominal geometry that will be used as the touchstone. You can either read this from a file, or use the current scan data, extract the sphere centers and treat it as nominal data.

Sphere Extraction – Allows you to control the sphere extraction parameters.

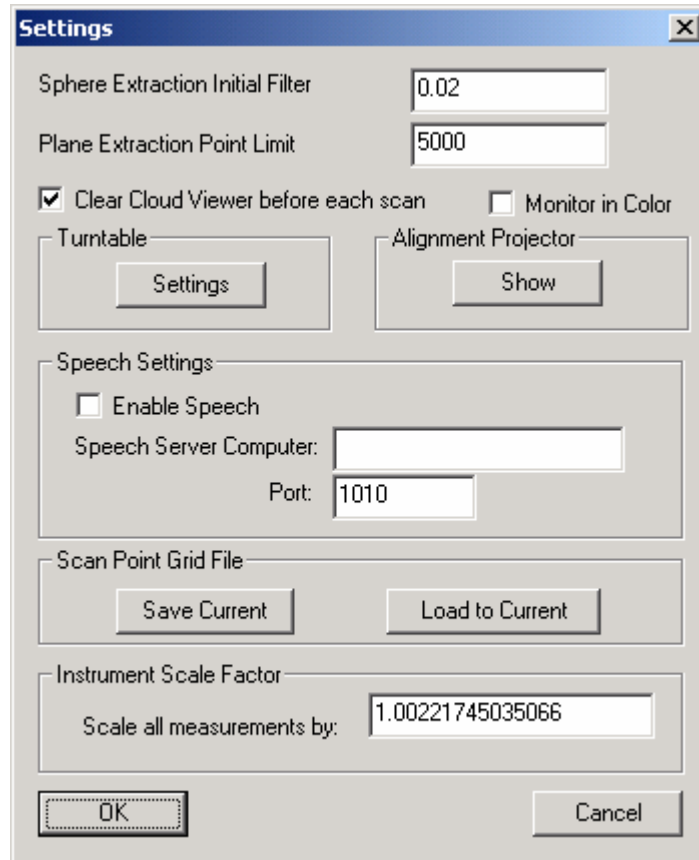
Inter-point Distance Tolerance – This is the tolerance for the OP check. If the distance of the inter-point comparisons exceeds this value, the op check is said to have failed.

Before entering the Ball check dialog, you must scan the plate. Then, check the settings and press . This will extract the spheres from the scan, then auto-correspond the scanned set to the nominal points. After they are correlated, a comparison will be performed.

Also notice that after a check is performed, information about the scale factor of the instrument is presented. This is the result of the “Scale-Free” best-fit of the measured and known ball-plate geometry. You will also see a button appear that will allow you (with several “are you sure?” prompts) to change the scale factor for all subsequent measurements.



Allows you to control general settings for the instrument interface.

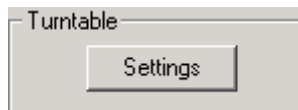


Sphere Extraction Initial Filter – Used by the sphere fit option in cloud viewer. Only points within this distance of the nominal 4 point sphere will be used in the full fit.

Plane Extraction Point Limit – The maximum number of points to be used in a plane fit from cloudviewer. If more points fall within the selected region, the set will be thinned to get this number of points.

☒ **Clear Cloud Viewer before each scan** - If selected, the cloudviewer will be cleared before a new scan is performed.

☐ **Monitor in Color** - Provides a color image in the camera view of the interface. This mode is not always supported by certain combinations of the SDK and different scanners.



Allows you to set the turntable properties:

Turntable Properties

COM Port: 1

Turntable steps per degree: 3125

Velocity: 3

Acceleration: 1

Move Table

Rotation (deg): 30

Jog Wiggle Goto Absolute

Reset Absolute Position Memory

Current Position: 0 Set

Preset Position Database

Edit

OK Cancel

Steps per Degree – varies with different turntables. See the documentation for your particular turntable for this value.

Move Table

Rotation (deg): 30

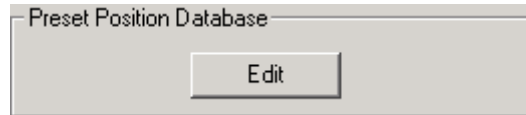
Jog Wiggle Goto Absolute

Move Table controls allow you to move the table to test communication. Be careful to let the motion finish before issuing another motion command.

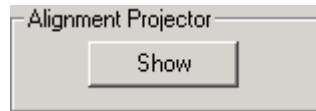
Reset Absolute Position Memory

Current Position: 0 Set

Most of the turntables used with the scanner have only relative position encoders. SA keeps track of the last position of the table in the interface settings file so that the table will behave as if it has absolute position reference. This area of the settings dialog allows you to reset the absolute position of the table.



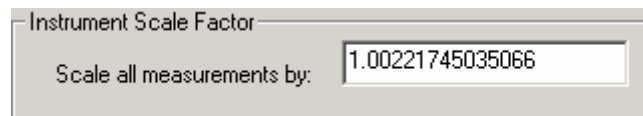
SA's measurement plan can control the position of the table by referencing any of these stored positions by name.



In cases where you will be repositioning known parts in a nominal location as part of an SA measurement plan, this function allows you to verify position and fine-tune if needed before scanning. When creating a measurement plan, position the part in the proper location, then use this interface to draw a perimeter around the part. Save the perimeter (or series of perimeters) in the database with a name that can be used to identify it later.

Speech Settings – these settings control where the interface should go to try to connect to the SA speech server. The speech component of this interface is most useful when using the PDA remote control.

Scan Point Grid File – These options allow you to store and load the raw scanner measurement data in grid form. This is useful for storing scan data to later perform sphere and other extractions. It is also good for scanner diagnostics. When you load a file, it is just like the data was just scanned. It will be in the current scan buffer, and is copied to cloud viewer. You could hit Find Spheres, for example, after loading the file to analyze the data.



This factor is applied to all measurement data from the scanner before it enters the instrument interface. The Ball-Plate check may be used to refine this value (it should be nominally 1.0), or you can enter another value.

Measurement Plan Functions

There are several measurement plan functions of specific interest to the VIVID interface. These will be presented in this section. There are a host of other functions for automatically registering to surface geometry, performing CAD to measured analysis, etc.

Step Properties

Step Type = Scan within perimeter

Step Title:

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument to scan	Enter Value	0
1	Perimeter Name	Scan perimeter name	Enter Value	
2	String	Parameter set name	Enter Value	buck1
3	Group Name	Group name	Enter Value	bucktest

OK Cancel

Scan within perimeter performs a scan (ignoring the perimeter field) using the parameter set name provided in the MP and placing the results in a point cloud with the name given in the group name field.

Step Properties

Step Type = Instrument Operational Check

Step Title:

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument to Check	Enter Value	0
1	String	Check Type	Enter Value	BALLPLATE

OK Cancel

After taking a scan (using the scan in perimeter command), you can issue an op check command. For the VIVID, the only supported op check is BALLPLATE. This will extract the spheres, perform the analysis, and append the results to the Instrument Report in SA.

Step Properties

Step Type = Issue Instrument Actuator Command

Step Title:

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument ID	Enter Value	0
1	String	Command	Enter Value	TablePosLeft

OK Cancel

This command moves the turntable to the preset position (defined in the turntable position database).

Step Properties

Step Type = Set Alignment Projector

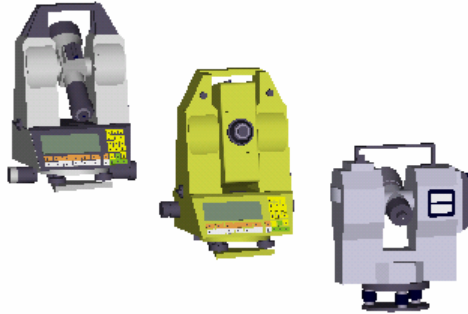
Step Title:

Arguments				
Arg#	Type	Description	Method	Value
0	Inst. ID	Instrument ID	Enter Value	0
1	String	Projector Profile	Enter Value	CT456
2	String	User Prompt	Enter Value	Please Align the Part

OK Cancel

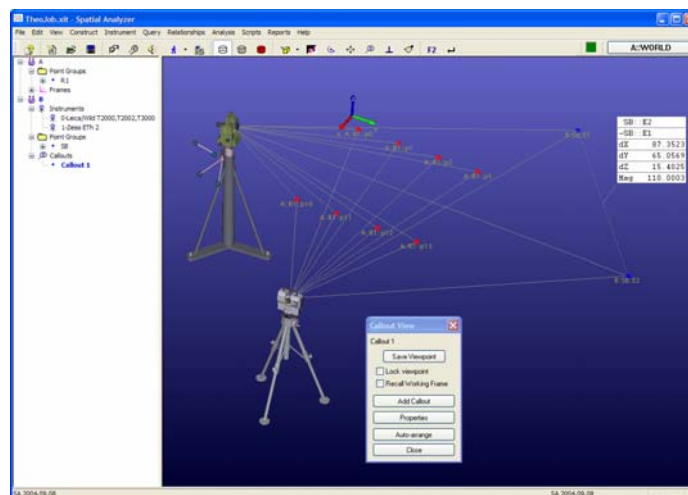
The figure above displays the Alignment Projector dialog properties which will use the user-specified projector profile. Also displays a prompt in the window specific by the user. Then the user presses OK in the dialog (or CANCEL), the measurement plan progresses.

THEODOLITE & TOTAL STATION INTERFACES



Theodolite manager provides a solution for interfacing all modern digital theodolites to SpatialAnalyzer. Theodolite Manager communicates through standard RS-232 serial ports to a wide variety of both conventional theodolites and total stations.

An image of an example SpatialAnalyzer job with Theodolites is shown below. The instruments are shown with measurements and a scale bar. The actual distance between the scale bar points is shown with a Point Comparison callout view.



Cabling and Power Options

Fundamentally, communication to all current digital theodolites is done through a serial link. However, over time, different instrument manufacturers have selected slightly different flavors of serial communication to talk to their instruments. Some units use TTL (Transistor-Transistor Logic) level signals, but otherwise essentially follow RS-232 protocols. Additionally, some instruments require 6 volts for power, others 12 volts and still others 4 volts. Because of this non-uniformity among manufactures some special cabling is usually required for a specific instrument.

Direct to the Instrument

This way of cabling an instrument is really most useful for applications where a single Total Station is to be used or where a limited network is configured. In this case, the instruments may be plugged directly into the computer or to a power augmenting device

supplied by the instrument manufacturer. Most manufacturers supply cables and power units for this purpose.

Using a Serial Distribution Interface (SDI)

In instances where a user wishes to use a network of theodolites cabling issues for power and signals can become a major annoyance. The SDI unit manufactured by Brunson Instrument Company addresses these issues, by providing a simple USB device that provides all instruments with power and digital signals while still maintaining common cabling for each instrument. This is accomplished by using a signal conditioning device embedded in the Theodolite Junction Box in the last few feet of the cabling. The Theodolite Junction box performs three primary tasks:

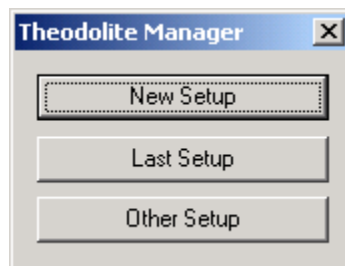
It alters the signal to levels suitable for each instrument.

It provides a convenient “trigger” button remote from the instrument itself.

It provides a junction point to which a short, theodolite specific cable can be attached. This Personality cable is necessary since there is no standard connector for all makes and models of theodolites.

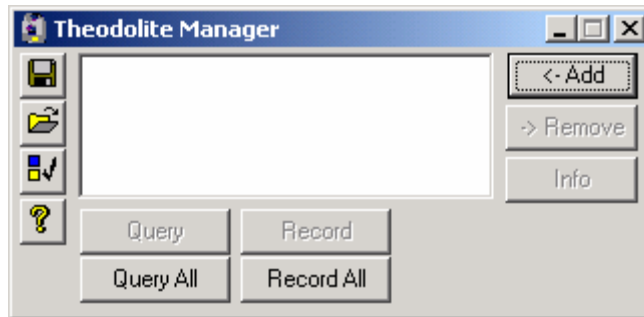
Adding a Theodolite to an SA Job

Just as with other instrumentation in SA, the first step is to add the desired instrument to the SA job. This is done by selecting Instrument->Add from the main menu. Now you need to start the corresponding instrument interface. This can be done from the Operating System Start-Bar, or by selecting Instrument->Run Instrument Module from the main menu. After selecting the appropriate instrument, you will see the following Initialization Dialog from the Theodolite Manager.



These three buttons allow you to either make a new configuration of instruments, recall the last configuration, or recall a stored configuration by name. The first time you run the software you must make a new configuration.

You will then see the main Theodolite Manager dialog.



The following sections will describe each component of the interface by section.

The area appearing as a large blank region is a listing of all active instruments being managed by this instance of the theodolite manager. It is possible to have multiple versions of theodolite manager tied into the same SA job at once.



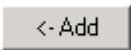
The control buttons allow the operator to perform global tasks related to Theodolite Manager. In order from top to bottom these buttons provide the capability to save a Theodolite Manager configuration, recall a configuration, set user options for a configuration, and view information about Theodolite Manager.

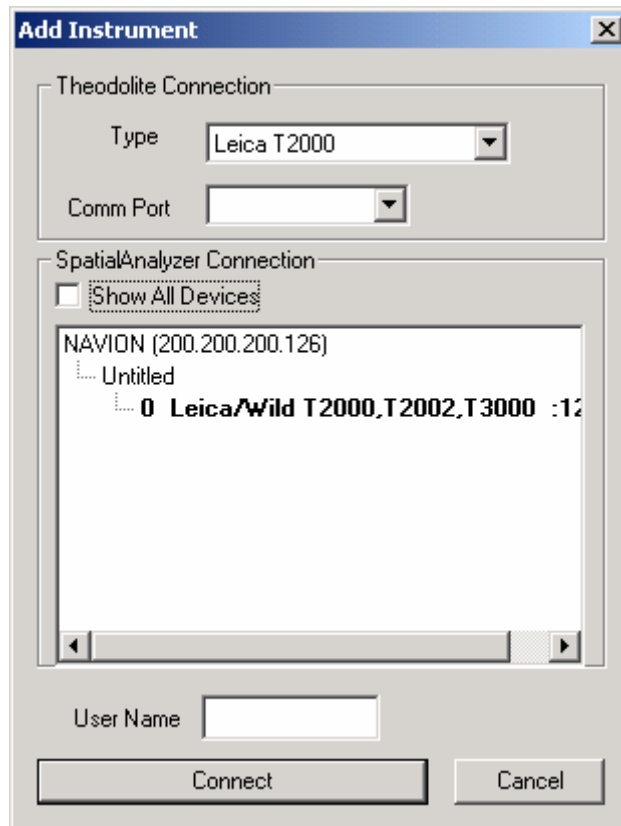


The buttons to the right of the instrument list are used to manage the instrument list. Here the user may add, delete, or display information about a specified theodolite connection.

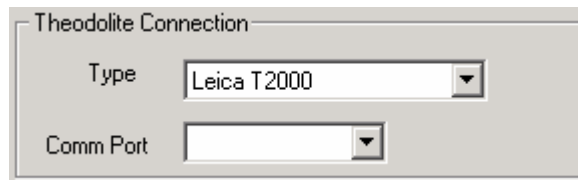


The Button to the bottom of the dialog are user to poll the instruments connected to Theodolite Manager. The top two only poll specifically highlighted instruments in the instrument list. If none are highlighted the buttons are deactivated.

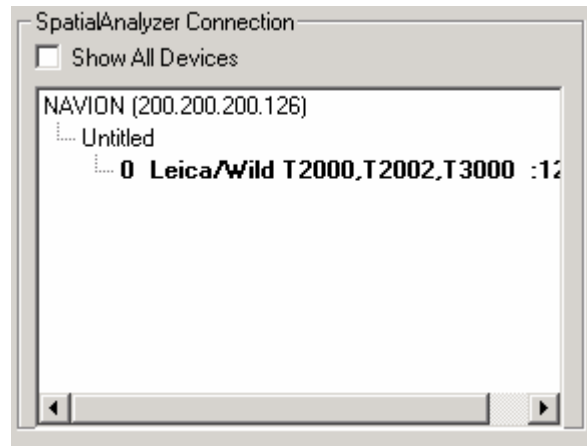
To add an instrument, press the  button. A progress bar will display briefly while Theodolite Manager scans for available communication ports. The Add Instrument dialog will then appear.



In the Theodolite connection area of the dialog users select the desired theodolite and its corresponding communications port.




The operator must then identify which SpatialAnalyzer job and which instrument within that job it corresponds too. By default the dialog limits its display to compatible and available instruments in all jobs on the network. If the show all devices checkbox is checked all instruments and SA jobs on the local network are displayed. The operator must simply highlight the desired instrument.

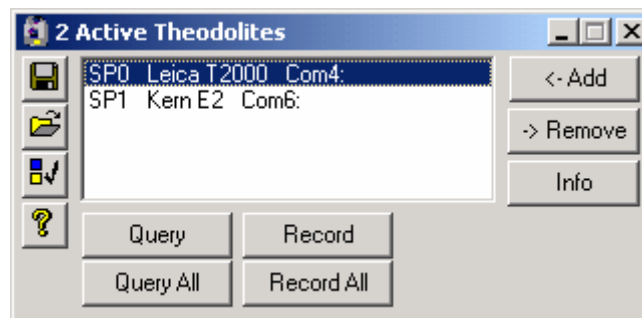


The operator also has the opportunity to optionally enter the user name for the operator of this instrument. If no name is given the instruments will be referred to as Station1, Station2, etc.

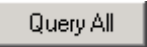


Once all selections are made the operator may hit the  button. At this time the instrument is initialized and an item is added to the instrument list.

In this case it identifies that SpatialAnalyzer Instrument 0 (SP0) is a Leica T2000 connected to serial port Com4:. After adding a second instrument the display might look like this.




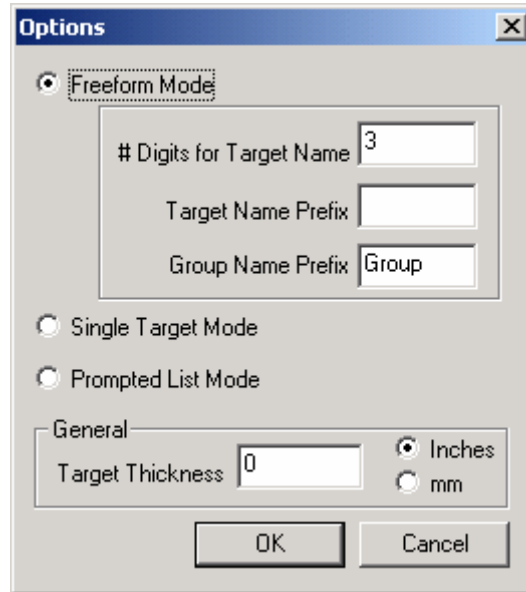
Notice that, in this case we have highlighted one of the instruments and the other buttons in the interface have therefore been activated. Multiple highlights are possible and are accomplished using the standard windows selection methods of holding down either Ctrl, to add select, or Shift, to region select multiple instruments.

To demonstrate that the instruments are now connected, move the instruments to a new position and hit the  button. You should see all the scopes move to their actual azimuth and elevation in SA.

Measurement Modes

Theodolite Manager allows operators to use theodolites in one of three primary measurement modes. These modes have been established based on a survey of common methods for applying theodolites. Some industries or applications may exclusively use one method while still others will vary their selection with each task at hand.

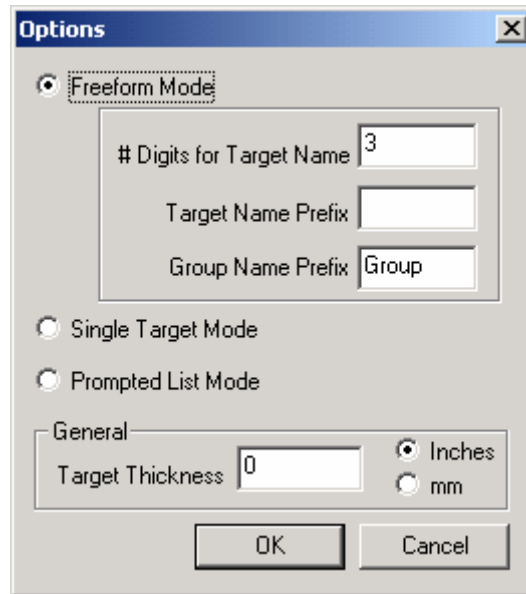
To change Measurement modes, hit the  (User Options) button. The following dialog will be displayed.



The image shows a Windows-style dialog box titled "Options". It contains three radio buttons for measurement modes: "Freeform Mode" (selected), "Single Target Mode", and "Prompted List Mode". Below these, there are three text input fields: "# Digits for Target Name" with the value "3", "Target Name Prefix" (empty), and "Group Name Prefix" with the value "Group". At the bottom, there is a "General" section with a "Target Thickness" input field set to "0" and two radio buttons for units: "Inches" (selected) and "mm". "OK" and "Cancel" buttons are at the bottom right.

The Target Thickness section will apply to all measurement modes. This is simply the target thickness (probe offset) used in SA for various analysis tasks such as point to surface queries.

Freeform Mode



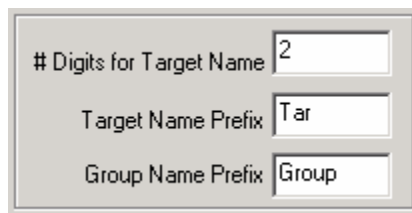
The screenshot shows the 'Options' dialog box with the 'Freeform Mode' radio button selected. The 'Freeform Mode' section contains three input fields: '# Digits for Target Name' with the value '3', 'Target Name Prefix' which is empty, and 'Group Name Prefix' with the value 'Group'. Below this section are two unselected radio buttons: 'Single Target Mode' and 'Prompted List Mode'. A 'General' section contains a 'Target Thickness' input field with the value '0' and two radio buttons: 'Inches' (selected) and 'mm'. At the bottom are 'OK' and 'Cancel' buttons.

The Freeform measurement mode is really designed for experienced crews who want and have the ability to enter target numbers at the theodolite. By far the biggest collection of these users are experienced operators of Leica T2000, T2002, and T3000 series theodolites. However, this methodology applies to other instruments as well. Newer Leica instruments and Zeiss Eth-2's for example, can use this method.

In freeform mode it is always necessary that the operator set the target number on the scope before recording. Recording may be accomplished at the instrument or the interface as desired.

Since SA uses both Target Names and Group Names the strategy employed is to arbitrarily divide the number into a group component and a target component. Several examples will be presented below to further illustrate the concept.

With the following settings:



This screenshot shows a portion of the 'Options' dialog box with the following settings: '# Digits for Target Name' is '2', 'Target Name Prefix' is 'Tar', and 'Group Name Prefix' is 'Group'.

The following target numbers would convert to.....

