Reference Manual for the ATS Version 2.0

David A. Rosenfeld
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

I. Introduction

This document, the *Reference Manual for the Algorithm Testing System Version 2.0*, provides detailed descriptions of all the major features of the Algorithm Testing System (ATS). The ATS is one of the tools used in NIST's Algorithm Testing and Evaluation Program for Coordinate Measuring Systems (ATEP-CMS)—a NIST Special Test Service for evaluating the performance of CMS data analysis software. (ATEP-CMS was recently approved as a NIST Special Test Service, numbers 1007OS and 1008OS.)

I.A Purpose of the Reference Manual

This Reference Manual is intended as a companion volume to the *User's Guide for the Algorithm Testing System Version 2.0* (see complete reference below). The purpose of the User's Guide is to get the user started with the ATS and to acquaint him with the ATS's basic operation. This Reference Manual contains more comprehensive, detailed descriptions of the major features of the ATS. It is possible to make use of many of the capabilities of the ATS without being concerned with many of the details provided here. Thus, the Reference Manual is not intended for the first time user, but is primarily useful for occasional look-up. It should also be noted that much of the information provided by the Reference Manual may be found in the ATS on-line help, often in a more summarized form.

I.B Other Relevant Publications

The following is a list of relevant publications and sources of information:

1. ATS on-line help.

2. Rosenfeld, David A., 1995, NISTIR 5674, *User's Guide for the ATS Version 2.0*, National Institute of Standards and Technology, Gaithersburg, MD.

3. Diaz, Cathleen, 1994, NISTIR 5366, *Concept for an Algorithm Testing and Evaluation Program at NIST*, National Institute of Standards and Technology, Gaithersburg, MD.

4. ASME, 1995, ASME B89.4.10-199x, *Methods for Performance Evaluation of Coordinate Measuring System Software*, Draft, American Society of Mechanical Engineers, New York.

5. Hopp, Theodore, and Levenson, Mark, *Peformance Measures for Geometric Fitting in the NIST Algorithm Testing and Evaluation Program for Coordinate Measuring Systems*, submitted to the NIST Journal of Research, National Institute of Standards and Technology, Gaithersburg, MD.

The ATS was developed to support the Algorithm Testing and Evaluation Program for Coordinate Measuring Systems (ATEP-CMS), a special test service provided by NIST. The ATS was developed in coordination with, and in support of the ASME B89.4.10 draft standard, *Methods for Performance Evaluation of Coordinate Measuring System Software*.

I.C Contacts for Further Questions and Comments

Further questions and comments, including the reporting of bugs, should be directed to any of the members of the ATEP-CMS project team at NIST. They are:

Cathleen Diaz
phone: (301)975-2889
email: cathyd@cme.nist.gov

Ted Hopp
phone: (301)975-3545
email: hopp@cme.nist.gov

David Rosenfeld
phone: (301)975-5518
email: dar@cme.nist.gov

The mailing address is:

National Institute of Standards and Technology
Bldg 220, Rm A127
Gaithersburg, MD 20899
fax: (301)258-9749

II. ATS Run Modes

The ATS may be operated with a number of different options. These options may be specified at the command line when you run the ATS. The command line syntax is,

ats [/t[<lines>]] [/e] [/x[<start>[,<len>]]] [/m] [/?|?]

The following table describes each option.

option	description	default
/? or ?	print message listing options	off
/t	run in text mode	graphics if available
<lines>	controls the text display mode. Can be 25, 43, or 50	25
/e	use expanded memory	off
/x	use extended memory	off
<start>, <len>	the starting address (in KB) of available extended memory[1]; how much extended memory to use[2]	all extended memory if /x is specified
/m	run in monochrome mode	color if available

[1] If a starting address is not given or equals 0, the ATS searches for the start of extended memory.

[2] If no length is supplied, all of extended memory will be used.

All options can also be specified with an initial '-' instead of '/'.

III. Key Semantics, Hotkeys

The following table lists and describes the ATS "hotkeys". It also explains the semantics of common ATS editor keys.

Key	Description	When Functional
<F1>	Invoke ATS on-line help	Always (unless status button displaying "Wait")
<F2>	Save comments	Comment editor
<F10>	Activate menu bar	Panels, Viewer (not from popup windows)
Alt+PgDn	Descend to viewer	Panels
Alt+PgUp	Ascend to panels	Viewer
mouse-click, ENTER	Select item	Panels, when cursor positioned on item
mouse double-click, keypad '+', INS	Insert ATS object	Panels
keypad '-', DEL	Delete ATS object	Panels
mouse double-click, keypad '+' [1]	Insert form error list item or form error schema item	Form error schema window
keypad '-' [1]	Delete form error list item or schema item	Form error schema window
ESC	Cancel	Okay/Cancel windows
	Okay	Okay window
	Terminate current operation (import/export/generation) and entire remainder of multiple process	When status window with "Press ESC to cancel" is displayed
	Terminate current operation (import/export) but not remainder of process	When status window without "Press ESC to cancel" is displayed
	Move to higher menu level or exit menu, if at highest level	Menu

Key	Description	When Functional
ESC (con't)	Quit, discard inputs	Comment Editor
	Exit	On-line help
mouse-click outside popup window	Ignored	Okay/Cancel windows
	Okay	Okay windows
Alt+O	Okay	Okay, Okay/Cancel windows
Alt+C	Cancel	Okay/Cancel windows
Alt+D[2]	Done - return to *"Error Schema"* list	editing area of Form Error Schema Window
	Done - exit window	upper sections of Form Error Schema Window
Ctrl+L	Repaint Display	Panels, Viewer (not from popup windows)
Ctrl+P	Print current object (calls print window)	Panels, Viewer (not from popup windows)
Ctrl+C	Copy current test	Test panel, Viewer when cursor last positioned in test panel
"*" (keyboard and keypad)	Tag object	Panels, when cursor placed over test, data set, or fit
Backspace	Previous help screen	On-line help

[1] In the Form Error Schema window, INS and DEL cannot be used since they have significance as editing keys (INS - typeover mode; DEL - delete character).

[2] From the panels and viewer, however, Alt+D will invoke the *"Disk"* section of the menu bar, if selectable. This capability is discussed in the following paragraph.

The ATS also supports single-key menu selections. To select an unprotected section of the menu from the panels or viewer, press "Alt" + the highlighted key of the menu section. Once you are in the menu (at any level), you may press the highlighted letter of a menu selection (without the "Alt" key) to select it.

IV. Highlighted versus Loaded Objects

While operating the ATS, it is important to bear in mind on what objects ATS operations are performed. Some operations are performed on the loaded object, while others are performed on the object highlighted by the cursor. We first define the difference between highlighting and loading. We then provide a table to clarify this issue.

Let us first review some of the basic object-selection operations in the ATS. Placing the cursor on an object (using the arrow keys only) highlights it. Highlighted objects are displayed in white on light purple (on monitors which support color). (It should be noted that the highlighted "object" may not be an object at all. The user may place the cursor on the blank line of a panel (which appears after the list of objects in the panel). This position too can be the subject of certain commands, such as insert (i.e. insert a new object at the highlighted position).) When you move the cursor to a new panel, the menu bar shifts to reflect the commands available for the objects of that panel.

Mouse-clicking an object or placing the cursor on an object and pressing ENTER selects an object. Selecting an object loads the contents of the object from the ATS database to local memory. It also loads all previously loaded descendants of the object to local memory. (A maximum of one object per level of the ATS hierarchy may be loaded.) Finally, selecting an object fills the viewer with information about the object. Loaded objects are displayed in black on gray (unless they are also highlighted). Once an object is selected, it will remain selected until you select another object, (even if you move the cursor elsewhere).

Thus, at a given time during the operation of the ATS, there may be three different "active" objects or positions:

- the selected object
- the highlighted position (may or may not be an object)
- the loaded object in the panel which contains the highlighted position

Which of these three objects will be the subject of an ATS command depends on the following issues. The viewer reflects information on the selected object (as above). Thus, all editing of the information in the viewer affects the selected object. We have noted above that the menu bar reflects the commands available for the objects in the panel containing the highlighted object. This is because ATS commands follow the panel in which the cursor is positioned. Which object in the cursor-positioned panel ATS commands operate on depends on the command. Some commands operate on the highlighted object, while some operate on the loaded object. There is no general rule regarding which commands operate on which type of object. Often, however, commands which relate to the positioning of objects (such as at which location in the panel are imported data sets placed) use the cursor-positioned object as the reference.