Target Number	Group Name	Target Name
1000	Group10	Tar00
2505	Group25	Tar05
2833	Group28	Tar33

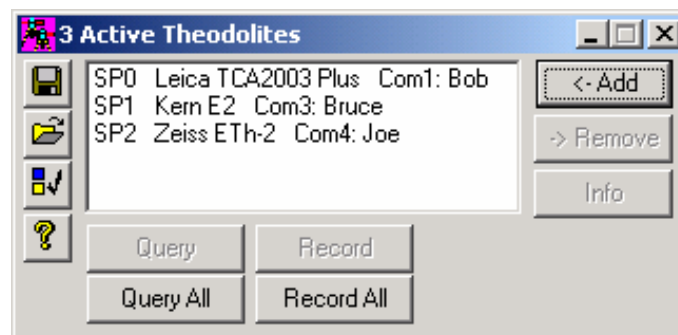
With the following settings:

# Digits for Target Name	3
Target Name Prefix	T
Group Name Prefix	Fender

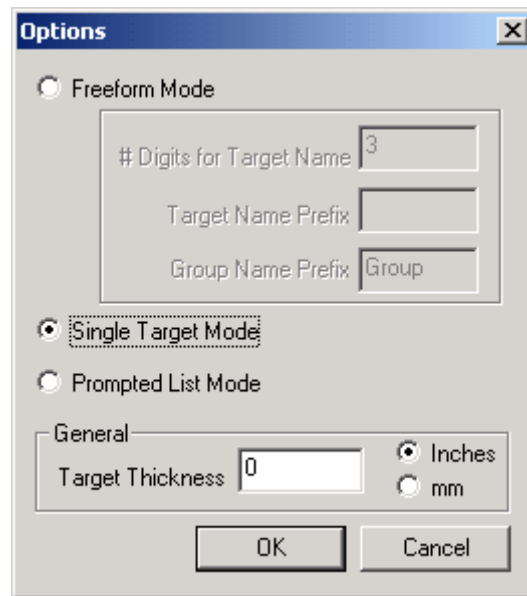
The following target numbers would convert to.....

Target Number	Group Name	Target Name
1000	Fender1	T000
2505	Fender 2	T505
2833	Fender 2	T833

Note that the primary user interface remains just the following simple dialog.

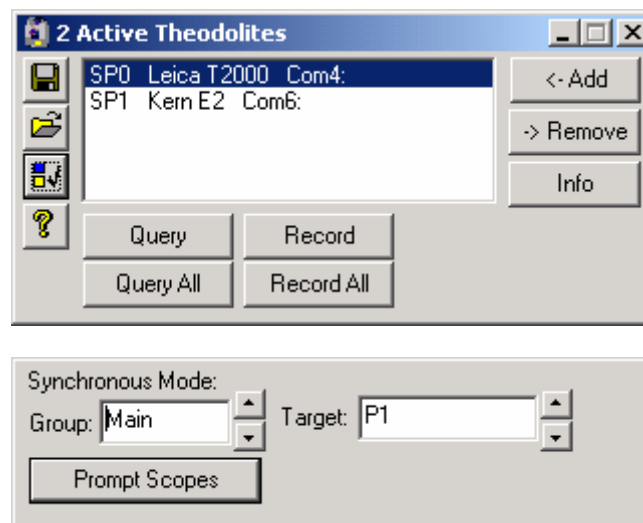


Single Target Mode



This mode is often used when building operations need to be performed or when a novice crew needs to be guided through the measurement process. The basic premise is that all theodolites will typically shoot the same target at the same time. This is particularly helpful for inexperienced users since sighting errors can be immediately identified and corrected.

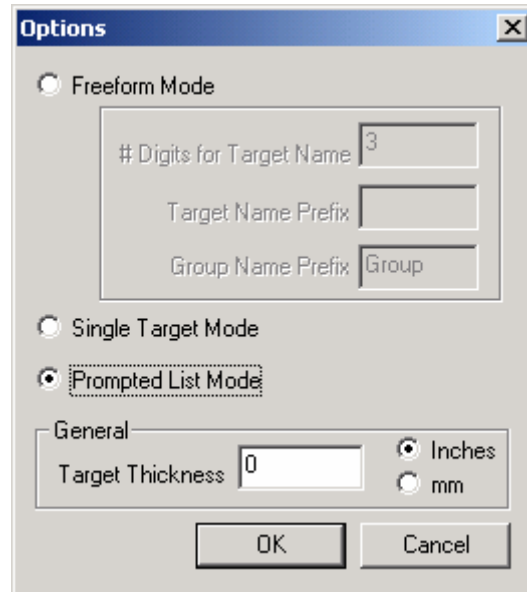
After enabling the Single Target Mode in the user options dialog, the interface will appear as follows.



Note the extra section added to the bottom of the standard dialog. This enables users to enter alphanumeric group and target names. When the Prompt Scopes button is hit, the theodolites will display the desired target to prompt the scope operators to point at it. Any incrementing or decrementing of the group or target name will automatically update

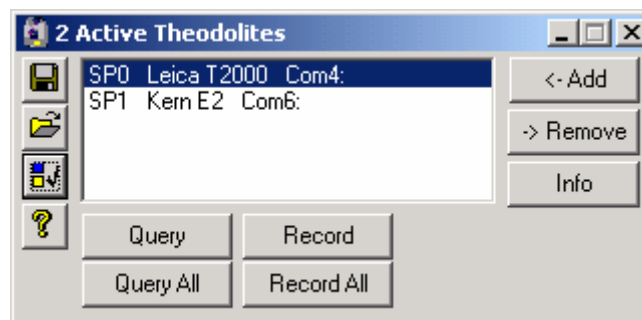
the instrument prompts. Recording may be accomplished through the standard user interface or by hitting the record button on the theodolite or its corresponding Theodolite Junction Box.

Prompted List Mode



This measurement mode is designed for tasks where the desired survey points are known right from the start. These names may either come from a CAD file, an Ascii file or may be entered at survey time by the operators.

When the Prompted list interface is initiated, the user interface appears as follows:



Read ASCII List Build List Get List From SA

☒ Synchronous ☐ Asynchronous

Station ID	Target	R	Q	S
Station0	No List	↓	R	Q S
Station1	No List	↓	R	Q S

Alternatively, if user names were entered as the instruments were added, this dialog might appear as follows:

3 Active Theodolites

SP0 Leica T2000 Com1: Bob
 SP1 Kern E2 Com2: Joe
 SP2 Zeiss ETH-2 Com3: Bruce

<- Add -> Remove Info


Query Record
 Query All Record All

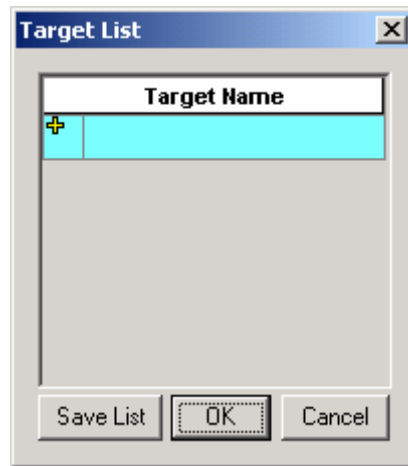
Read ASCII List Build List Get List From SA

☒ Synchronous ☐ Asynchronous

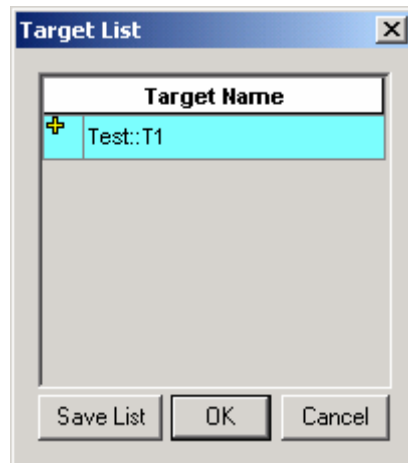
Station ID	Target	R	Q	S
Bob	No List	↓	R	Q S
Joe	No List	↓	R	Q S
Bruce	No List	↓	R	Q S

In this mode, each instrument has it's own list of targets to be surveyed. This list may be created on the spot using an ascii editor such as notepad, or by using the **Build List** option. Or it may be read from an existing file with the **Read ASCII List** option. Alternatively the user may hit the **Get List From SA** and they will then be prompted in SA to select points. All points are identified by both group and target names with the following format (group::target).

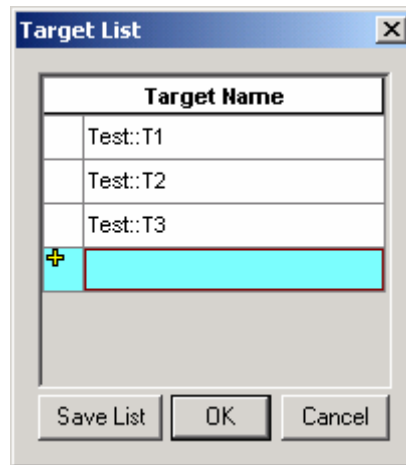
When the  button is hit, the operator sees the Construct Target List Dialog shown.



By typing a "group::target" name in the green field and hitting enter a new target is added to the list.

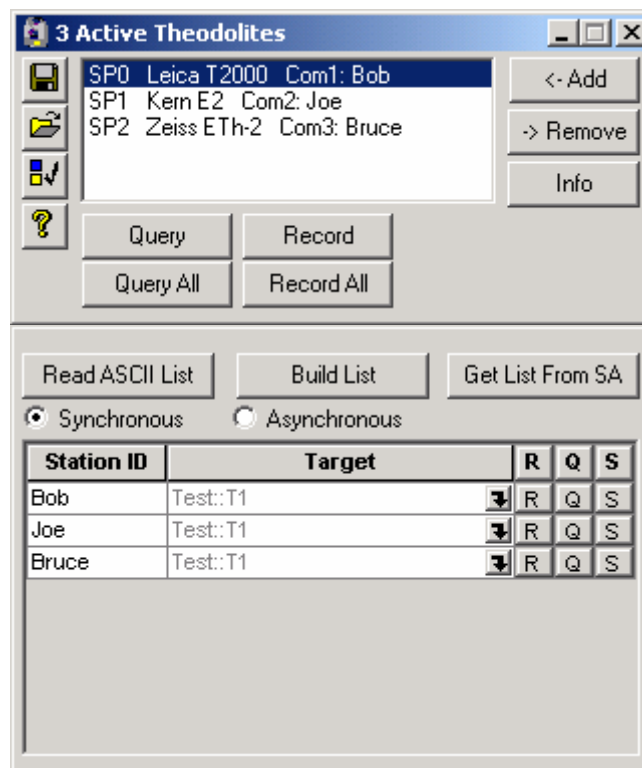


After several are entered our list might appear as follows:



Hitting OK will retain the list for this job. Hitting Save List will allow the operator to specify a filename so that this may be recalled later (always a good idea).

Returning to the main dialog, we see that the instrument list now has two lines. One line corresponding to each instrument station (or user).



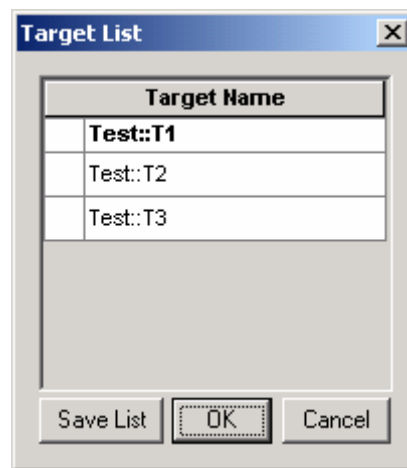
The buttons to the far right hand side in each row allow the operator to Record - R, Query - Q, or Skip -S a specific instrument.

The second column is the current target for a given user. When the dialog is first presented this column displays “No List” since a target list has yet to be loaded. After the target list is loaded, this column will display the current target for each instrument.

In Synchronous mode all instruments will be prompted to shoot a specific target. As each operator acquires data, they will be prompted to “Wait”. When the last theodolite acquires data, all scopes will be prompted with the next target name. In Asynchronous mode all operators may proceed through the list at their own rate.

A target hot-spot box to choose the current target for that scope (more about that later). To the right are buttons for Record, Query, and Skip.

The down arrow in the second column is called the Target Hot Spot Control. The target hot spot pops up the target list for each instrument. When you hit the hotspot you get the list:

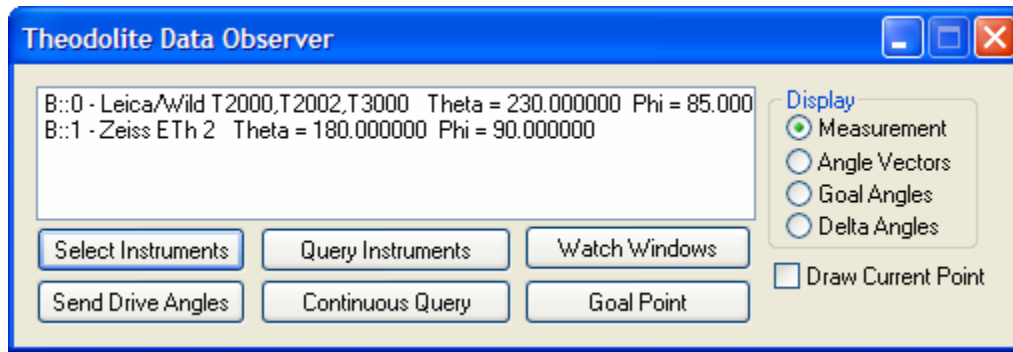


This is similar to the building list except the user cannot add new targets and the left column displays target status for that particular instrument.

The left column presents icons to indicate state of the target (Shot, Skipped, UnShot). The bold item is the current target for that user. You can click select what target you want to shoot next, and then either continue from there or (optionally) only prompt skipped or undone targets. When you shoot or skip the scope will proceed to the next target in the list. In Asynchronous mode all scopes use the same list but can progress through them at their own pace, skipping as required, and going back as required. Note that the operator will not normally view the list in this way, normally they would view all this information in the main dialog. The list is just a control to allow the user to move around the job freely.

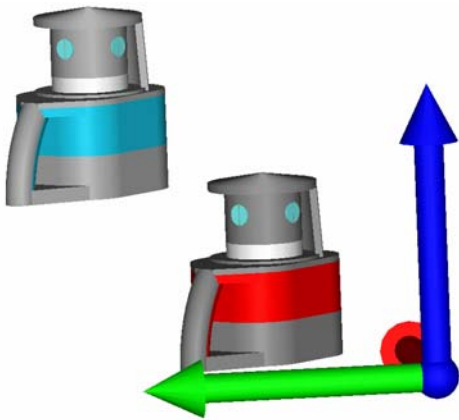
Theodolite Data Observer

Theodolite Data Observer is activated from SA’s Instrument menu. This dialog makes it easier to drive Theodolite systems from the graphic view in SA. The interface is shown in the figure below.



This interface allows the display of Measurements, Angle Vectors, Goal Angle, and Delta Angles. Each view of the data is applicable for different applications. Use the Select Instrument button to choose the instruments to control from this control. Select the desired Goal Point then Send Drive Angles and Watch Window options to setup a real-time graphical build mode for the currently selected instruments. Continuously Query will repeatedly record angle measurements from the instruments. This mode is important for tracking the theodolites movements through a measurement process.

ARCSECOND NETWORK AND TRANSMITTER SIMULATOR INTERFACES



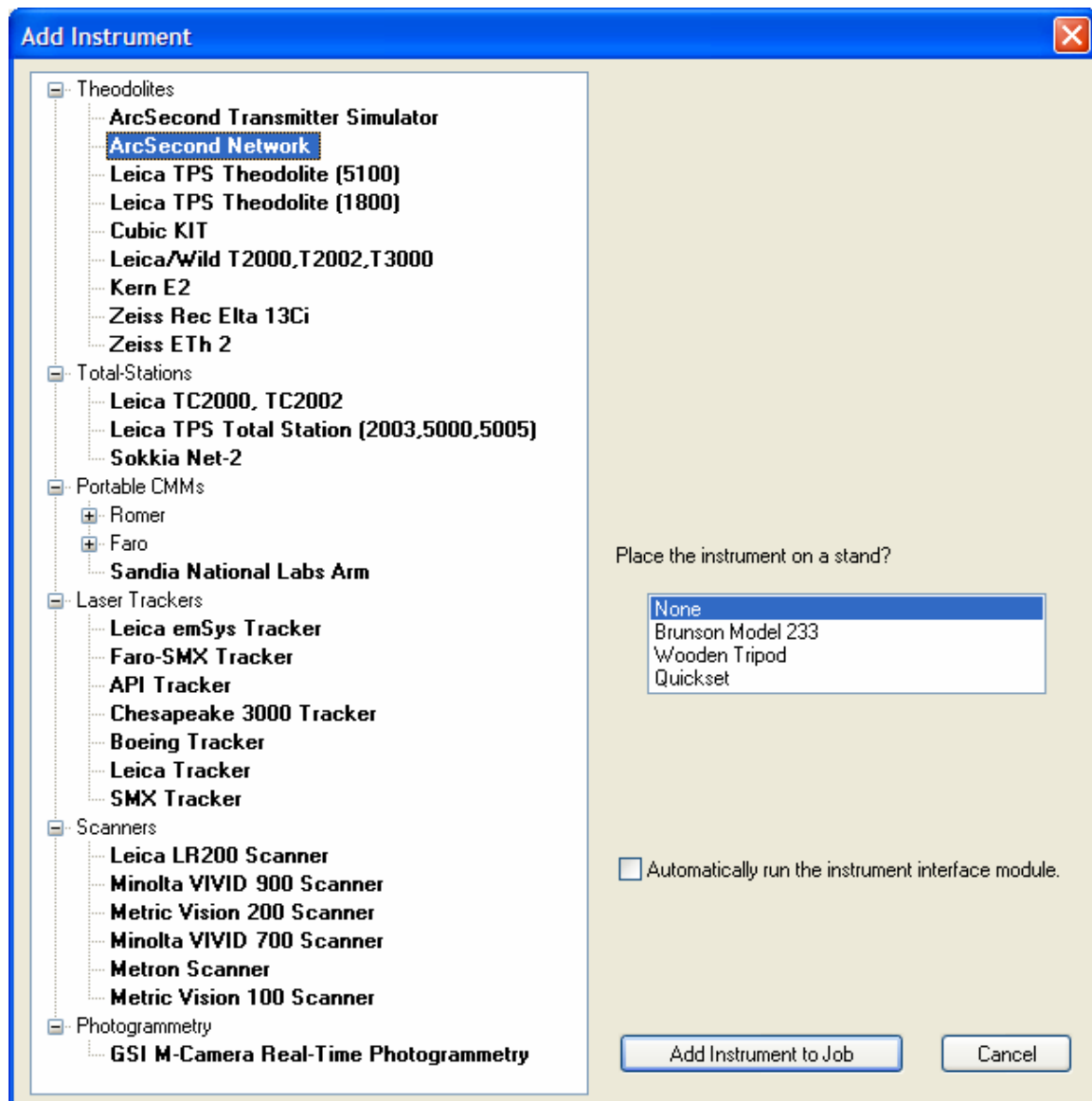
The purpose of this section is to briefly present the features of the SA-AS interface.

When adding an Arcsecond instrument to SA, you can choose an ArcSecond Transmitter Simulator for analysis and experimentation, or an ArcSecond Network for real-time measurement.

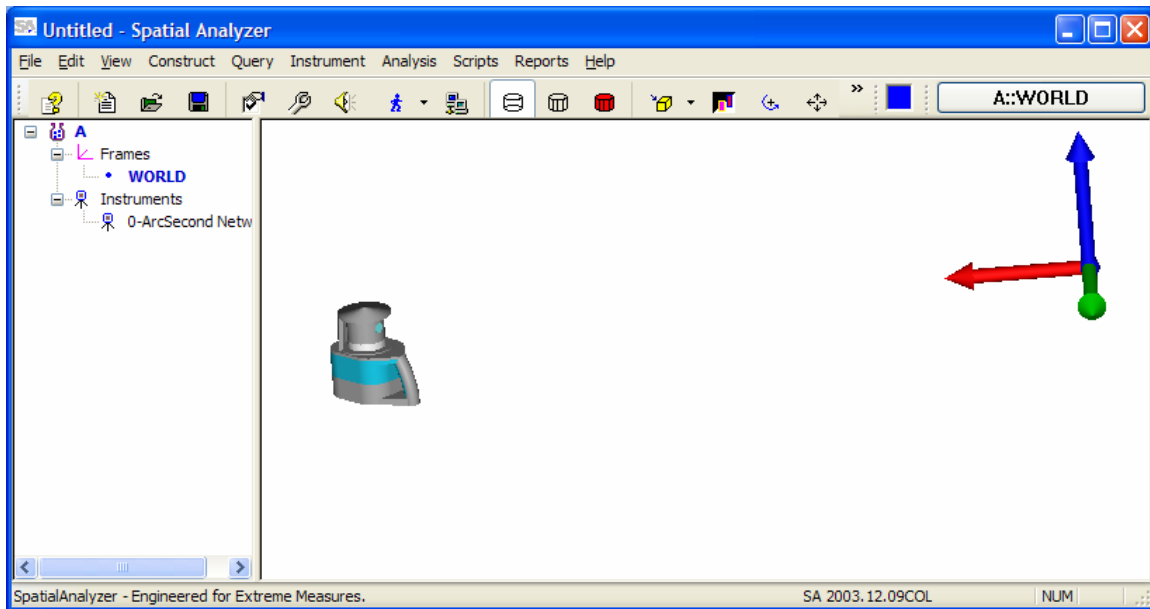
The ArcSecond Transmitter simulator enables you to model a setup and objectively determine expected performance for a particular application. The Network connection interface provides a real-time interface between SA and Arcseconds 3DIWorkbench.

Adding a Arcsecond Network

The first step when adding a Arcsecond Network interface to SA is to select Instrument >> Add Instrument from the menu. The figure below shows the Add Instrument dialog. Select the Arcsecond Network instrument from the list of available instruments.



The instrument is added with a graphical model of the transmitter:

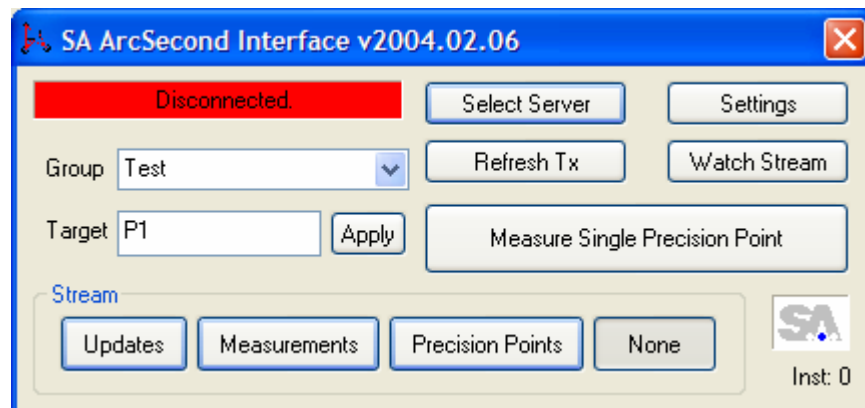


Note that if you add the transmitter simulator, the band around the base is RED instead of light-blue to indicate it is a simulation instrument.

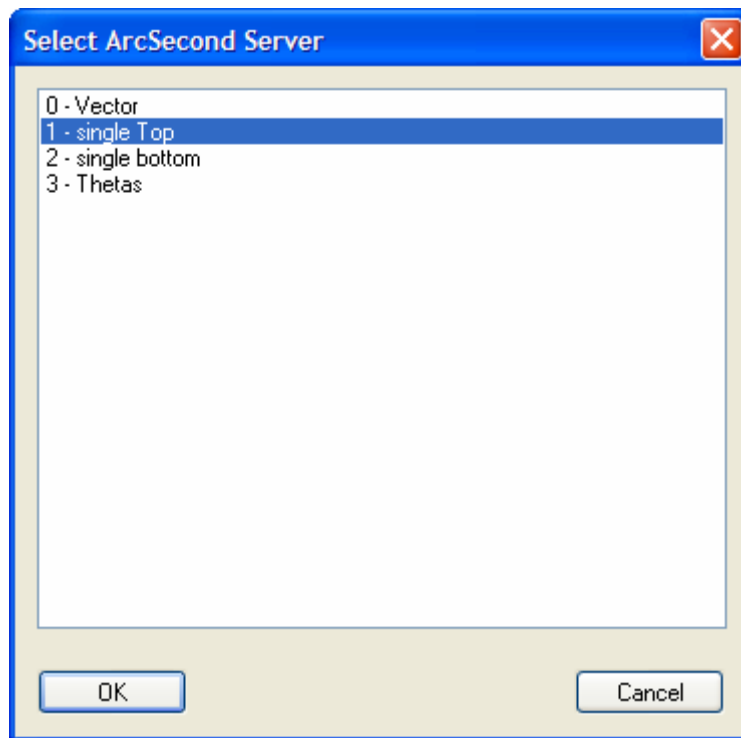
Connecting to 3DiWorkbench

To proceed, it is recommended that 3Di Workbench is running and configured.

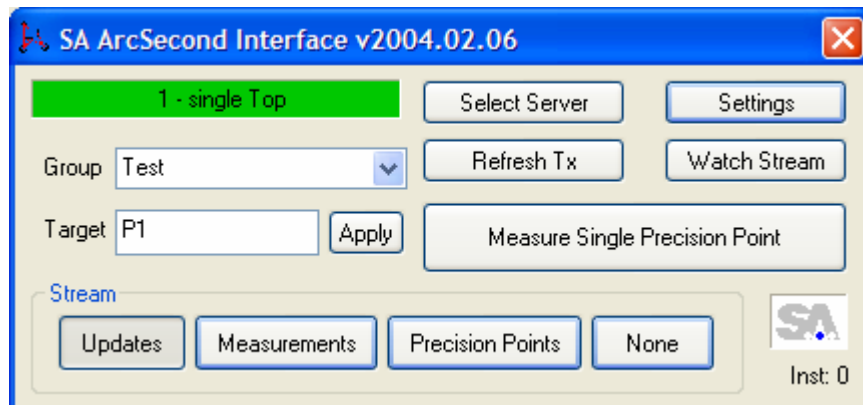
From the Instrument menu, select Run Interface Module and Connect. This will run the SA-AS interface, connect to the ArcSecond network in SA, and display the main interface dialog:



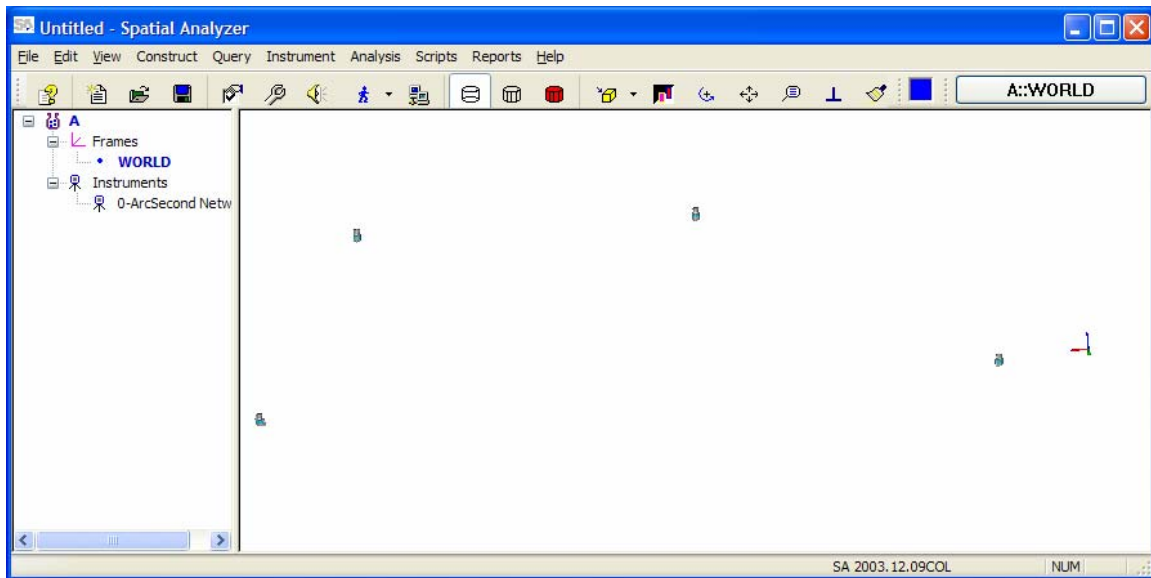
Press the Select Server button and pick a server from the list of those available in workbench:



Once you hit OK, 2 things will happen: The main interface screen will indicate a connection with a green block:



In addition, the transformations for all of the instruments in the workbench configuration will be sent to SpatialAnalyzer. SA will then draw all of the transmitters in their correct location relative to the initial network element that was added:

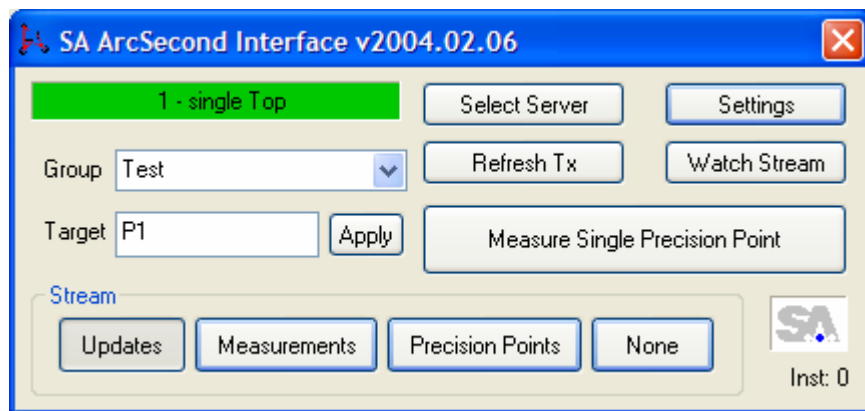


The Input device in workbench should be set to “Connected” and “Streaming”.

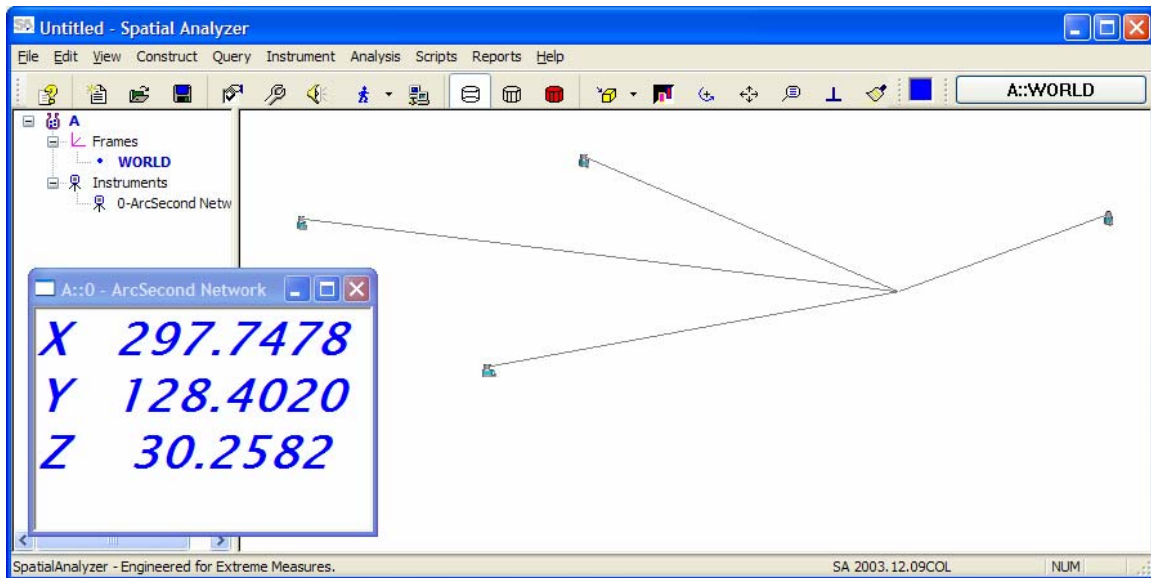
The Server in workbench should be set to “Enabled” and “Precision”.

Sending Updates to SA

Check the Updates option in the Streaming area of the SA-AS interface:



This will continuously update the server position in SA. This will update watch windows as well as draw the measurement rays from the transmitters to the current point:



Watch Stream

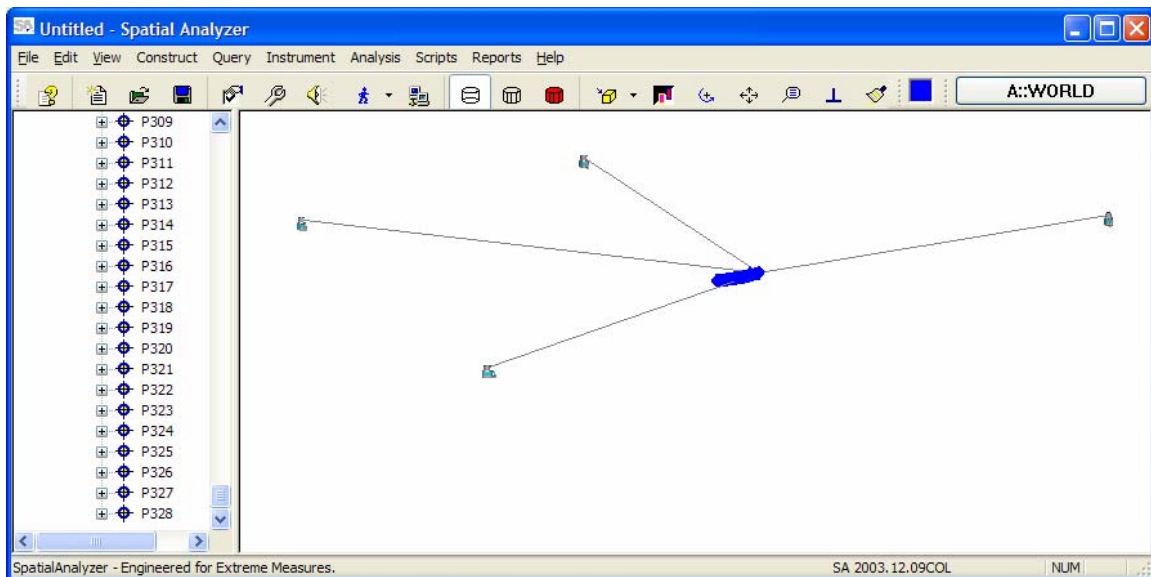
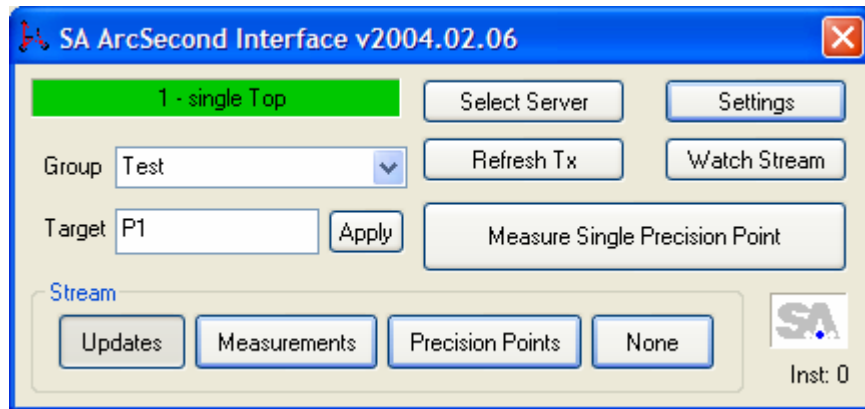
As means of determining which transmitters were used to compute the point a modeless dialog showing the current point coordinates and the transmitter numbers being used to compute its position. An example of the Watch Stream dialog is shown below.



Metric will only light up when the data has metric. The 0,1,2,3 will become 0,_,2,3 if transmitter one is blocked, etc.

Sending Measurements to SA

If you uncheck Updates and check Measurements, you will get a continuous stream of measured points added to the SA database:



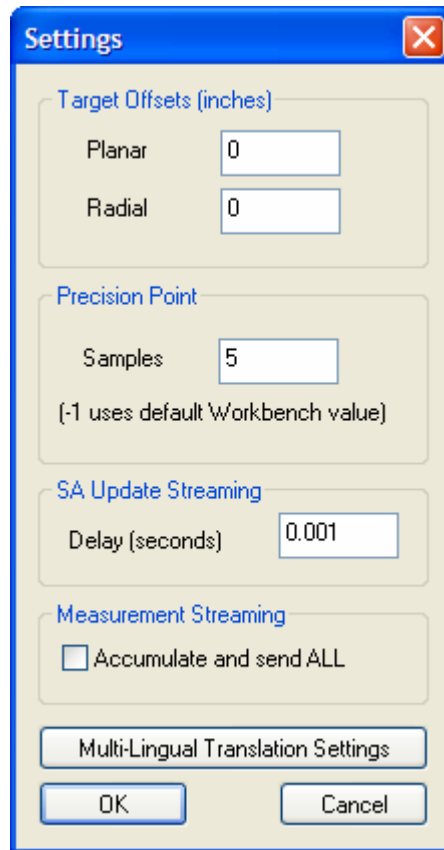
Unchecking the Measurement box will stop the measurement stream.

Measuring Discrete Points from the Interface

At any time, you can press the Measure button and a Workbench “precision point” will be measured. This can occur while updates are streaming, for example, so you can see the current position, then press Measure to actually record a precision measurement.

Interface Settings

If you press the Settings button, the following dialog appears:



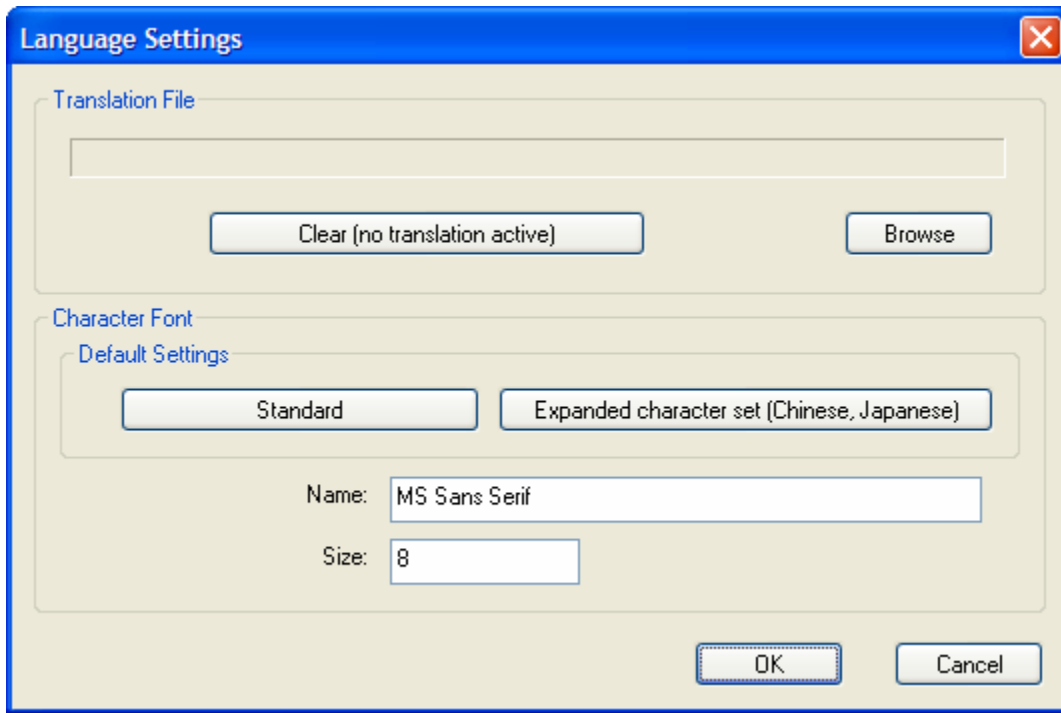
The target offset values are tooling values used in SpatialAnalyzer to properly fit geometric shapes. If you fit a plane, for example, with a 1" tooling ball on the end of the detector rod, the resulting plane will automatically be offset by 0.5" in SA when the fit is performed.

The precision point setting controls the number of samples used in the ArcSecond filtering algorithm to reduce noise. A value of -1 will use the workbench default setting. If you select, 20, for example, 20 measurements will be used in the filter.

The SAUpdate Streaming delay is used to prevent the interface from hammering SA will updates at the raw measurement frequency. This delay allow SA to have time to render the graphics, update watch windows, project points to surfaces, and handle requests from whatever other measurement instruments are connected to the SA job.

Language Settings

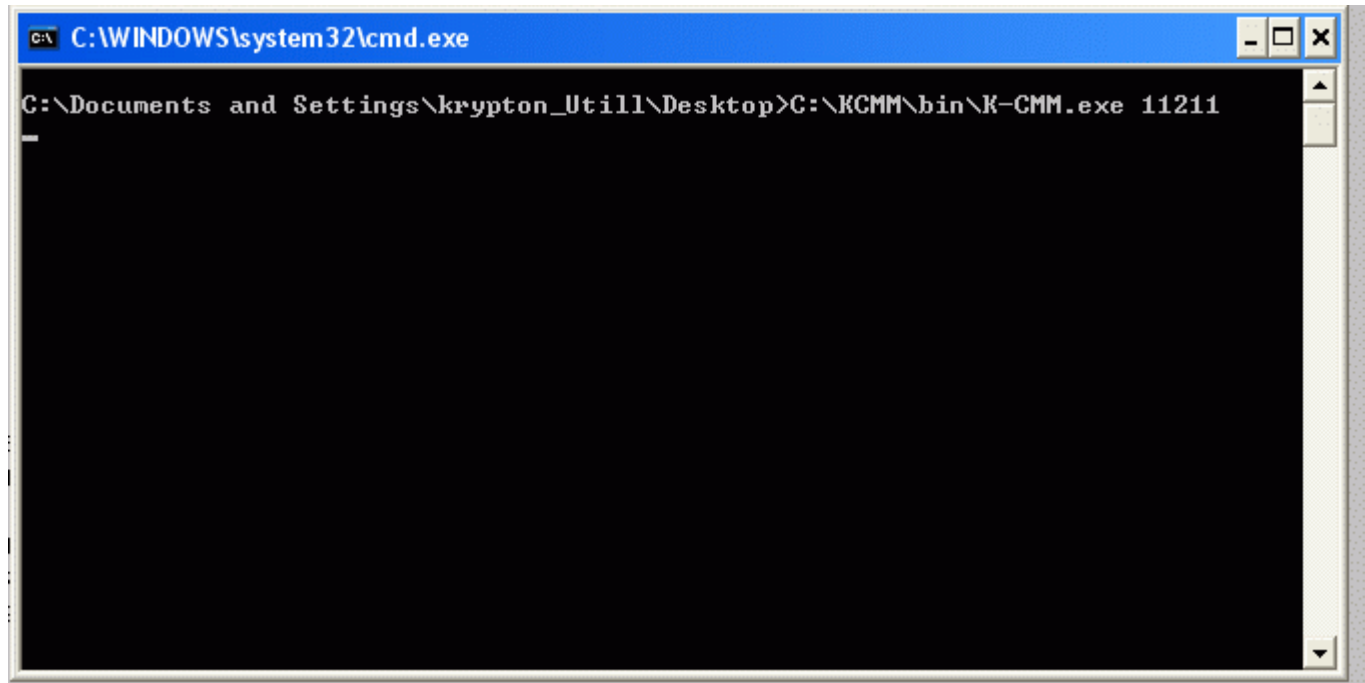
Multi-Lingual translation settings allows you to load language files to translate the entire interface into another language:



NRK has a translation tool that a user may use to enter the appropriate translations. This file is then converted into a .lan file and then may be used in the interface.

KRYPTON – SPACE PROBE INTERFACE

The interface between the KCMM software and NRK's Interface requires a TCP/IP connection. The KCMM software must be setup to provide this mechanism. To do so, start the KCMM software from the command line and specify port information:



A KCMM software update may be necessary to support this feature. Contact Krypton support should a KCMM software update be necessary.