The table below defines which command is carried out on which object.

Operation	Performed On Which Object
select	highlighted object
insert	new object inserted at the highlighted location
delete	highlighted object
generate	loaded object
fit	loaded object
analyze	loaded object
import (this level)[1]	imported objects placed at the highlighted location
import (succeeding level)	imported objects become children of loaded object[2]
export	loaded object, or descendants of loaded object (if exporting descendants of an object)
print	highlighted object
tag	highlighted object

[1] E.g. if the cursor is positioned at the dataset level and you request to import data sets.

[2] The new objects are appended to the end of the existing object list.

V. Geometry Local Coordinate Systems

The ATS defines a local right-handed coordinate system for every geometry generated by the ATS. Its purpose is primarily internal, to allow the ATS to mathematically represent the geometry. The local coordinate systems do, however, have some relevance to the user. The form error descriptions of section VI in particular are described in terms of the geometries' local coordinate systems. The local coordinate systems are referred to as *qrs* systems.

The ATS also defines u and v directions within every geometry. These measure the horizontal and vertical extents of the geometry, respectively. (The v or vertical direction applies to all geometries except lines and circles.) These directions are of interest to the user primarily to understand the nominal geometry definitions and form error descriptions within the ATS, as discussed in sections V and VI.

The bulk of this section presents tables defining the qrs coordinate directions, and the u and v directions for each geometry. Beforehand, however, we present a general discussion of the method through which the ATS derives local directions from the global xyz coordinate systems for

geometries with an orientation. We refer to this local coordinate system as the x'y'z' system. Afterwards, a table will relate this system to the qrs for each geometry supported by the ATS.

All ATS geometries except the sphere have an orientation. For a line, orientation means the direction of the line. For a circle, plane and torus, it means the direction of the normal to the plane of the geometry. For a cylinder and cone, it means the axis direction. The ATS defines this direction (viz-a-viz the global xyz axes) by means of an azimuth and declination. The azimuth is the planar angle from the global +x axis in the xy plane; the declination is the polar angle from the +z.

Referring to Figures 1 and 2 at right, a set of local axes may be derived from the global axes in the following steps:

1. Rotate the xz plane az° toward the +y axis, as shown in Figure 1. This rotates both the x and the y axes. The resulting axes are named x" and y'.
2. Rotate z and x" decl° along the x"z plane, as shown in Figure 2. This rotates the z and x" axes, resulting in z' and x". y' remains unchanged.
3. The resulting x'y'z' axes form a right-handed local coordinate system.

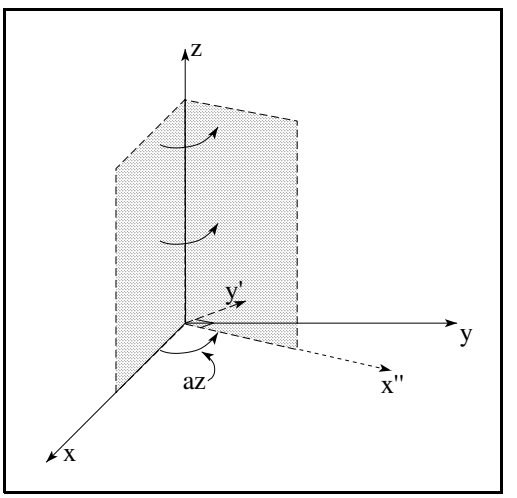

Figure 1 - Rotate xz plane azimuth°

This is the basic methodology the ATS uses to derive a local coordinate system from an azimuth and declination. (The azimuth and declination themselves are entered by the user as part of the nominal geometry description. Nominal geometry descriptions are discussed in section V.) Generally, the local x'y'z' system forms the geometry's local qrs coordinate system. In some geometries, however, there are slight discrepancies between the two systems. The correlation between the two systems is summed up in the table on the next page.

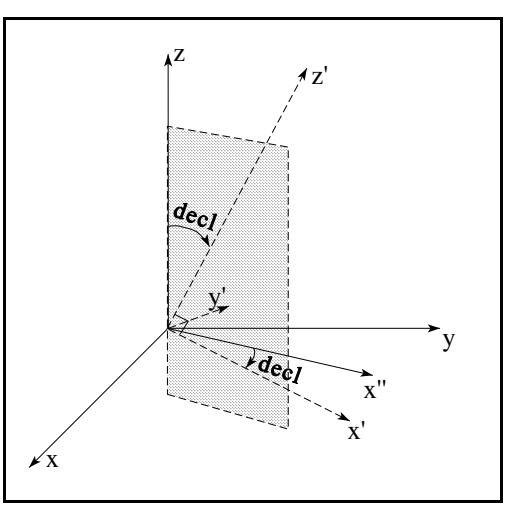

Figure 2- Rotate z and x" decl°

Local Coordinate Axis	Parallel to						
	line	circle	plane	sphere	cylinder	cone	torus
q	z'	x'	x'	x	x'	x'	x'
r	-y'	y'	y'	y	y'	y'	y'
s	x'	z'	z'	z	z'	z'	z'

The line differs in that the local z' (the direction defined by the azimuth and declination) which is used as the line direction, is considered to be the line q axis. The sphere ignores the local coordinate system, because the ATS considers a sphere to have no direction. Thus, it uses the global xyz coordinate system.

The following tables defines how each geometry is measured within its coordinate system. As above, geometric extents are measured according to u and v, or horizontal and vertical directions, measuring from the origin of the local coordinate system.

Geometry	Center	U Direction	V Direction
line	point on line supplied by user as line location	position on line along the +q axis from origin	N/A
circle	center of circle (supplied by user as circle location)	position on circumference of circle (0°-360°) from circle's intercept with +q	N/A
plane	point on plane supplied by user as plane location	position on plane from origin in direction of +q axis	position on plane from origin in direction of +r axis
sphere	center of sphere (supplied by user as sphere location)	azimuthal position (line of longitude) on sphere from local +q (0°-360°)	elevation position (line of latitude) on sphere, from south pole (-90°, intercept of -s with sphere), to north pole (+90°, intercept of +s with sphere)
cylinder/ cone	point on axis (the local s axis) supplied by user as cylinder/ cone location	position on circumference of geometry (0°-360°) from its intercept with +q	position along geometry axis (the s axis) from local origin
torus	center of torus (supplied by user as torus location)	position on major circle of torus (0°-360°) from its intercept with +q	position on minor circle of torus (0°-360°) from intercept of outer width of torus with qr plane

VI. Nominal Geometry Specifications

When defining a test, the user must define a schema of nominal geometry definitions. These definitions are used by the ATS to generate data sets for the test. Each ATS geometry is defined with a different set of parameters. The following tables describe the set of parameters used for each geometry. In general, all parameters may have negative values. All parameters which measure geometric angles are entered in decimal degrees.

Line

Attribute	Parameters	Description
location	x, y, z	global coordinate location of a point on the line.
direction	azimuth declination	direction of the line. az - angle from global +x in xy plane. decl - polar angle from +z.
extent	start, length	extent of the line relative to the point identified by "location". start - the start of the line relative to location. length - the length the line extends from start.

Circle

Attribute	Parameters	Description
location	x, y, z	global coordinate location of the center of the circle.
direction	azimuth declination	direction of the normal to the plane of the circle. az - angle from global +x in xy plane. decl - polar angle from +z.
radius	r	radius of the circle
extent	arc start arc length	extent of the circle/arc in degrees. arc start - start of the circle relative to the local +q axis. arc length - the extent of the arc from arc start.

Plane

Attribute	Parameters	Description
location	x, y, z	global coordinate location of a point on the plane.
direction	azimuth declination	direction of the normal to the plane. az - angle from global +x in xy plane. decl - polar angle from +z.
extent	u start u length v start v length	extent of the plane relative to the point identified by location. Extent measured in both plane directions, its local u and local v. In each direction, a "start" value measures the start of the plane direction relative to "location", and a "length" value measures the length the plane direction extends from start.