Metris Contact Info:

Interleuvenlaan 86
B-3001 Leuven Belgium
Email: info@metris.com
Tel : +32 16 74 01 01
Fax: +32 16 74 01 02
<http://www.metris.com/>

KCMM Software Script

Be sure to update the KCMM software to use the script located in SA Samples directory (Krypton-Spatial Analyzer.script). The script is shown below:

```
// Interface to New River Kinematics Spatial Analyzer software
// Author: Kristof

initialise
echo("Spatial Analyzer script active");
end

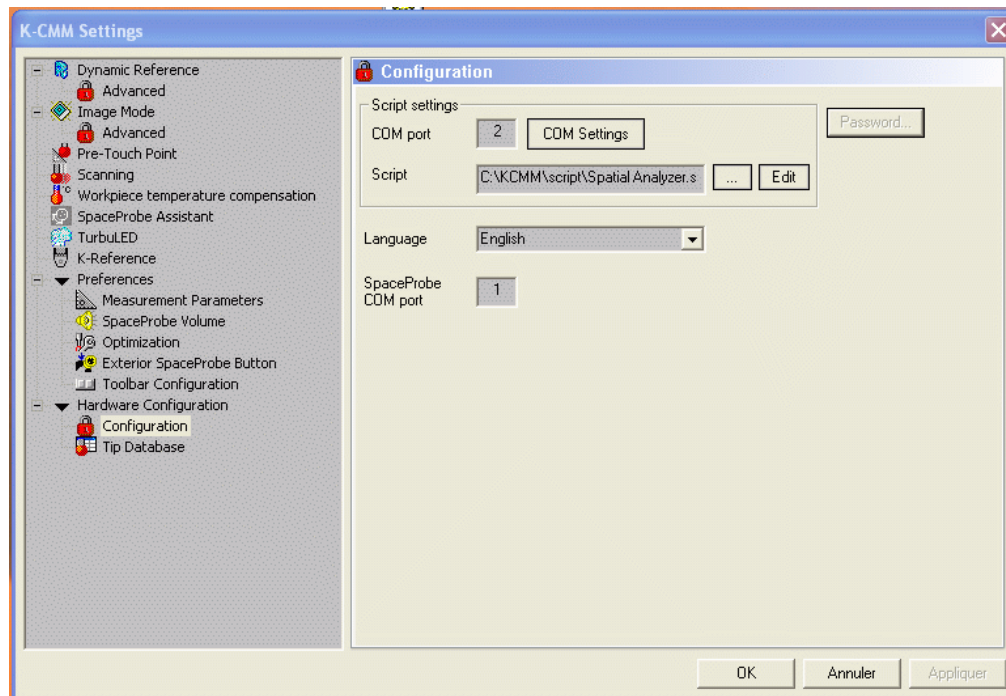
trigger
send("R=%f\n\r",#TipRadius);
echo("R=%f\n\r",#TipRadius);
send("%+05.1f%+09.4f%+09.4f%+09.4f%+09.4f%+09.4f%+09.4f\n\r",1,#TriggerOrigin,#TriggerRotationEulerZYZ);
echo("%+05.1f%+09.4f%+09.4f%+09.4f%+09.4f%+09.4f%+09.4f\n\r",1,#TriggerOrigin,#TriggerRotationEulerZYZ);
wait(50);
end

time(100)
send("R=%f\n\r",#TipRadius);
echo("R=%f\n\r",#TipRadius);
send("%+05.1f%+09.4f%+09.4f%+09.4f%+09.4f%+09.4f\n\r",0,#Origin,#RotationEulerZYZ);
echo("%+05.1f%+09.4f%+09.4f%+09.4f%+09.4f%+09.4f\n\r",0,#Origin,#RotationEulerZYZ);
end

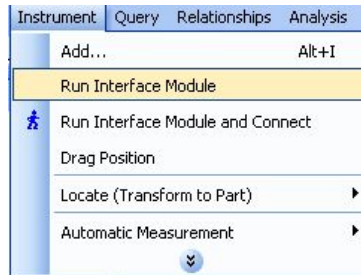
button(1)
send("%+05.1f%+09.4f%+09.4f%+09.4f%+09.4f%+09.4f\n\r",2,#Origin,#RotationEulerZYZ);
end
```

KCMM Settings Configuration

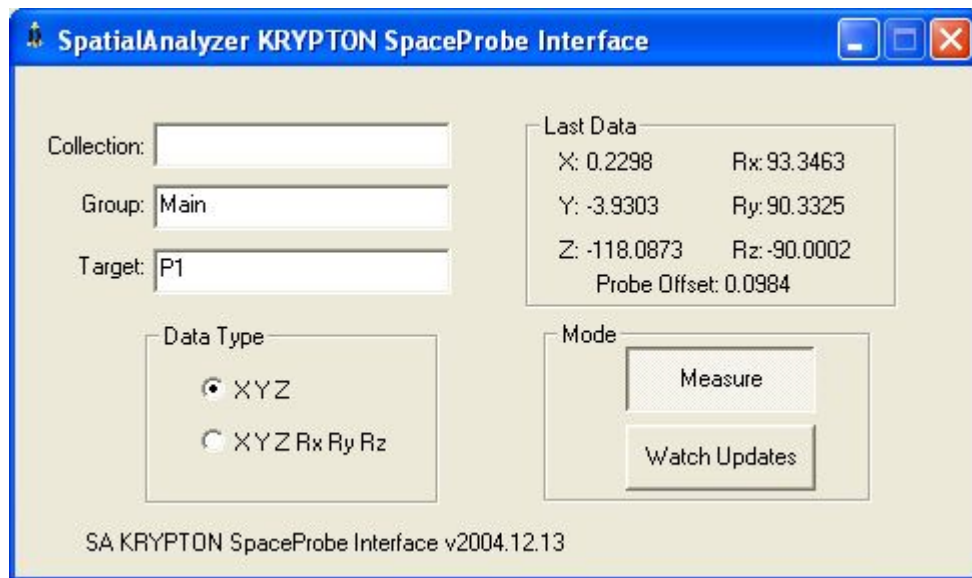
The script is configured via the K-CMM Settings within the KCMM software. Select Hardware Configuration / Configuration to configure the script information.



The first time you run the KRYPTON interface from SA, you need to specify the same TCP/IP port information that the KCMM will be using:



With the proper server name and port information specified, the SA Instrument interface should appear.



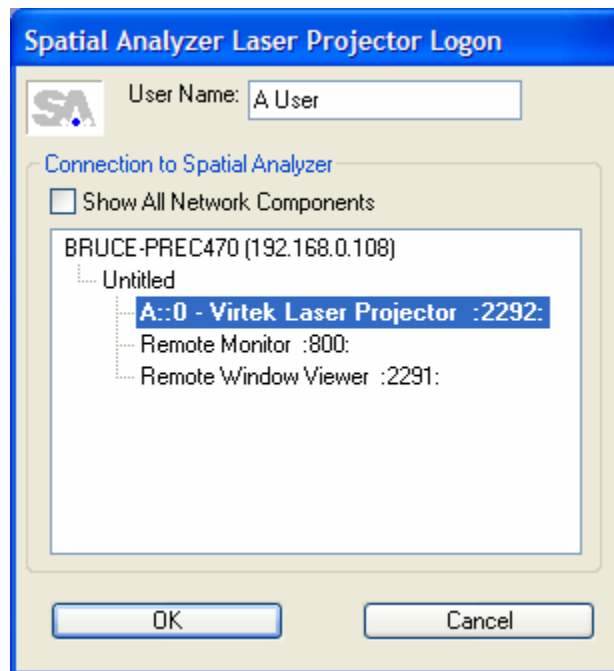
With Mode set to Measure, acquired points will be processed based on the Data Type Mode:
XYZ: will acquire a point to SA with the current Collection / Group / Target naming information.
XYZ Rx Ry Rz: will acquire a frame to SA

With Mode set to Watch Updates, regardless of the Data Type, XYZ updates will be sent to SA.

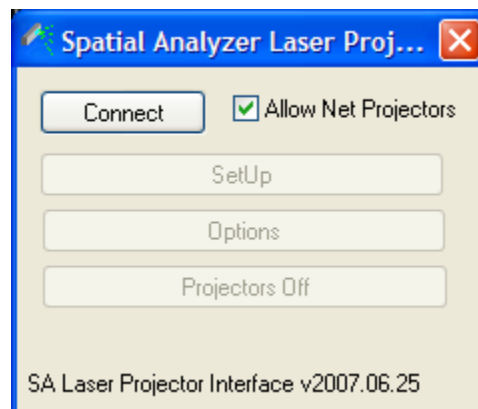
LASER PROJECTOR INTERFACE

First, make sure your projector has current software/firmware installed. For example, at the time of this writing, the onboard projector software version needs to be 5.5.0.1, which is compatible with the current Virtek PDK (projector dev. Kit) v. 1.6.0.17. Also, your pc must have .NET 2.0 installed, available from <http://www.microsoft.com/downloads/>. And finally, if you are running a parallel projector, you must have Virtek's GiveIO parallel driver installed.

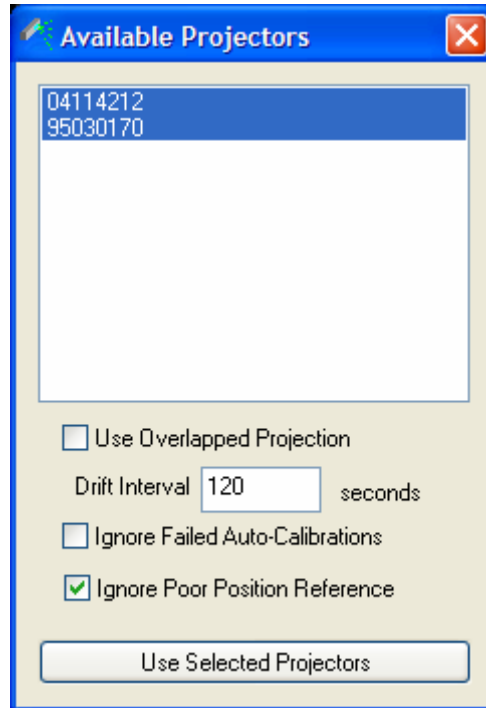
Start SpatialAnalyzer and add a Virtek Laser Projector. Even if you are running several projectors, add only one model in SA. The projector model in SA is capable of representing all your projectors as a single context. Run the Laser Projector interface. As with any instrument, you'll see the Logon dialog first.



Select the Laser Projector as shown. The Laser Projector interface window will be shown.

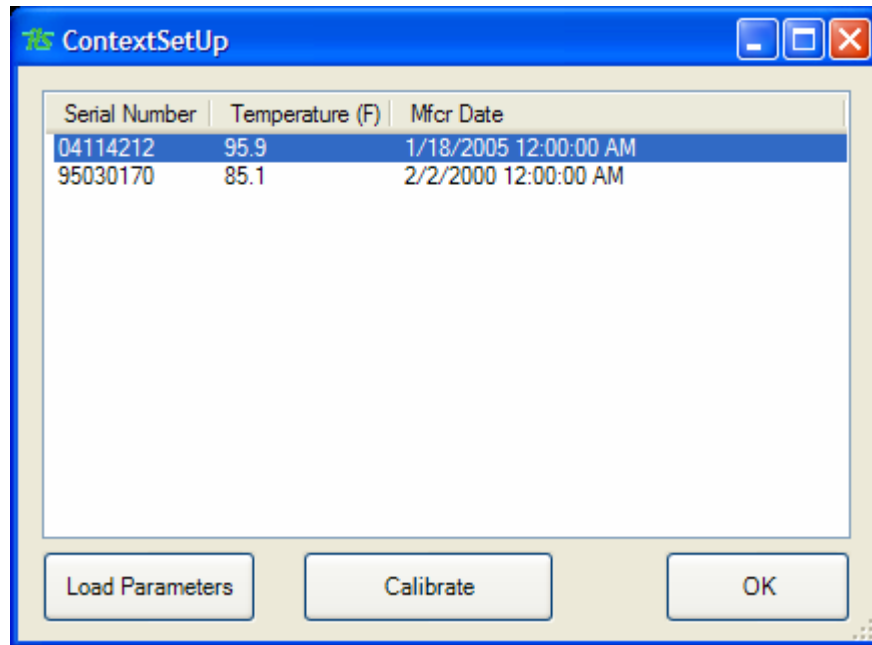


If you wish to connect to any network projectors, check 'Allow Net Projectors'. Press 'Connect' to search for available projectors. Once any projector is found, you'll here a sound and the 'SetUp' button will become enabled. Press 'SetUp' when enabled. The Available Projectors/Configuration window will be shown, displaying the serial number(s) of all projectors found.

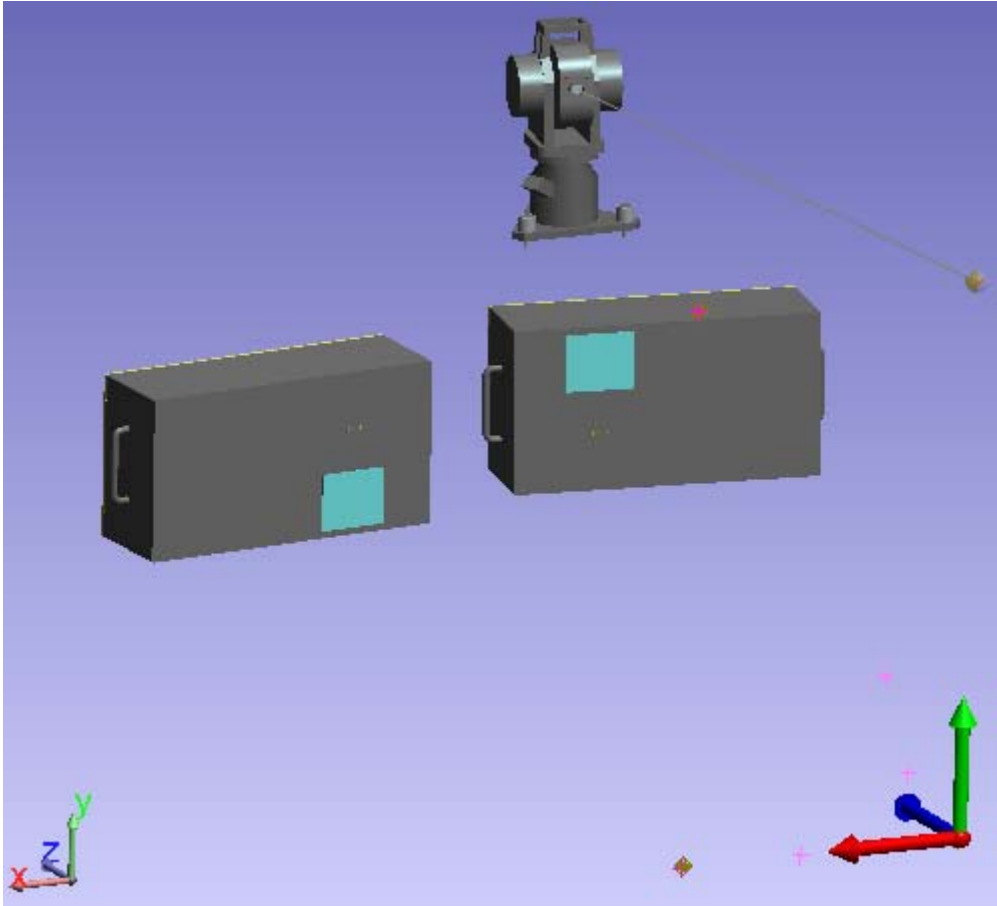


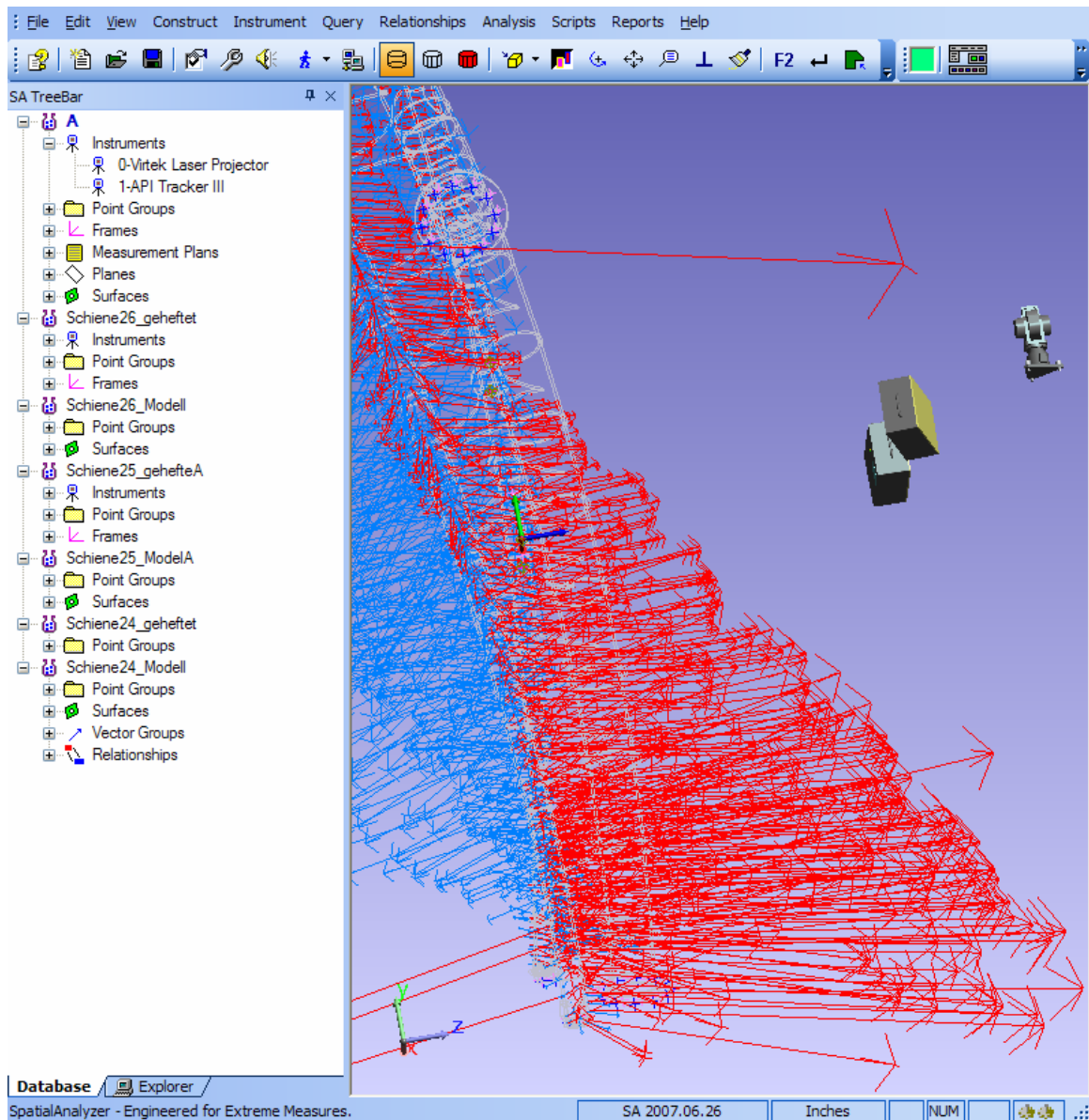
If there is only one available projector, the 'Use Overlapped Projection' check will have no effect. If more than one projector is to be used, then checking 'Use Overlapped Projection' will ensure that all projectors attempt to project all parts of all projected entities. Otherwise, the projectors will share the load by automatically clipping projected entities. As long as this Available Projectors window is up, any network projectors that become available pop in. NOTE: Ignore Poor Position Reference is checked in this case. This forces calibration to succeed, even in demo type situations where calibration point locations may not be ideal.

Select the projector(s) you wish to use, and press 'Use Selected Projectors'. The Context Set Up window will appear.

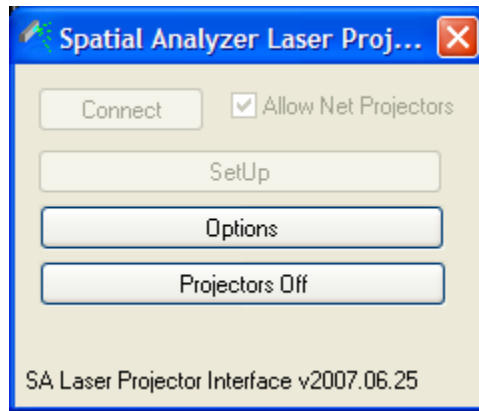


This is where the final setup/calibration takes place. For modern network projectors, the projector parameters are persisted on the projector. For older parallel projectors, the parameters are persisted on the pc. If your projector needs to have the parameters loaded, the 'Calibrate' routine will let you know. If you are in doubt, just press the 'Calibrate' button. If calibration can occur, it will, if not, you'll be prompted for the *.cal Virtek projector parameters file for your unit. If you know parameters are needed, you can press 'Load Parameters' and locate the parameters file. Press the 'Calibrate' button. You'll be prompted for the Virtek calibration file (ASCII file containing sets of Cartesian coordinates and corresponding galvanometer coordinates). Once calibration is complete, press 'OK' and the projectors are ready to use, or OK without running a calibration and calibrate from SA (described below). Once calibrated, the projector models in SA will match reality.

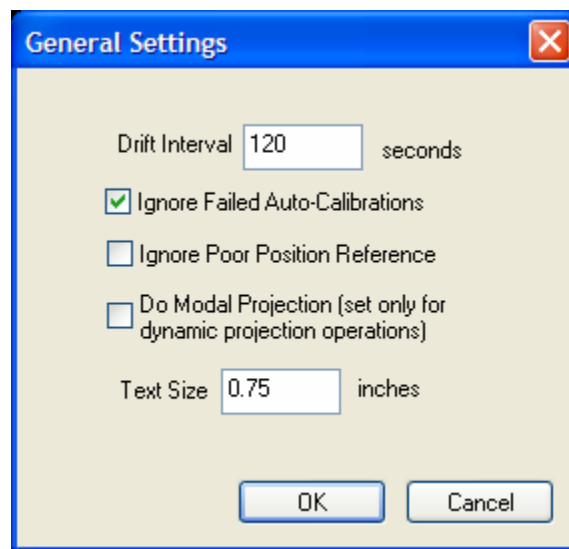




Back in the main dlg, the new Options button becomes enabled once there is a context, so user can edit general settings even after editing is up and going. The Projectors Off button is simply a way to toggle projection of the current entity set.



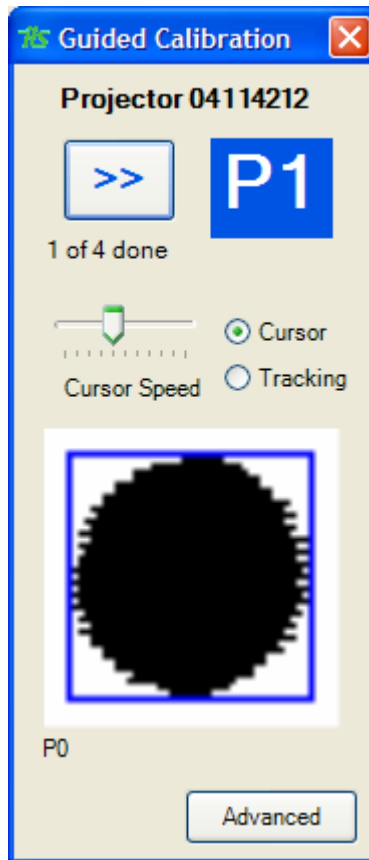
Options button pops this...



Any time a **Watch Window** is run, all calibrated projectors in the job will be used to project the arrow showing the direction, and the magnitude distance to the watched point or object. This is a good case for leaving 'Use Overlapped Projection' (described above) unchecked, so that only the projector in the best position will be used to project the arrow and the magnitude text.

In the Spatial Analyzer Instrument menu, select 'Laser Projector'. You'll see 3 options:

Calibration – This is a point based calibration (no galvo coordinates) which can be performed directly on points in SA. When this is used, you'll see the Guided Calibration window.



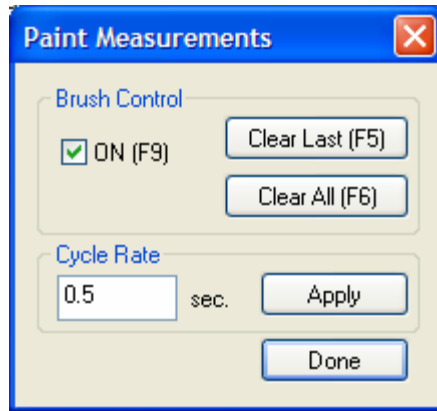
This guides you through locating each point with the mouse, or tracking a reflective target. The graphic shows the returned reflection from each target, to check/ensure that each has a reasonable angle of attack. The 'Advanced' button provides several additional options, such as selecting a saved Virtek cal file (with galvo coordinates), as well as creating a Virtek cal file.

Projection – This is the primary function. Select Projection, and you'll be prompted to select objects (points, geometries, etc.) to project. So long as the projector has a clear shot, the objects will be projected.

Paint Measurements Dynamically – This provides a projection which follows the path of an instrument target's center point as it is moved through the projector's usable frustum.

To use this, make certain that 'Do Modal Projection' is checked in the projector Options. Then, you must locate the instrument and projector in a common reference frame. For the projector, you can simply calibrate to a file, then import that (ascii) file into SA. You can then measure the calibration locations with the instrument, and do an SA Best Fit, using the calibration points as nominals, to bring the instrument into the projector's coordinate system.

Once co-located, the projector is ready to trace the instrument's target. In the SA Instrument menu, select Laser Projector >> Paint Measurements Dynamically. You will be prompted to select the Laser Projector (you can select from the TreeBar on the left), and then the instrument. Once selections are made, you will see the Paint Measurements dialog...



Now, put the instrument in a scanning mode, and all measurements which can be seen by the projector will be projected.