Sphere

Attribute	Parameters	Description
location	x, y, z	global coordinate location of the center of the sphere.
radius	r	radius of the sphere.
extent	arc start arc length pole angle start pole angle end	extent of the sphere in both its u and v directions. arc start and length measure the sphere extent around the equator, from the +q (0°-360°). pole angle start and end measure its extent in the longitudinal direction, from the south pole, -s (-90°) to the north pole, +s (+90°).

Cylinder

Attribute	Parameters	Description
location	x, y, z	global coordinate location of a point on the axis.
direction	azimuth declination	direction of the cylinder axis. az - angle from global +x in xy plane. decl - polar angle from +z.
radius	r	radius of the cylinder.
extent	arc start arc length axis start axis length	extent of the cylinder in both its u and v directions. arc start and length measure the cylinder circumference from the +q (0°-360°). axis start and length measure the axis extent, relative to "location".

Cone

Attribute	Parameters	Description
location	x, y, z	global coordinate location of a point on the axis.
direction	azimuth declination	direction of the cone axis. az - angle from global +x in xy plane. decl - polar angle from +z.
distance to surface	d	perpendicular distance from "location" to the cone surface.
half angle	a	cone half angle, or the angle between the cone axis and surface.
extent	arc start arc length axis start axis length	extent of the cone in both its u and v directions. arc start and length measure the cone circumference from the +q (0°-360°). axis start and length measure the axis extent, relative to "location".

Torus

Attribute	Parameters	Description
location	x, y, z	global coordinate location of the center of the torus.
direction	azimuth declination	direction of the normal to the plane of the torus. az - angle from global +x in xy plane. decl - polar angle from +z.
major radius	r	major radius of torus.
minor radius	R	minor radius of torus.
extent	major arc start major arc length minor arc start minor arc length	extent of major and minor arcs of torus. major arc start and length measure the major arc extent relative to the local +q (0°-360°). minor arc start and length measure the minor arc extent relative to the intersection of the plane of the torus (its local qr plane) with the outer edge of the torus (0°-360°).

VII. Form Errors

This section provides tables and descriptions of all the form errors implemented in the ATS. It lists and describes the form errors which are implemented for each geometry, and explains the parameters used in defining each error. However, the mathematics used in form error calculation are not discussed. A document in preparation will be devoted to this issue. This section assumes you are familiar with the local coordinate systems applied to ATS geometries, as discussed in Section IV.

The parameter abbreviations used in this section are:

amp - amplitude
ori - orientation
loc - location
freq - frequency
dir - direction

All parameters may be set to negative values.

Line Form Errors

Form Error	Parameters	Explanation
Oriented Random	amp ori	Plane of error will be ori° from +s axis of local qs plane*. Each point on line will be moved along this plane a random magnitude in range [-amp,amp].
Oriented Sine	amp freq phase ori	Plane of error will be ori° from +s axis of local qs plane*. Line will assume a sinusoidal curve along this plane with given amp, freq, and phase. Freq - frequency of curve along entire line. Phase - phase of curve at start of line (not at line's local origin).
Oriented Step	amp loc ori	Line will step amp units. Start of step will occur at loc, measuring from line's local origin. Direction of step will be ori° from +s axis*.
Oriented Bend	angle loc ori	Line will bend angle°. Start of bend will occur at loc, measuring from line's local origin. Direction of bend will be ori° from +s axis*.

* Angle measurements are made in a counterclockwise direction when viewing from the +q axis.

All line form errors have an orientation. Form errors of other geometries may be propagated in the direction of the normal to the surface of the geometry. A line, however, has no unique normal. Thus, to fully define a form error, an orientation is required.

Circle Form Errors

All circle form errors are propagated within the plane of the circle.

Form Error	Parameters	Explanation
Random	amp	Each point on circle moved in direction of normal to circle at that point. Movement is a random magnitude in range [-amp, amp].
Sine	amp freq phase	Circle will assume a sinusoidal curve with given amp, freq, and phase. Freq - frequency of curve along entire circle or arc. Phase - phase of curve at start of circle/arc.
Closed Step	amp step start step length	Circle will step "amp" units for some or all of its circumference. Start of step will be "step start" degrees from local +q axis. Step will extend "step length" degrees.