Brush Control ON (F9) toggles the 'brush'. In other words, the laser is accepting data and projecting while this is checked, and ignoring incoming data when it is unchecked. Each time this control is unchecked and then re-checked, a new 'block' of data begins.

Brush Control Clear Last (F5) removes the last 'block' of data collected while the brush was ON from the projector's display list.

Brush Control Clear All (F6) clears all projection.

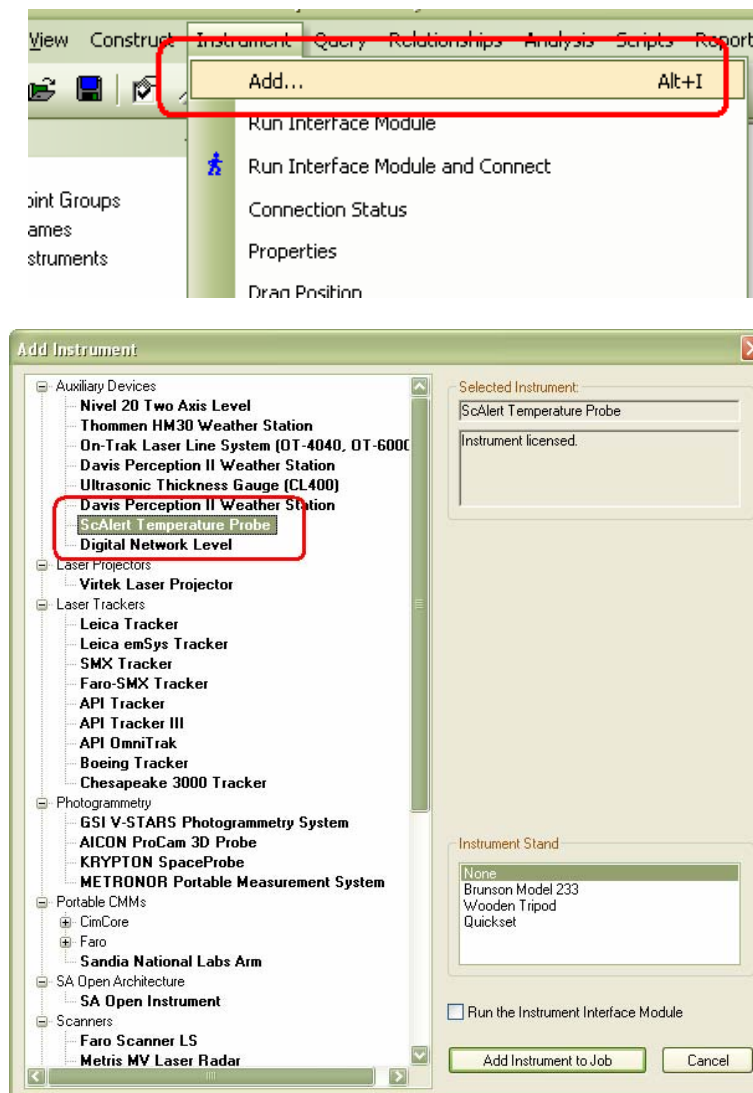
The Cycle Rate is the rate at which the instruments measurements will be sent to the projector. 0.5 seconds is a good default, and the number should be raised only for large point sets.

SCALERT TEMPERATURE MONITORING INTERFACE

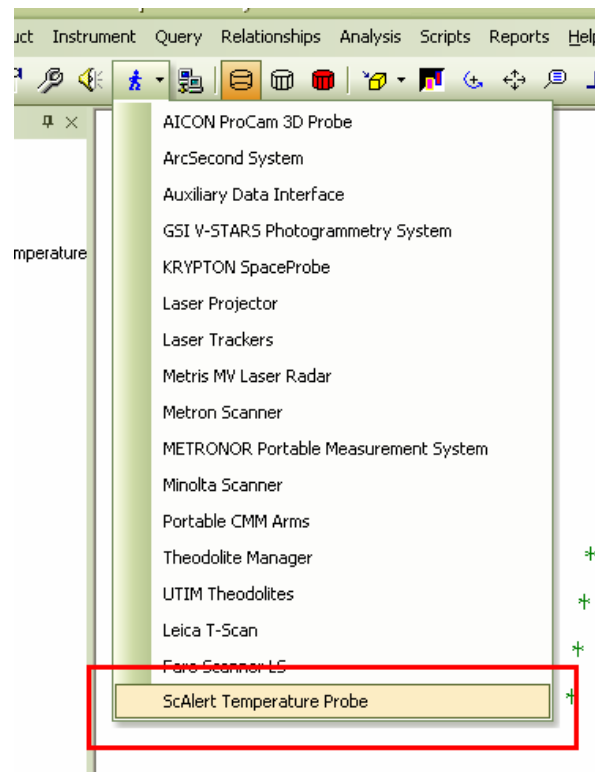
The ScAlert interface allows integrating Spatial Analyzer with the ScAlert wireless scale monitoring system. The current interface supports temperature monitoring using thermistors which communicate with the ScAlert base unit. For more information on the ScAlert device see <http://www.4gmetrology.com>.

Basic Setup and Monitoring

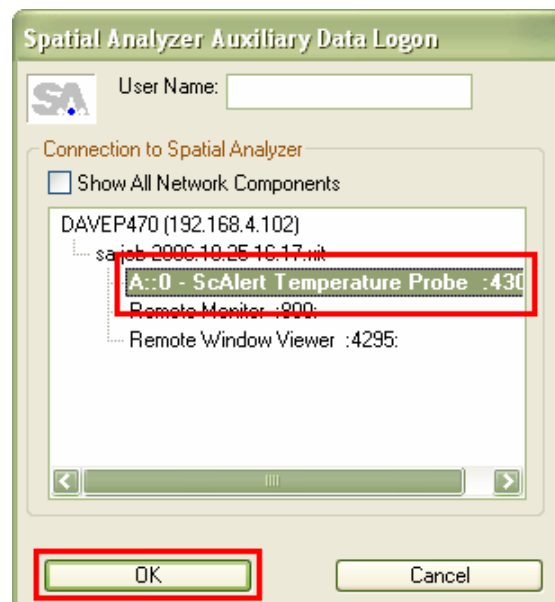
To use an ScAlert device with Spatial Analyzer, add an ScAlert instrument to the current job. Select Instrument>>Add and then choose the “ScAlert Temperature Probe” instrument within the “Auxiliary Devices” group.



Once the instrument has been added to the current job, use the Run Instrument Interface toolbar drop-down to launch the ScAlert interface application.



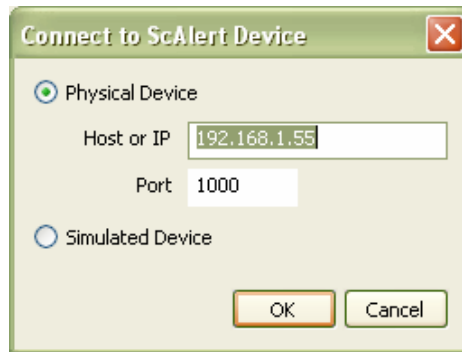
The ScAlert interface application will start and prompt you to log into Spatial Analyzer. Pick the instrument which was added to the job above and then click OK.



In the ScAlert interface window, click on the Connect toolbar button.



The connection dialog will be displayed. Here you can choose to connect to a physical ScAlert device or to a simulated device. To connect to a physical ScAlert device, you'll need to enter the correct IP address (or hostname) and port number. The default values match the device factory default settings.



If a connection is successfully established, the window should look similar to the below (the below was taken while using the simulator).

The screenshot shows the ScAlert application window. It has a menu bar (File, Edit, Instrument, Watch List, View, Help) and a toolbar with icons for File, Edit, Instrument, Watch List, View, and Help. The main content area is divided into two sections: 'Sensor States' and 'Watch List'.

Sensor States

Sensor	Collection	Group	Target	Age (secs)	Temp	Avg Temp
Sim01				1	46.7889 C	X
Sim02				1	39.0111 C	X
Sim03				1	31.9167 C	X
Sim04				1	23.8389 C	X
Sim05				1	21.5000 C	X
Sim06				1	40.4000 C	X
Sim07				1	45.8389 C	X
Sim08				1	42.6778 C	X
Sim09				1	21.3722 C	X
Sim10				1	39.0556 C	X
Sim11				1	19.0889 C	X
Sim12				1	17.7667 C	X
Average					32.4380 C	

Watch List

Temp Source	Axis	Material	Ref Temp	Current Temp	Nominal	Change	Allowable

At the bottom of the window, the status bar displays: SerNum: FF5C000000180A00 | Avg: 32.44 C StdDev: 10.61 C | Local Clock: 12/15/06 12:13:48 | ScAlert Clock: 12/15/06 12:13:48

Basic Configuration

The Sensor column displays the name which has been assigned to each sensor. The Age column indicates how old the current reading is in seconds, and the Temp column indicates the most recent temperature reading. When a sensor row is selected in the list, the sensor serial number is also displayed in the window status area.

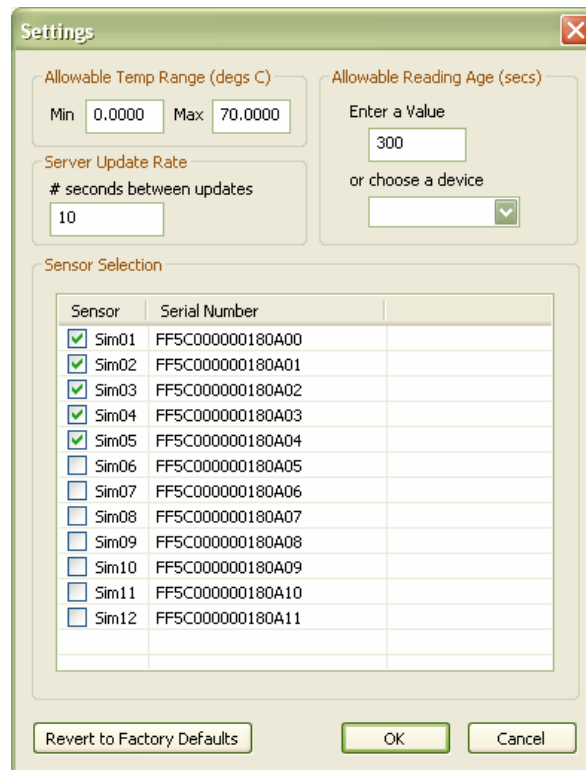
In addition to listing the sensors connected to the ScAlert server device, an Average row is displayed in the list. This row provides the average temperature value, computed from all sensors which show an "X" in the Avg Temp column. Additionally, the average value and the standard deviation are displayed in the window status bar area.

By default, all sensors connected to the ScAlert server device are shown in the list and are included in the average/std deviation calculation. The following will explain how to customize which sensors are shown and which sensors are included in the average/std deviation calculation.

To configure the application settings, click on the settings toolbar button.



This will open the Settings dialog.

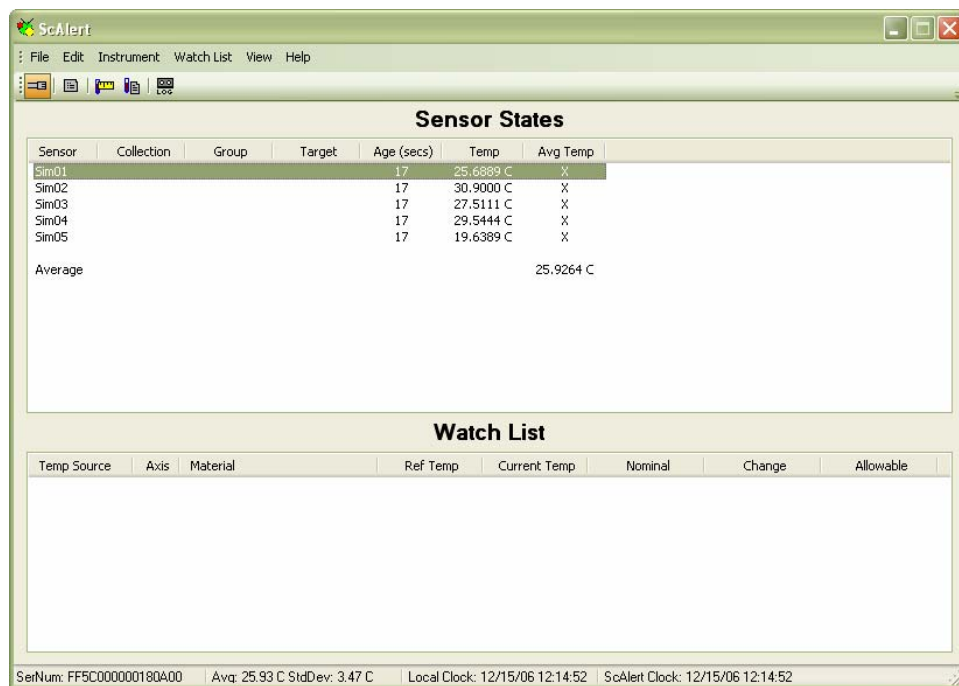


The settings dialog allows selection of allowable temperature range and reading age, and the frequency of updates from the ScAlert server device.

The settings dialog also allows selection of which sensors are displayed in the main window. The Sensor Selection list shows all sensors connected to the ScAlert device and the checkboxes can be used to designate which sensors are of interest and should be displayed in the main window. If no sensors are checked, then all sensors are displayed in main window.

Finally, the Revert to Factory Defaults button is provided to allow clearing out all settings changes/sensor selections, etc. and reverting to a clean pre-configured state.

The screenshot below shows only the first 5 sensors, since those were the only ones checked in the settings dialog.



The screenshot shows the ScAlert application window. The 'Sensor States' section displays a table with the following data:

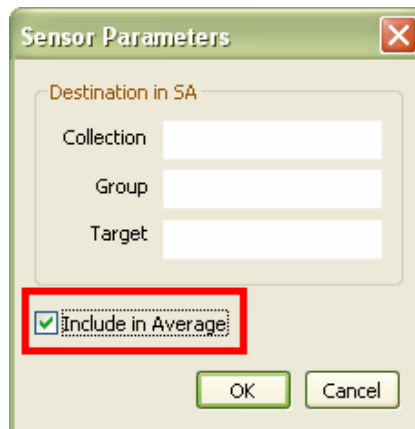
Sensor	Collection	Group	Target	Age (secs)	Temp	Avg Temp
Sim01				17	25.6889 C	X
Sim02				17	30.9000 C	X
Sim03				17	27.5111 C	X
Sim04				17	29.5444 C	X
Sim05				17	19.6389 C	X
Average					25.9264 C	

The 'Watch List' section is currently empty. The status bar at the bottom shows: SerNum: FF5C000000180A00, Avg: 25.93 C StdDev: 3.47 C, Local Clock: 12/15/06 12:14:52, and ScAlert Clock: 12/15/06 12:14:52.

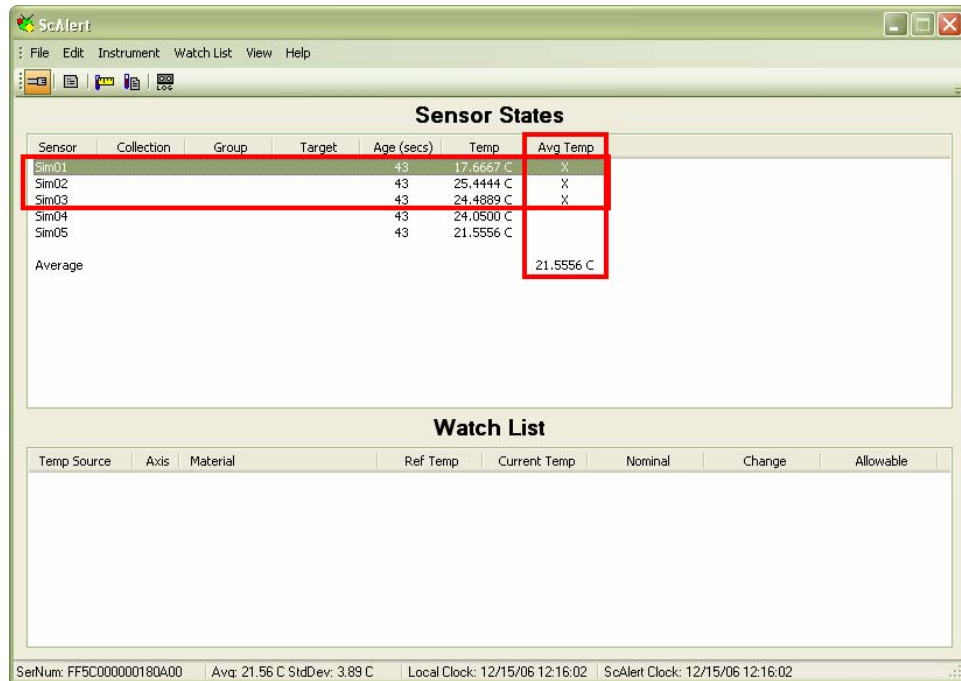
To control which sensors are included in the Average/Std Deviation calculation, for each sensor which should be included, select that sensor and click the Sensor Parameters toolbar button (or double-click the sensor item in the list).



This will open the Sensor Parameters window. In this window, use the checkbox to indicate that this sensor should be included in the average. If you have not manually added any sensors to the average, then the default is to include all listed sensors. Once you manually add one or more sensors to the average, then only those will be included in the average. The "X" in the Avg Temp column ensures that it is clear which sensors are currently being included in the average/std deviation calculations.



Manually selecting the first 3 sensors to be included in the average results in a display similar to the below.



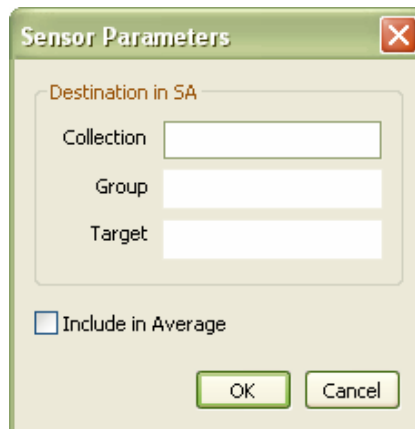
Repeating the above process for the 3 sensors currently in the average, but this time unchecking the “Include in Average” checkbox, will revert to the state where all sensors are included in the average.

Sending Data to SA

The Collection/Group/Target columns are used when sending data to SA. A Collection/Group/Target must be assigned to a row in order for that row to be sent over to SA. In order to do this, select a sensor (or the average) and click the Sensor Parameters toolbar button.



The Sensor Parameters window will be displayed.



The Destination in SA area is used to specify the Collection/Group/Target where the data for this item will be sent.

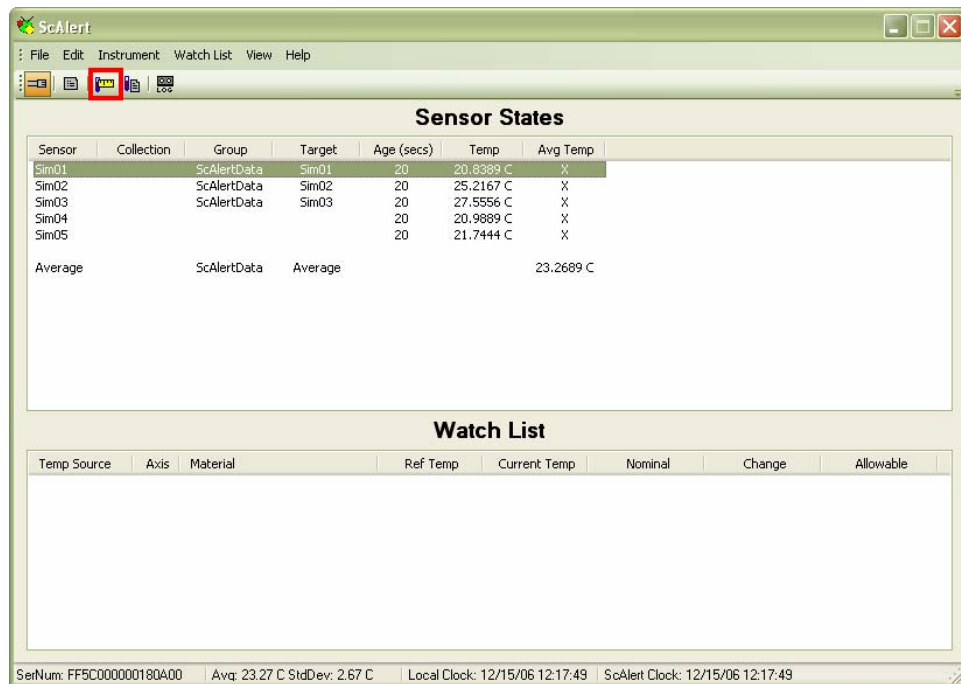
The Included in Average checkbox indicates whether or not this item should be included in the average/std deviation calculation. Note that if this checkbox has not been checked for any items (the default initially), then all sensors will be included in the average. As soon as one or more sensors are manually selected into the average, only those selected are included in the average/std deviation calculation. An "X" will be displayed in the Avg Temp column for those sensors which are currently included in the average/std deviation calculation.

The below shows the first 3 sensors with Group/Target fields configured and included in the average. Only these first 3 sensors are now used to compute the average/std.

deviation. Also, the Average item has been assigned a Group/Target which will allow it to also be sent over to SA.

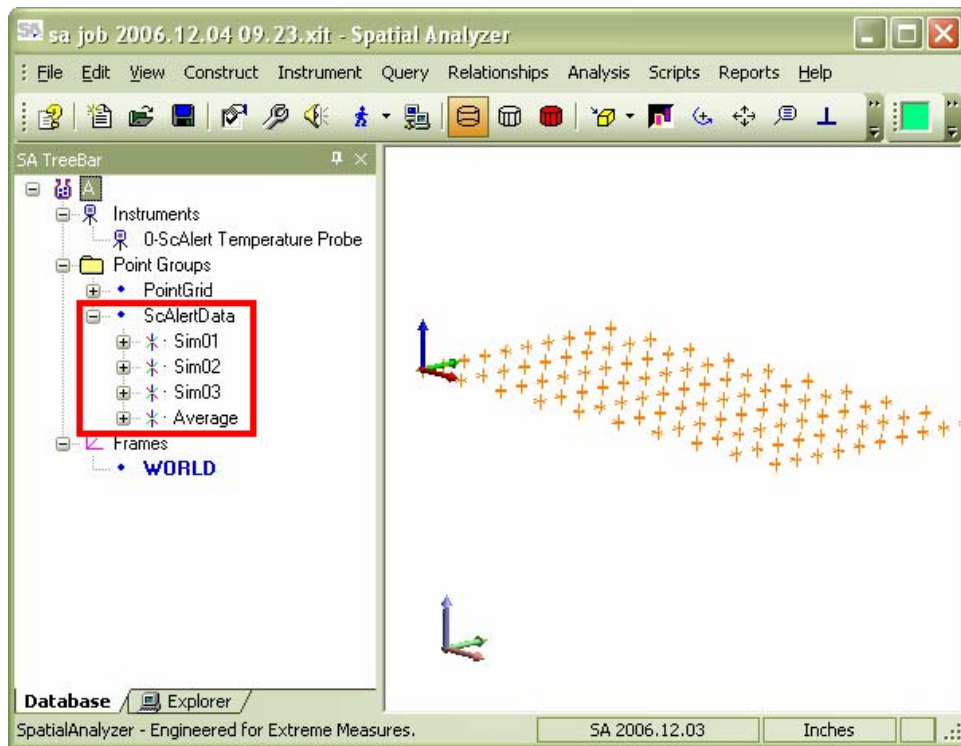


To send data to Spatial Analyzer, click on the Measure button on the toolbar. All items in the list which have a Group/Target assigned will be sent to SA.



Viewing Data in SA

In SA, you should see the received data in the job database tree window as shown below.



Double-clicking on one of the sensor groups (Sim01 for example), shows the reported temperature value with units.

Point Information [X]

Col: A
 Group: ScAlertData
 Target: Sim01

X: 0.0000
 Y: 0.0000
 Z: 0.0000

[Edit Values]

[Tolerance]

[Measurement Details (1)]

Notes:

Targeting Offsets

0.0 0.0

[Apply offsets to all measurements]

Name	Value	Units
Sim01	26.033333	C

[OK]

In the above window, clicking the Measurement Details button, then double-clicking on one of the measurements, provides additional details about the measurement.

Auxiliary Data Measurement [X]

Instrument: A::0 - ScAlert Temperature Probe
 Target: A::ScAlertData::Sim01
 Time: 12/04/06 10:27:34
 User:

Info:

name=Sim01
 sensor number=1
 serial number=FF5C000000180A00
 type=Thermistor
 age of reading=38 secs

☒ Use In Calculation

Name	Value	Units
Sim01	26.033333	C

[OK] [Cancel]

From the SA job database tree, double-clicking on the average group (Average in this example), displays the average and standard deviation.

The dialog box titled "Point Information" displays the following information:

- Col: A
- Group: ScAlertData
- Target: Average
- X: 0.0000
- Y: 0.0000
- Z: 0.0000

Buttons: Edit Values, Tolerance, Measurement Details (1)

Notes:

Targeting Offsets:

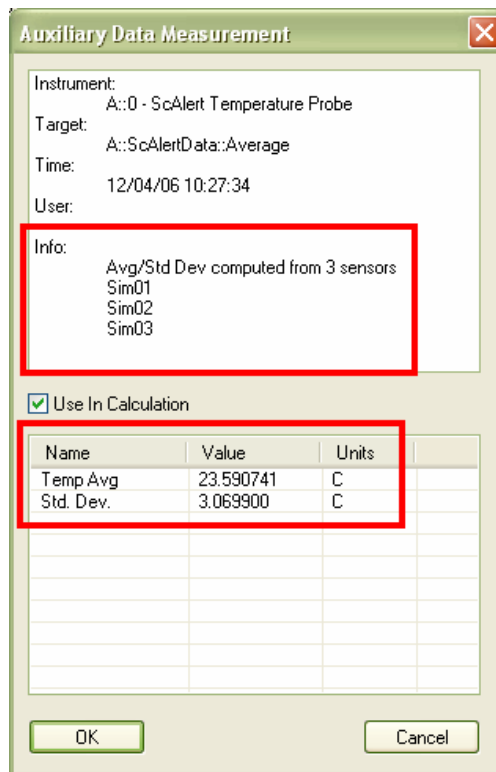
0.0 0.0

Apply offsets to all measurements

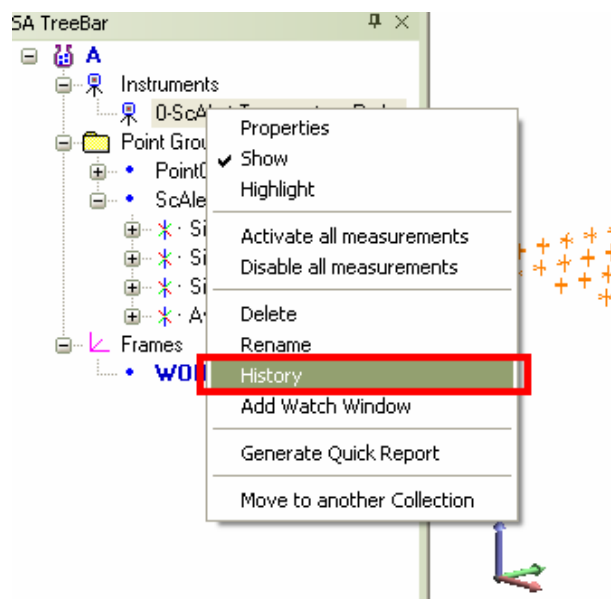
Name	Value	Units
Temp Avg	23.590741	C
Std. Dev.	3.069900	C

OK

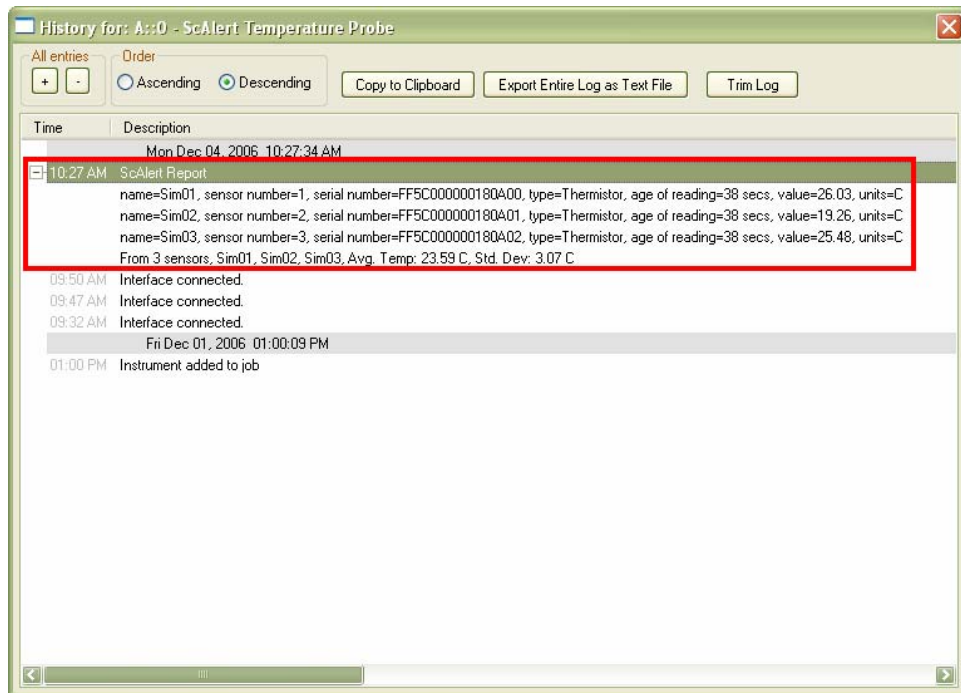
Clicking on Measurement Details, then double-clicking on a measurement will provide info on which sensors were used to compute the average and standard deviation.



In addition to sending data to the specified Collection/Group/Target, all data is also added to the instrument log in SA. Right-click the instrument in the job database tree and select History.



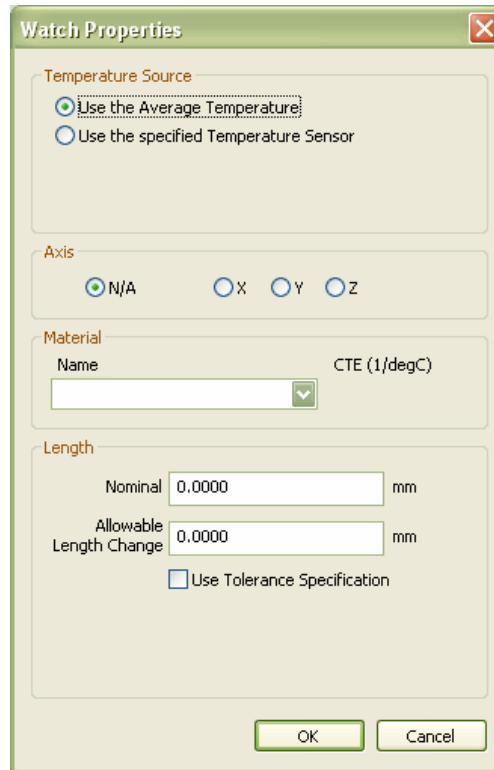
In the instrument history window, expand the ScAlert Report item. This will display the sensor information as well as the average/std deviation information.



Watch List

The Watch List is used to allow monitoring of temperature changes in terms of material expansion/contraction. The material changes are monitored relative to configured limits and the user is notified when those limits are exceeded.

To start using the Watch List, select Watch List>>Add. This will open the Watch Properties window.



The 'Watch Properties' dialog box is a standard Windows-style window with a title bar and a close button. It contains four main sections: 'Temperature Source', 'Axis', 'Material', and 'Length'. The 'Temperature Source' section has two radio buttons, with 'Use the Average Temperature' selected. The 'Axis' section has four radio buttons, with 'N/A' selected. The 'Material' section has a 'Name' dropdown menu and a 'CTE (1/degC)' label. The 'Length' section has two text input fields for 'Nominal' and 'Allowable Length Change', both set to '0.0000' with 'mm' units, and a checkbox for 'Use Tolerance Specification' which is currently unchecked. 'OK' and 'Cancel' buttons are at the bottom right.

Section	Property	Value / Option
Temperature Source	Use the Average Temperature	Selected
	Use the specified Temperature Sensor	Unselected
Axis	N/A	Selected
	X	Unselected
	Y	Unselected
	Z	Unselected
Material	Name	[Empty dropdown]
	CTE (1/degC)	[Empty input]
Length	Nominal	0.0000 mm
	Allowable Length Change	0.0000 mm
	Use Tolerance Specification	Unchecked
	Buttons	OK / Cancel

From this window, the temperature source can be selected as either the Average or a specified sensor. The axis can either be N/A or set to X, Y, or Z and will be displayed along with the watch in the main window. Choosing a Material type sets the coefficient of thermal expansion (CTE) used in the calculations. The Length settings control the length changes which are allowed before the watch goes into an alert state. The allowable length change can be specified directly, or in terms of minimum job tolerance and % tolerance allowed for scale.

Once a Watch item has been added to the list, it will be displayed, along with the nominal length, the current length change (calculated using the Ref Temp and the Current Temp), and the allowable length change.

The screenshot shows the ScAlert software window. The 'Sensor States' table lists five sensors (Sim01-Sim05) with their respective temperatures and an average temperature of 23.1911 C. The 'Watch List' table shows a single entry for 'Average' with a nominal length of 10000.0000 mm and a change of 0.3734 mm. The status bar at the bottom indicates the average temperature is 23.19 C and the local clock is 12/15/06 12:19:22.

Sensor	Collection	Group	Target	Age (secs)	Temp	Avg Temp
Sim01	ScAlertData	Sim01	21	19,7222 C	X	
Sim02	ScAlertData	Sim02	21	23,3944 C	X	
Sim03	ScAlertData	Sim03	21	21,8778 C	X	
Sim04			21	23,9222 C	X	
Sim05			21	27,0389 C	X	
Average	ScAlertData	Average				23.1911 C

Temp Source	Axis	Material	Ref Temp	Current Temp	Nominal	Change	Allowable
Average	N/A	Steel	20.0000 C	23.1911 C	10000.0000 mm	0.3734 mm	0.5347 mm

SerNum: FF5C000000180A00 Avg: 23.19 C StdDev: 2.41 C Local Clock: 12/15/06 12:19:22 ScAlert Clock: 12/15/06 12:19:22

By default, the Ref Temp is 20 deg C. The Ref Temp can be set to the Current Temp using Watch List>>Set Reference Temp>>Use Current Temperature. It can also be set back to 20 deg C using this menu.

If the change in length due to temperature change is greater than or equal to 80% of the allowable length change, the watch item will turn yellow as shown below.


This screenshot is similar to the first one, but the 'Average' entry in the 'Watch List' table is highlighted in yellow, indicating that the change in length (0.4173 mm) is greater than or equal to 80% of the allowable length change (0.5000 mm). The status bar at the bottom shows the average temperature is 23.57 C and the local clock is 12/15/06 12:20:09.

Sensor	Collection	Group	Target	Age (secs)	Temp	Avg Temp
Sim01	ScAlertData	Sim01	17	23,4833 C	X	
Sim02	ScAlertData	Sim02	17	22,0611 C	X	
Sim03	ScAlertData	Sim03	17	21,4000 C	X	
Sim04			17	23,5500 C	X	
Sim05			17	27,3389 C	X	
Average	ScAlertData	Average				23.5667 C

Temp Source	Axis	Material	Ref Temp	Current Temp	Nominal	Change	Allowable
Average	N/A	Steel	20.0000 C	23.5667 C	10000.0000 mm	0.4173 mm	0.5000 mm

SerNum: FF5C000000180A00 Avg: 23.57 C StdDev: 2.06 C Local Clock: 12/15/06 12:20:09 ScAlert Clock: 12/15/06 12:20:09

If the change in length due to temperature change is greater than or equal to 100% of the allowable length change, the watch item will turn red as shown below.



The screenshot shows the ScAlert application window. It has a menu bar (File, Edit, Instrument, Watch List, View, Help) and a toolbar. The main area is divided into two sections: 'Sensor States' and 'Watch List'.

Sensor States Table:

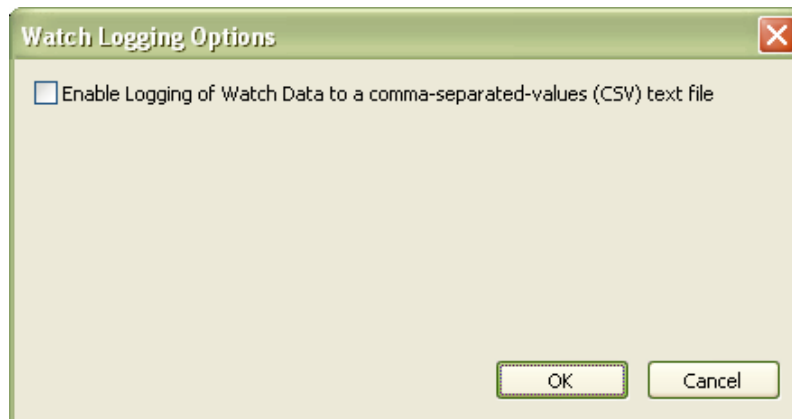
Sensor	Collection	Group	Target	Age (secs)	Temp	Avg Temp
Sim01		ScAlertData	Sim01	2	19.8500 C	X
Sim02		ScAlertData	Sim02	2	26.2444 C	X
Sim03		ScAlertData	Sim03	2	24.9333 C	X
Sim04				2	25.4778 C	X
Sim05				2	27.0500 C	X
Average		ScAlertData	Average			24.7111 C

Watch List Table:

Temp Source	Axis	Material	Ref Temp	Current Temp	Nominal	Change	Allowable
Average	N/A	Steel	20.0000 C	24.7111 C	10000.0000 mm	0.5512 mm	0.5000 mm

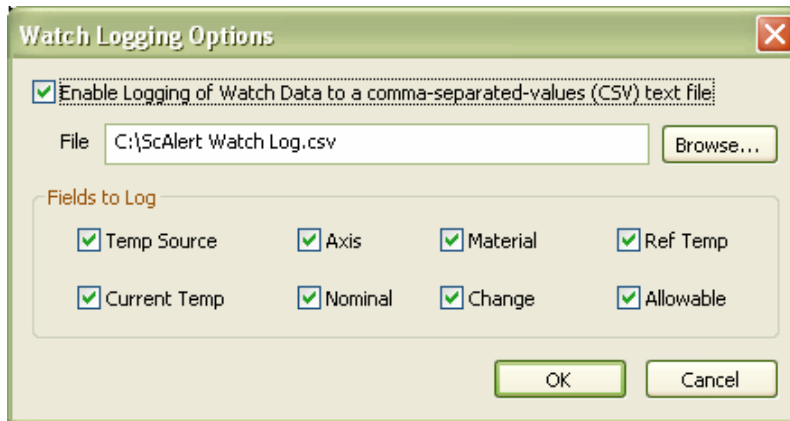
The 'Watch List' table has a red background, indicating a warning condition. The status bar at the bottom shows: SerNum: FF5C000000180A00, Avg: 24.71 C StdDev: 2.53 C, Local Clock: 12/15/06 12:20:34, ScAlert Clock: 12/15/06 12:20:34.

The contents of the Watch List can be logged to a comma-separated-values (.csv) text file if desired. To enable logging, select “Watch List>>Logging Options” from the menu to open the logging options window.



The 'Watch Logging Options' dialog box is shown. It has a title bar with a close button. Inside, there is a checkbox labeled 'Enable Logging of Watch Data to a comma-separated-values (CSV) text file'. The checkbox is currently unchecked. At the bottom right, there are 'OK' and 'Cancel' buttons.

Logging is disabled by default, but can be enabled by checking the “Enable Logging...” checkbox. Once checked, the rest of the logging options appear and can be used to control where the log file is stored as well as which fields are included in the log.

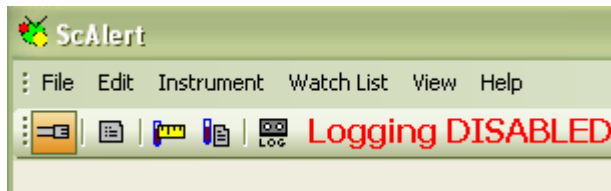


When logging is enabled, the log file is written to (appended) every time the ScAlert sensor states are retrieved from the server. This rate can be controlled in the main settings dialog. The log file can be opened in any text editor, or if double-clicked, will open in Excel.

Logging can also be controlled using the log toolbar button shown below.

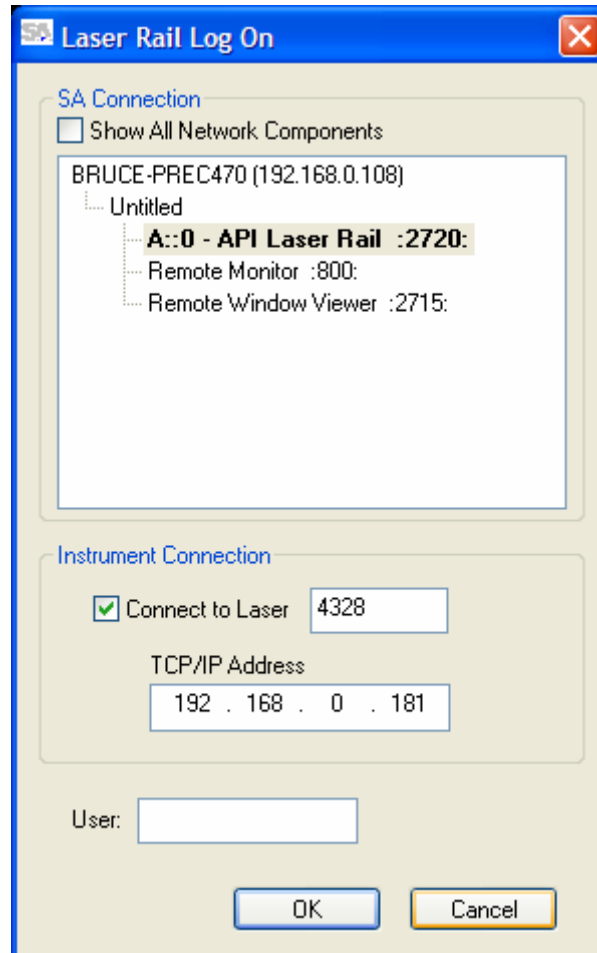


The toolbar button indicates whether logging is currently enabled or disabled and by clicking it, allows you to quickly turn logging on and off.

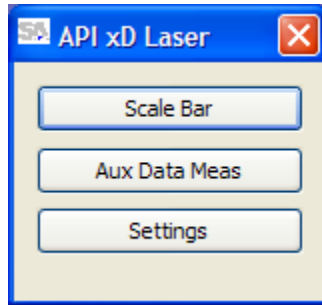


API LASER RAIL INTERFACE

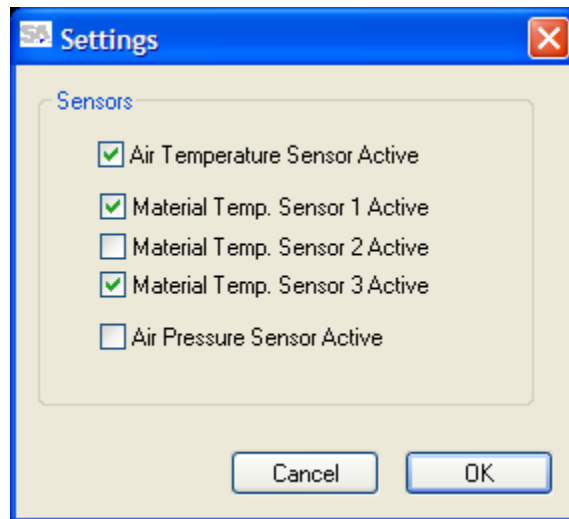
The XD Laser is an Auxiliary Device in SA. To use the interface, add the Laser Rail to SA like any other instrument. Upon running the interface, the usual instrument log on dialog will appear with parameters specific to the XD.



Select the laser (prm file name), and the ip address of the laser, and press OK. Note: You may also simulate the laser rail simply by un-checking 'Connect to Laser' and pressing ok. Once connection and initialization are complete, the main interface window will appear.



To set up the XD, press the Settings button for options.



Adjust to desired settings, and press OK. The selected settings will be persisted for the next run of the laser.

Now, scale bars can be created, or Auxiliary data measurements can be made. To create scale bars, press the Scale Bar button for the Scale Bar interface.

Scale Bar

A

Collection:

Group:

Target:

B

Collection:

Group:

Target:

Add ScaleBar to SA

Length

As with other instrument interfaces in SA, the Collection will default to “A” if left blank. Group and target names can be incremented by pressing the associated ‘+’ buttons. To create a scale bar, simply adjust names as desired, move the target to desired positions, and Measure each one. Whenever there is enough data to make a scale bar, the results to be sent to SA will be shown in the ‘Add ScaleBar to SA’ section of the interface window.

Scale Bar

A

Collection:

Group:

Target:

154.317407

B

Collection:

Group:

Target:

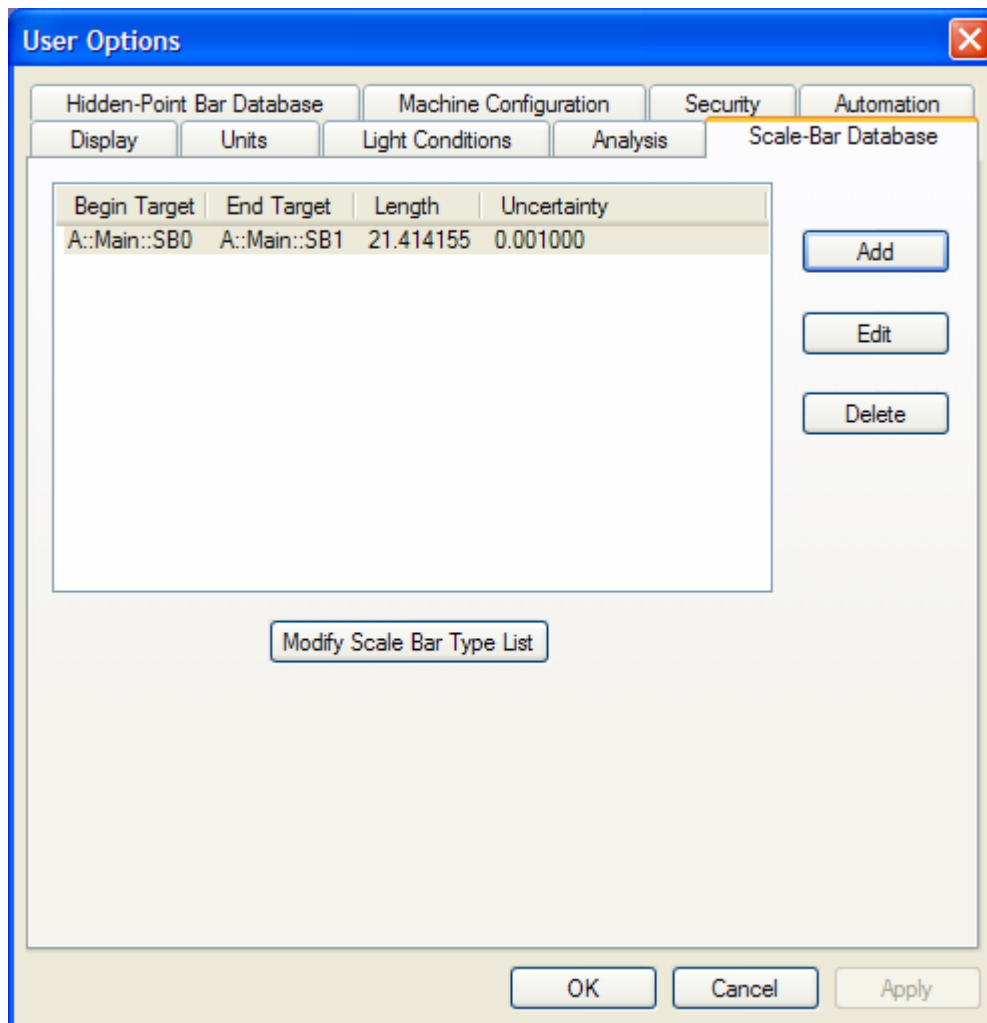
698.236933

[Add ScaleBar to SA](#)

A::Main::SB0 to A::Main::SB1

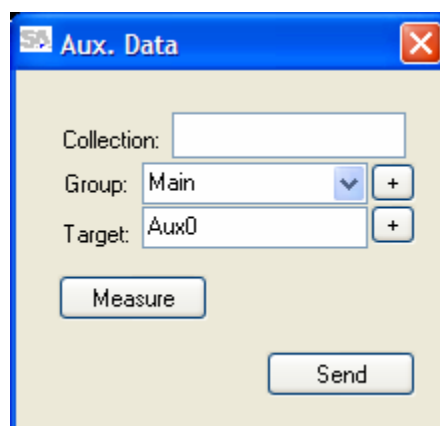
Length 543.919526

Just press 'Send' to send the scale bar to SA. (In this example, the interface is showing units of mm, and SA is showing inches.) In SA, under User Options, view the Scale-Bar Database to see the new definition.

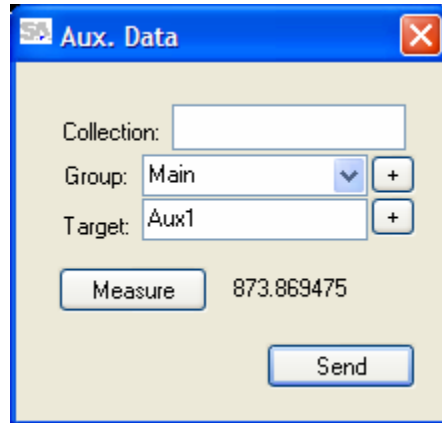


Continue building scale bars as desired.

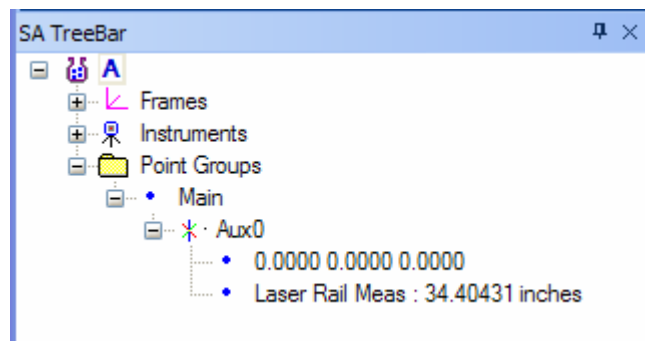
To add 1-D auxiliary data measurements to SA, close the Scale Bar dialog, and press 'Aux Data Meas' in the interface. The Aux. Data window will appear.



To record the laser rail distance, press Measure. The measurement will appear.



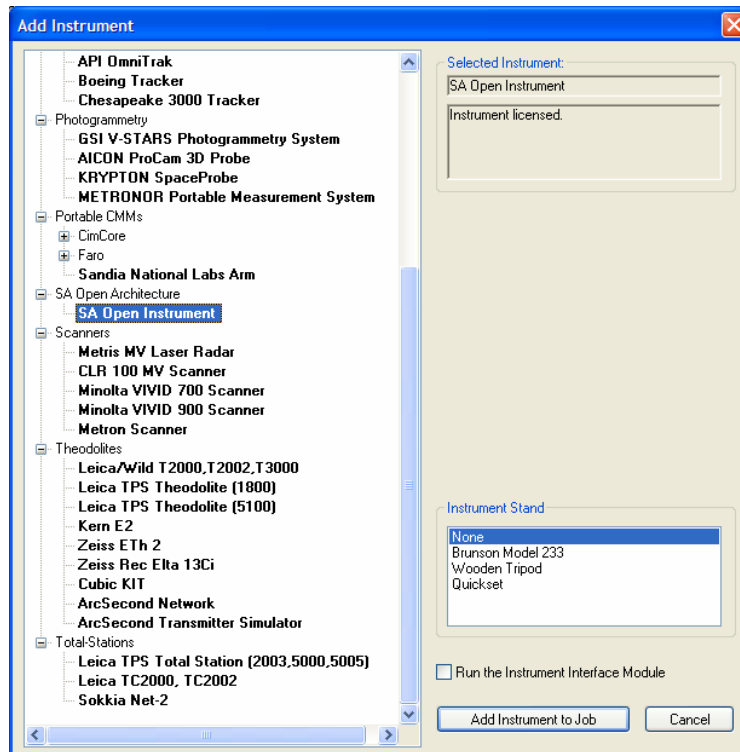
To send a measurement, simply adjust its name as desired, and press Send. The data will appear in the SA tree view. (Also in this example, the interface is displaying mm, and SA is showing inches.)



Continue measuring 1-D data points as desired.

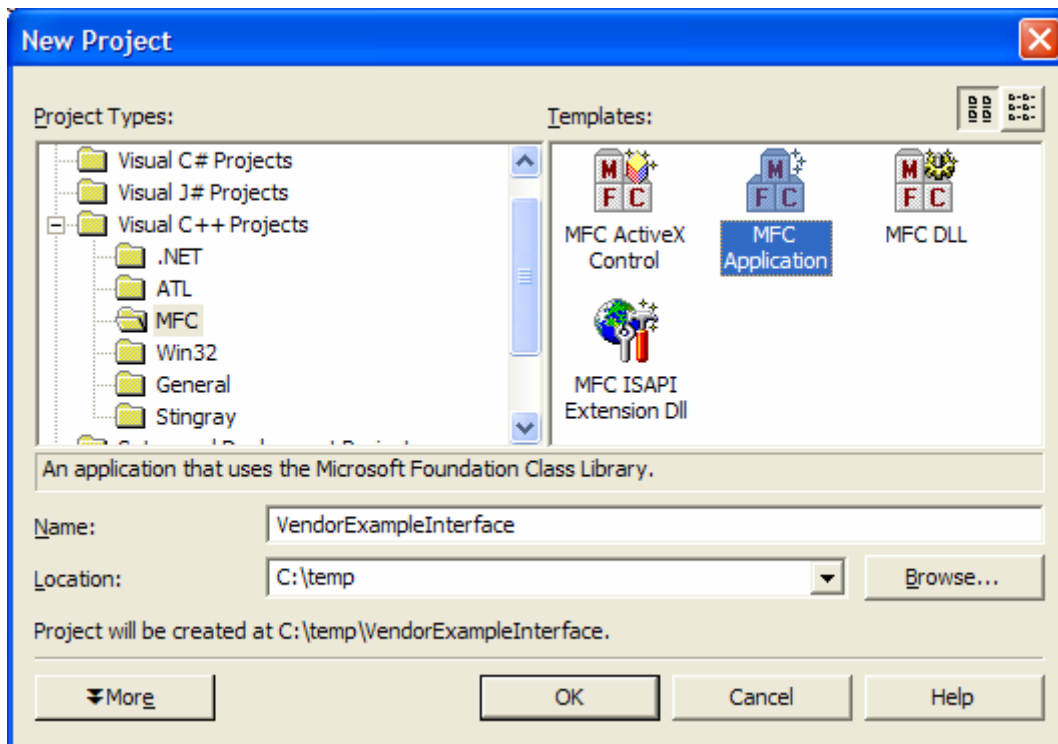
OPEN INSTRUMENT INTERFACE

The Spatial Analyzer (SA) Open Instrument interface is an ActiveX control that allows you to create your own instrument interface and send measurement information to SA. You must be running a recent version of SA that supports the Open Architecture / Instrument. When viewing the Add Instrument dialog within SA, the instrument is shown below:

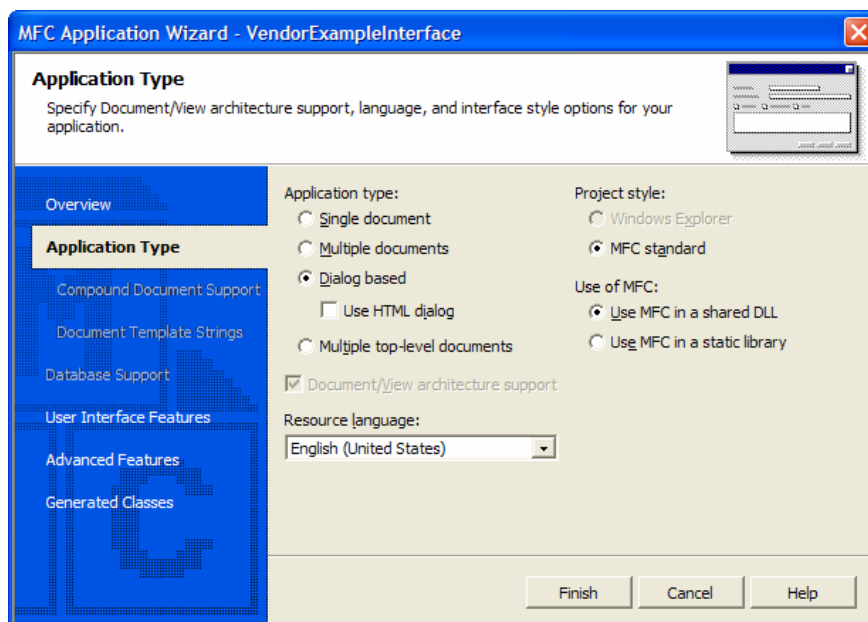


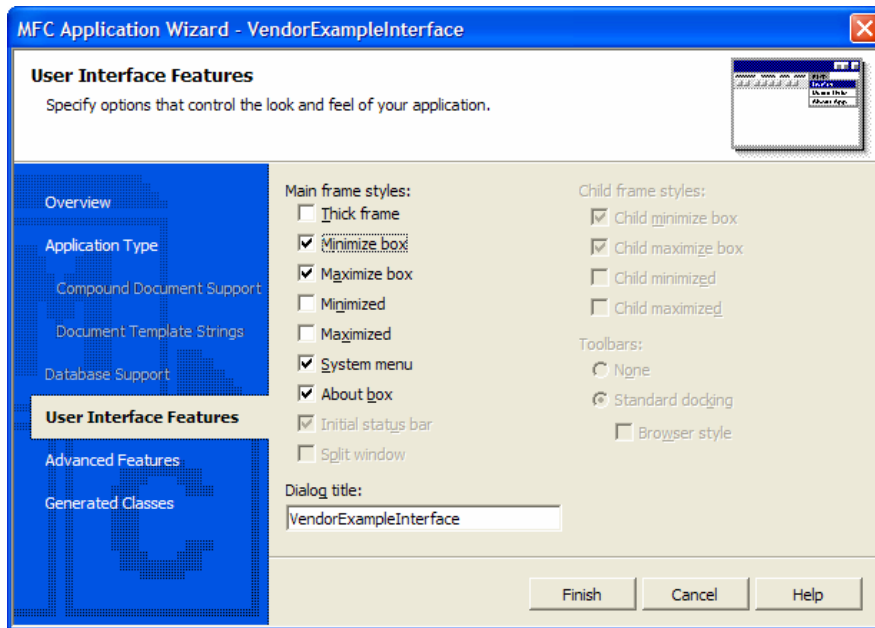
Creating the Interface (Visual C++)

Create a new dialog project

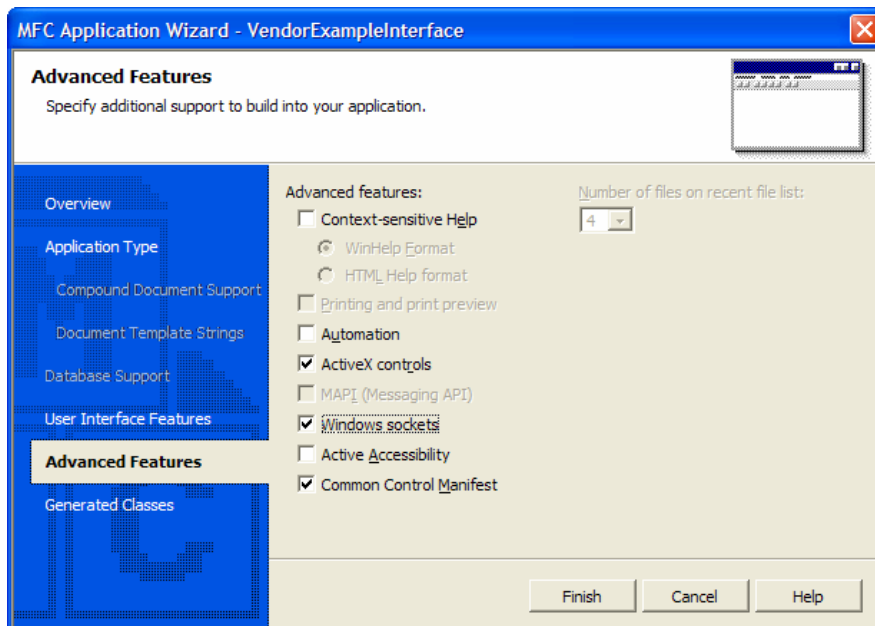


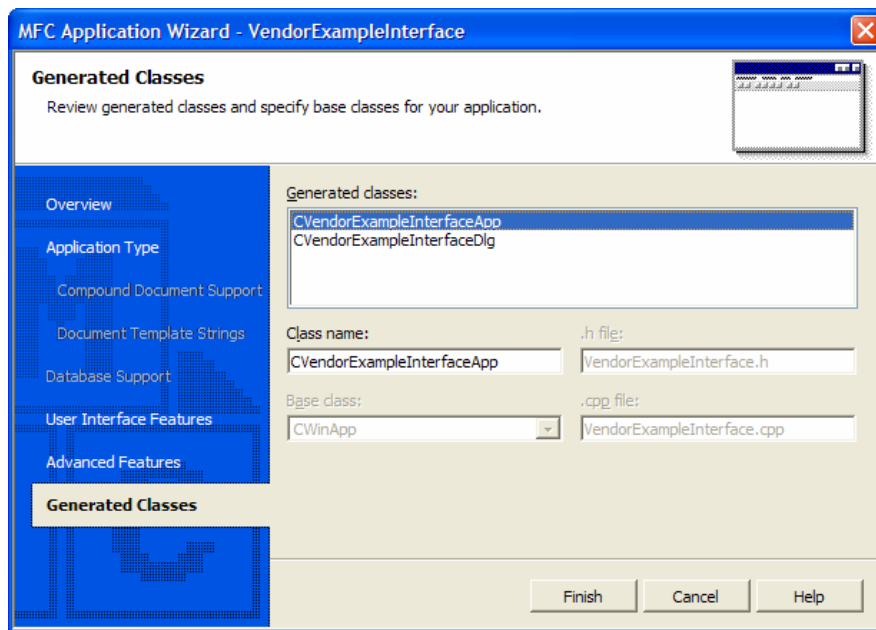
Setup the initial wizard configuration





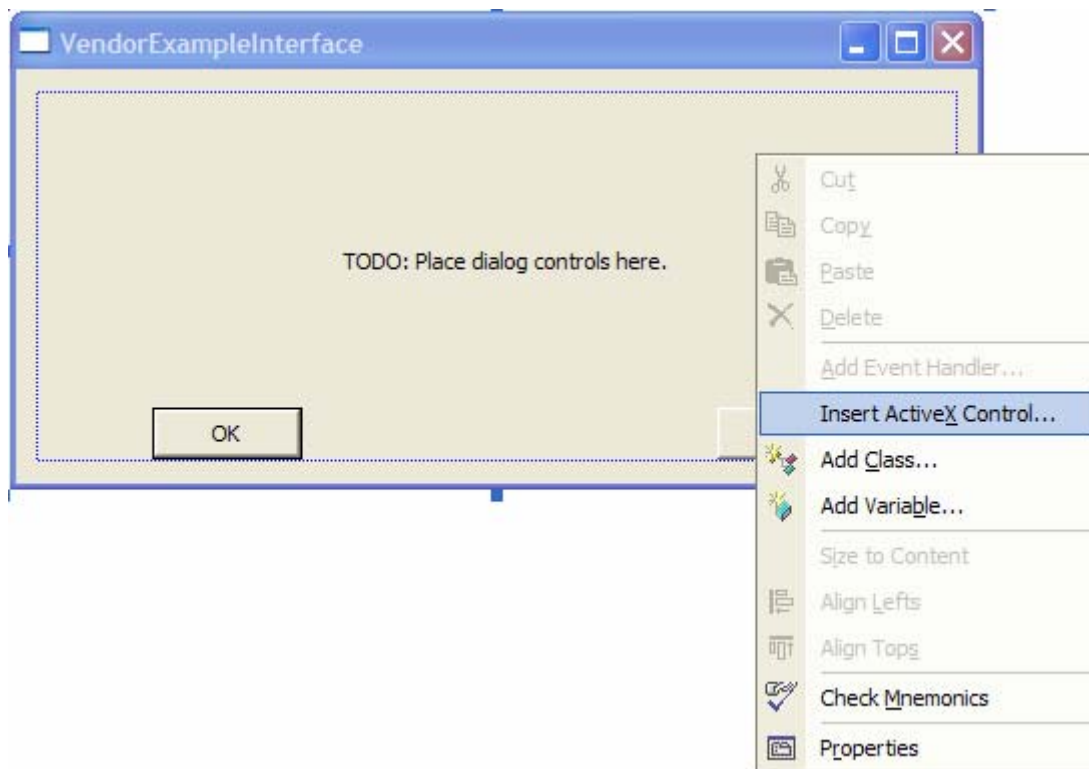
Be sure that “ActiveX controls” and “Windows sockets” are checked in the Advanced Features dialog:



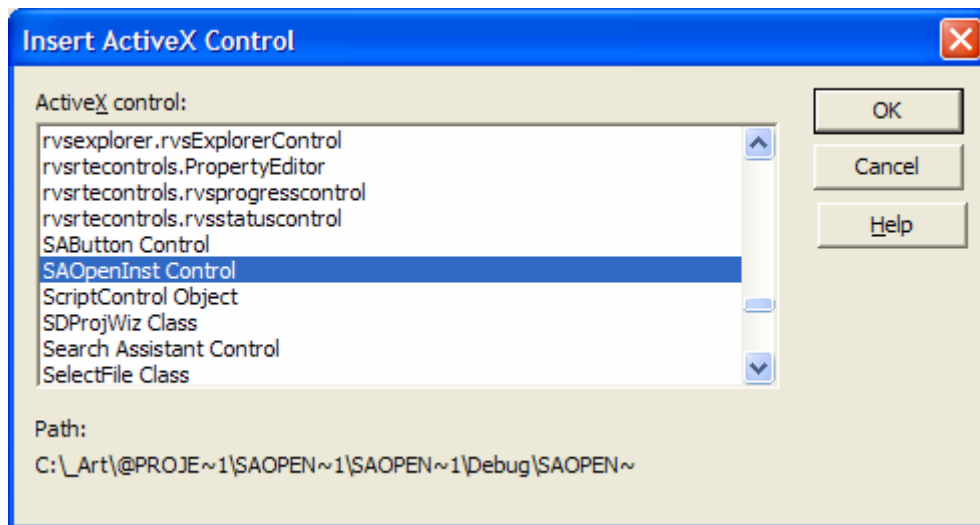


Click the “Finish” button when you’re ready to generate the project.

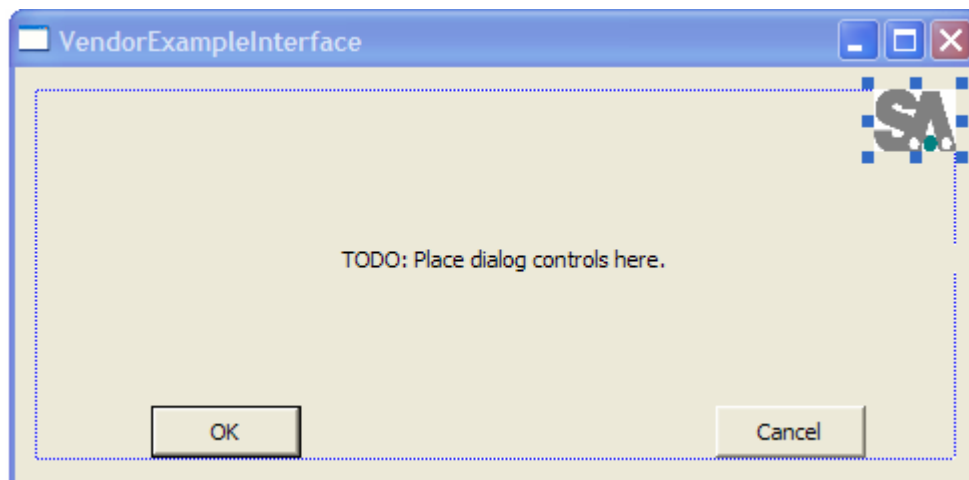
Insert the SA Open Instrument ActiveX control into your dialog:



Select the “SAOpenInst Control” from the Insert ActiveX Control shown below:

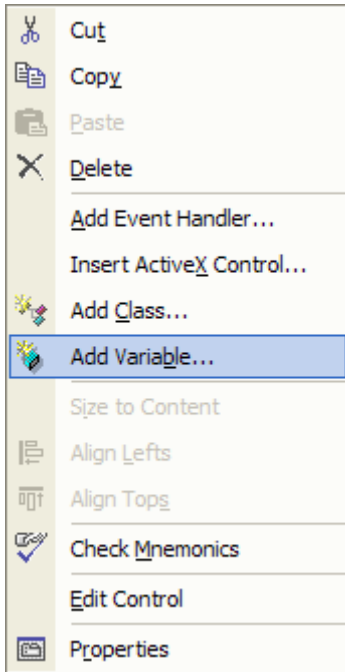


Move / Resize the control to the desired location within your dialog:

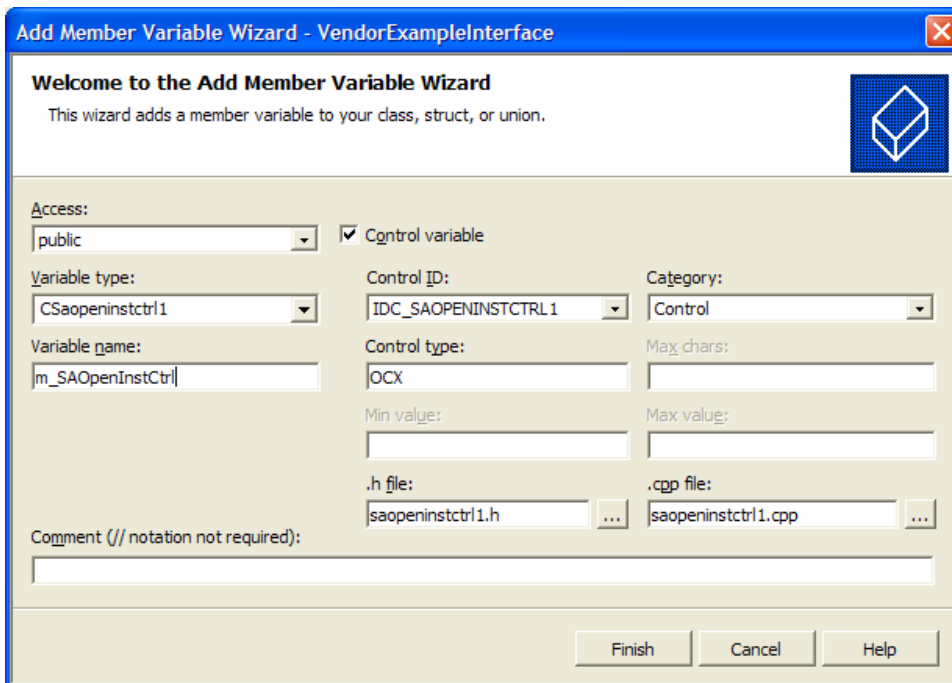


Now insert the control into your dialog .h and .cpp code:

Right Mouse click on the SAOpenInst Control and select the “Add Variable” menu item



Fill in the wizard fields as desired. In particular the Access and Variable name fields.



Click the “Finish” button.

Note: this will also create the saopeninstctrl1.h/cpp files necessary to communicate with the control.

With the control variable added to your dialog and the auto generated saopeninstctrl.h/cpp files, you're now ready to use the control's methods to implement your interface!

List of Open Instrument Interface Control Methods

The control methods are as follows:

BOOL PopLogonDialog(void):

Display the logon dialog to allow the user to select the desired SpatialAnalyzer connection.

@param none

@return TRUE for a successful logon, otherwise FALSE.

BOOL IsConnected(void):

Determine if the interface is connected to SpatialAnalyzer

@param none

@return TRUE if connected, otherwise FALSE.

void ClearMeasurements(void):

Clear the measurement buffer.

@param none

@return none

BOOL AddMeasurement(DOUBLE XValue, DOUBLE YValue, DOUBLE ZValue, DOUBLE XUncert, DOUBLE YUncert, DOUBLE ZUncert)

Add a measurement to the buffer to be sent to SA.

@param XValue the x-component of the measurement to be sent.

@param YValue the y-component of the measurement to be sent.

@param ZValue the z-component of the measurement to be sent.

@param xUncertVal the x-uncertainty of the measurement to be sent.

@param yUncertVal the y-uncertainty of the measurement to be sent.

@param zUncertVal the z-uncertainty of the measurement to be sent.

@return TRUE if successfully, otherwise FALSE.

BOOL SendMeasurements(LPCTSTR collectionName, LPCTSTR groupName, LPCTSTR targetName, DOUBLE planarOffset, DOUBLE radialOffset)

Send the pending measurement(s) to the current SA connection.

@param collectionName the name of the desired collection within SA for the measurement.

@param groupName the name of the desired group within SA for the measurement.

@param targetName the name of the desired target within SA for the measurement.

@param planarOffset the planar offset (inches).

@param radialOffset the radial offset (inches).

@return TRUE if the measurement was successfully sent, otherwise FALSE.

void ClearCloud(void):

Clear the cloud buffer.

@param none

@return none

BOOL AddCloudPoint(DOUBLE XValue, DOUBLE YValue, DOUBLE ZValue):

Add a cloud point to the buffer to be sent to SA.

@param XValue the x-component of the cloud point to be sent.

@param YValue the y-component of the cloud point to be sent.

@param ZValue the z-component of the cloud point to be sent.

@return TRUE if successfully, otherwise FALSE.

BOOL SendCloud(LPCTSTR cloudName, LPCTSTR collectionName):

Send the pending cloud data to the current SA connection.

@param cloudName the name of the desired cloud within SA.

@param collectionName the name of the desired collection within SA for the cloud.

@return TRUE if the cloud was successfully sent, otherwise FALSE.

void OnSetColGroupTarget(LPCTSTR collectionName, LPCTSTR groupName, LPCTSTR targetName):

Event handler that will be invoked to notify the user of a collection, group, target change request.

@param collectionName the desired collection name

@param groupName the desired group name

@param targetName the desired target name

@return none

void OnConnectToHardware(void):

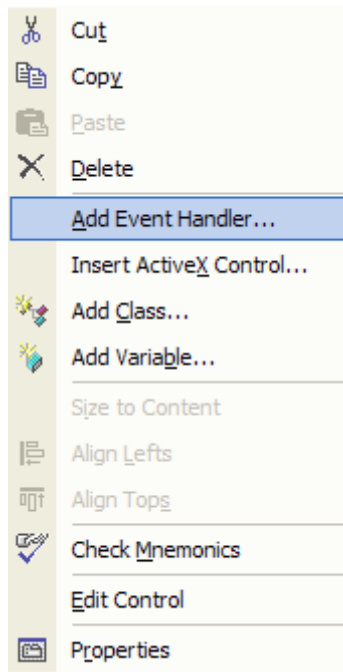
Event handler that will be invoked after a connection to SA has been established.

User may put hardware specific connection logic here.

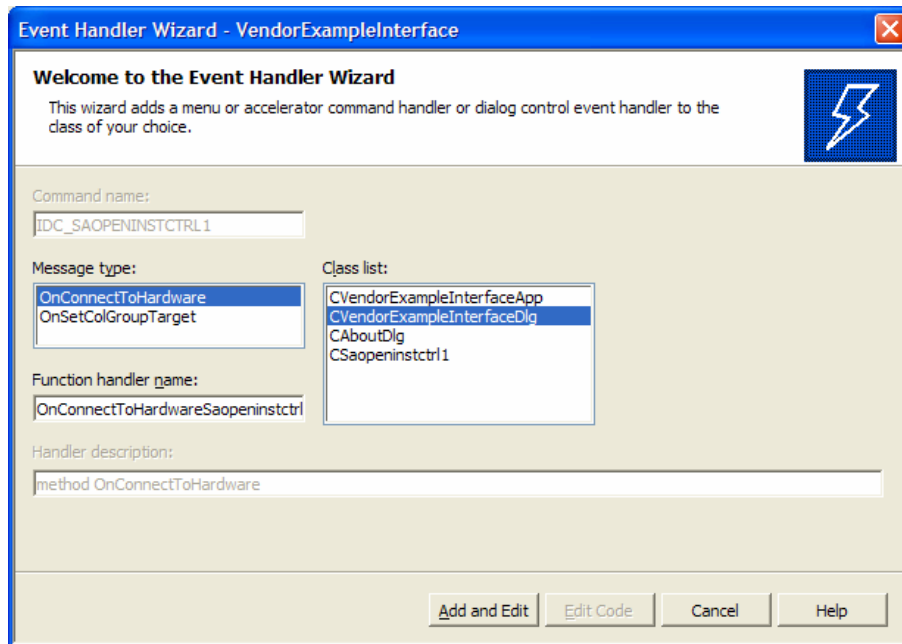
@param none

@return none

Event handlers are callbacks from the SAOpenInst Control into your application. To register with an event handler, right mouse click on the control and select the "Add Event Handler" menu item.



Select the desired event handler from the Event Handler Wizard dialog and Visual Studio will add the necessary code to your dialog:



Order of Operation

1. PopLogonDialog must be called first to setup a connection with Spatial Analyzer and the desired open instrument.
2. Always clear your measurements or cloud data after a successful send to SA.
Thus, the proper order of operations would be:

3. AddMeasurement for each measurement that you wish to send to SA.
4. SendMeasurements to send all buffered measurement data
5. ClearMeasurements to clear all buffered measurement data once successfully sent to SA.

Example Code

Be sure to check out the provided Visual Studio C++ example for a working example of the Open Instrument Interface. The dialog demonstrates sending measurement and cloud data to SA. The demo application looks like this:

The screenshot shows a Windows-style dialog box titled "SAOpenInstTest". At the top right is a "Logon" button. The dialog is divided into two main sections: "Measurements" and "Cloud".

The "Measurements" section contains the following controls:

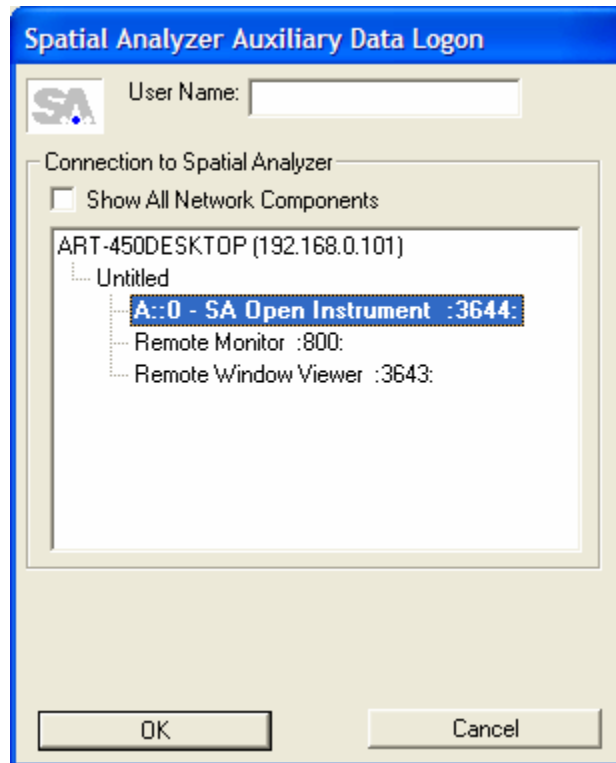
- Collection: [Empty text box]
- Group: [Main]
- Target: [P0]
- X-Pos: [0], X-Uncert: [0]
- Y-Pos: [0], Y-Uncert: [0]
- Z-Pos: [0], Z-Uncert: [0]
- Probe Radius: [0]
- Buttons: Clear, Add, Send

The "Cloud" section contains the following controls:

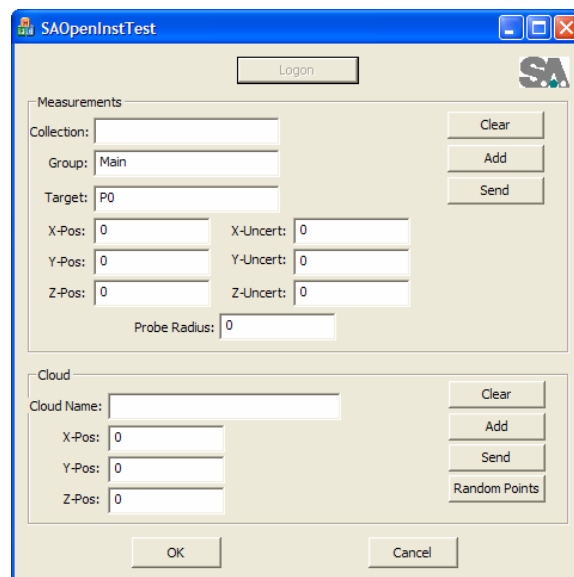
- Cloud Name: [Empty text box]
- X-Pos: [0], Y-Pos: [0], Z-Pos: [0]
- Buttons: Clear, Add, Send, Random Points

At the bottom of the dialog are "OK" and "Cancel" buttons.

With Spatial Analyzer running with a job containing an SA Open Instrument, press the Logon button. Select the desired instrument connection and press the "OK" button.



Upon successful connection to SA, the dialog will activate all the measurement / cloud dialog widgets:



Once connected to SA, enter point or cloud data. Add each item to the send buffer via the corresponding Add button. When ready to send to SA, press the corresponding Send button.

This example also supports the OnSetColGroupTarget event handler / callback. If you were to run a Measurement Plan from within SA that contains the “Set Instrument Group and Target” command, you should see the demo application’s collection / group / target fields reflect the measurement plan command data.

Version Information: Right mouse click on the SAOpenInst ActiveX Control and select the About menu item for product version information:



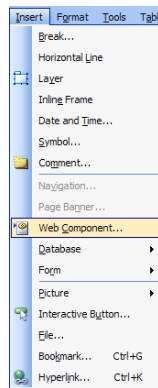
The Example Code directory also contains a Visual Basic example similar to the provided Visual C++ example.

For development assistance, please contact support@kinematics.com

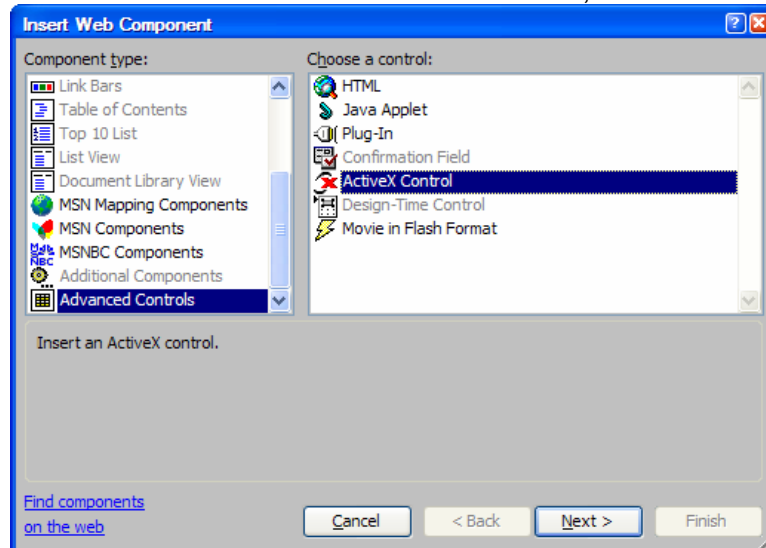
SABUTTON CONTROL INTERFACE

Directions for using the SAButton control within HTML via MicroSoft's FrontPage:

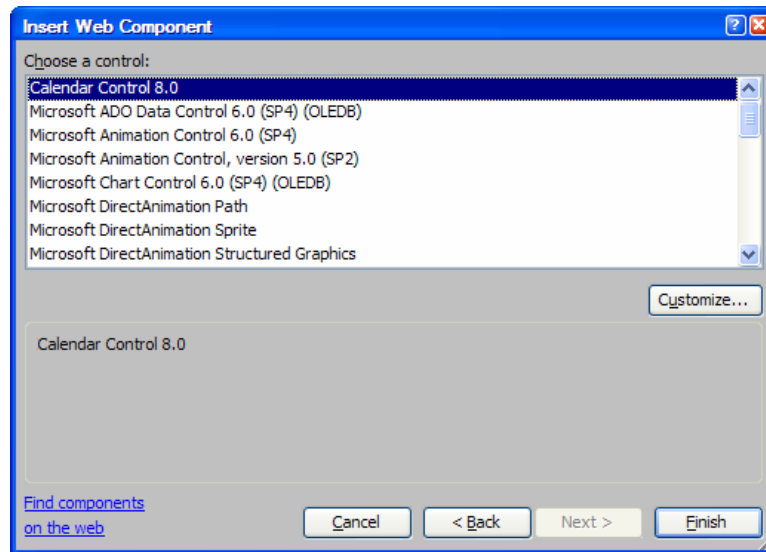
1. Insert the control into your HTML



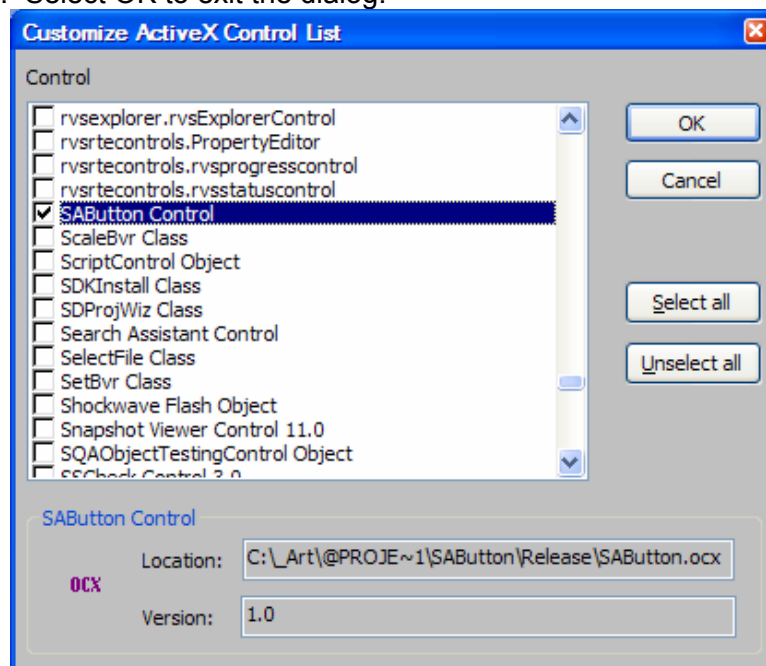
2. From the "Insert Web Component" dialog, select Component type: Advanced Controls with Choose Control: ActiveX Control. Then, select the Next> button:



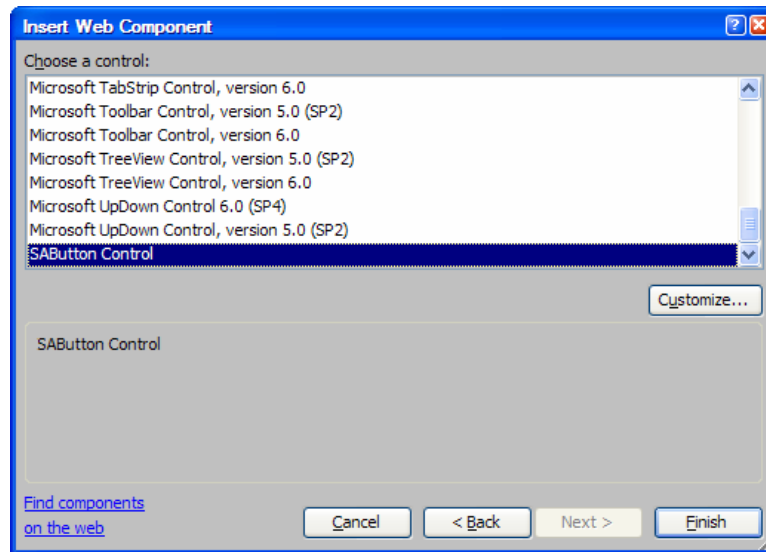
3. Now Select the Customize... button:



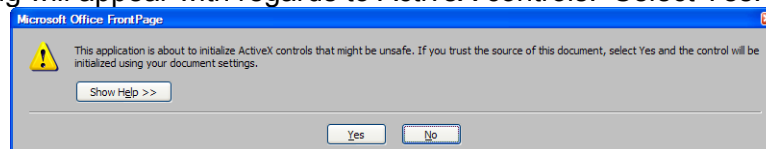
4. Scroll down the Control list and make sure that the “SAButton Control” is checked. Select OK to exit the dialog.



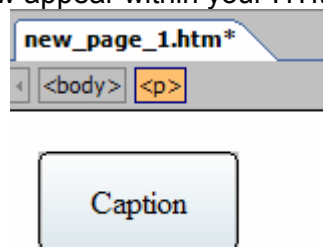
5. Now scroll down and select the SAButton Control from the list and select Finish:



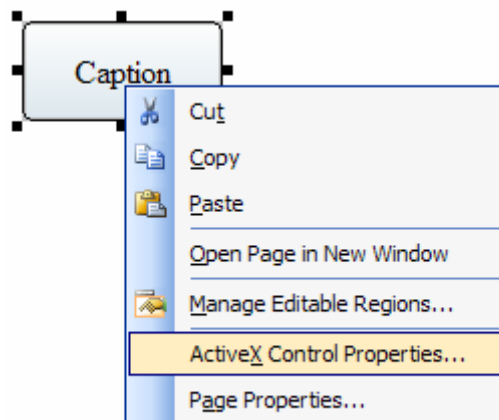
6. A warning will appear with regards to ActiveX controls. Select Yes:



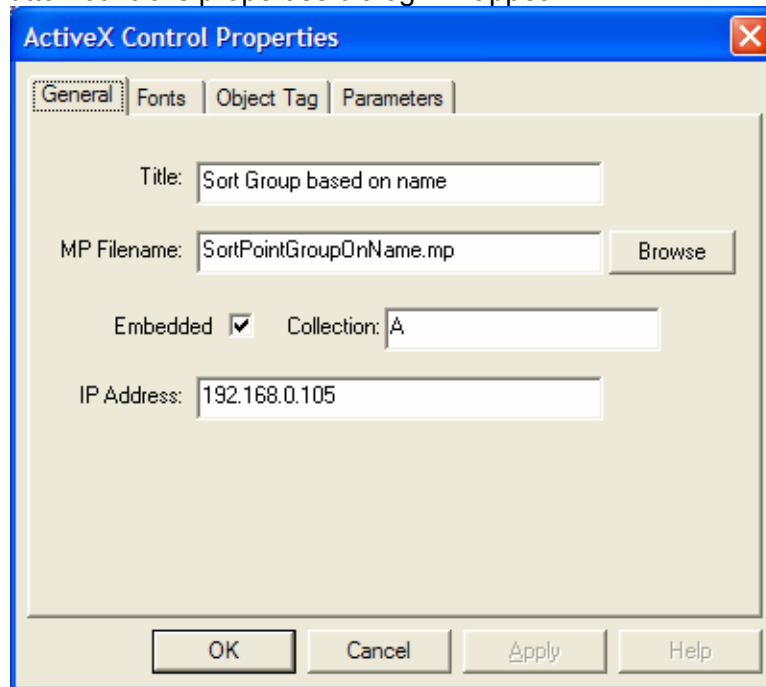
7. The button control will now appear within your HTML document:



8. Right mouse click on the button control and select "ActiveX Control Properties..."



9. The SAButton control's properties dialog will appear:



10. Fill in the desired options and select the OK button.
11. Repeat the above process for as many SAButton controls as desired.
12. When complete, save the html file.
13. The contents should look similar to the following:

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>HTML Example for using SpatialAn</title>
</head>
<body>
<p>HTML Example for using SpatialAnalyzer's SAButton control:</p>
<p>
<object classid="clsid:5B220038-2A1E-4B5B-8505-C559CC0E1D35" id="SAButton1"
width="118" height="50">
    <param name="title" value="Sort Group based on name">
    <param name="MPFilename" value="SortPointGroupOnName.mp">
    <param name="Embedded" value="1">
    <param name="Collection" value="A">
    <param name="IPAddress" value="192.168.0.105">
</object>
</p>
</body>
</html>
```