Sphere & Torus Form Errors

Form Error	Parameters	Explanation
Random	amp	Each point on geometry moved in direction of normal to geometry a random magnitude in range [-amp,amp].
Surface Sine	amp (2) freq (2) phase (2)	Generates a sinusoidal wave on geometry surface by defining sine errors in both the geometry's local u and v directions. Each sine error is defined according to an amp, freq, and phase. Freq - frequency of curve along entire geometry direction[1]. Phase - phase of curve at start of geometry direction.

[1] Recall from section IV that for a sphere the u direction extends 360° from the sphere's +q, whereas the v direction extends 180° - from the south pole (local -s) to the north pole (+s). For a torus, however, the v direction (measuring the minor radius) also extends 360° - measuring from the intersection of the plane of the torus with the outer edge of the torus's minor radius.

Plane Form Errors

Form Error	Parameters	Explanation
Random	amp	Each point on plane moved in direction of normal to plane a random magnitude in range [-amp,amp].
Surface Sine	amp (2) freq (2) phase (2)	Generates a sinusoidal wave on planar surface by defining sine errors in both the plane's local u and v directions. Each sine error is defined according to an amp, freq and phase. Freq - frequency of curve along entire plane direction. Phase - phase of curve at start of plane direction.
Surface Step	amp loc dir	Generates a step in surface of plane of "amp" magnitude. loc and dir specify a line on the plane, to right of which is step. dir - direction of step line. Measured in degrees counterclockwise (when viewed from the +s axis) from the +q axis in the local qr plane. loc - perpendicular distance from plane's local origin to line of step. If positive, origin will lie to left of line; if negative, origin will lie to right of line.
Surface Bend	angle loc dir	Generates bend in surface of plane of angle°. loc and dir specify a line on the plane (see step error above) to right of which begins bend.

As above, the "error line" of the step and bend errors, or the line beyond which the step or bend begins, is determined by parameters loc and dir. It should be noted that dir, the direction of the error line, affects the interpretation of loc, its location. This is because loc is the perpendicular distance from the origin to the error line. Thus, the error line direction determines the perpendicular direction we measure from the origin.

It is quite possible that an entire step or bend error will lie outside the bounds of the plane. (This is the case for all step and bend errors which have locations determined by loc.) In such a case, the error will not affect the generated data. Note in particular that the default values for plane extent (u/v start=0; u/v extent=1) place the local origin at the lower left corner of the plane boundary, and the default values for the line of error (loc=0; dir=0) place it on the +q axis, thus making the error lie entirely below the geometry extent.

Cylinder & Cone Form Errors

Form Error	Parameters	Explanation
Random	amp	Each point on geometry moved in direction of normal to geometry a random magnitude in range [-amp,amp].
Axis Sine	amp freq phase ori	Plane of error will be ori° from geometry's local qs plane. Axis will assume sinusoidal wave with given amp, freq and phase. Freq - frequency of curve along entire axis. Phase - phase of curve at start of axis (not at local origin).
Axis Bend	angle loc ori	Axis will bend angle°. Start of bend will occur at loc, measuring along axis from geometry's local origin. Direction of bend will be ori° from +q axis (measuring counterclockwise when viewing from +s).
Axis Step	amp loc ori	Axis will step amp units. Start of step will occur at loc, measuring along axis from geometry's local origin. Direction of step will be ori° from +q axis (measuring counterclockwise when viewing from +s).
Surface Sine	amp (2) freq (2) phase (2)	Generates a sinusoidal wave on surface of geometry surface by defining sine errors in both the geometry's local u and v directions. Each sine error is defined according to an amp, freq, and phase. Freq - frequency of curve along entire geometry direction. Phase - phase of curve at start of geometry direction.
Taper	loc angle	Geometry's radius tapers (inward or outward). Start of taper at loc, measuring along axis from geometry's local origin. Angular difference between ideal surface and tapered surface = angle°. (angle > 0 → outward taper. angle < 0 → inward taper.)
Radius Step Along Axis	loc amp	Geometry's radius steps beyond point loc. along axis (measuring from geometry's local origin). Step is amp units (positive or negative).
Radius Step Around Axis	amp start extent	Geometry's radius steps "amp" units for part or all of geometry's circumference. Start of step will be at start° from local +q axis. Step will extend extent°.

For a taper error, the parameterization above only models tapers which begin beyond a location (loc) along the positive direction of the axis. To model a taper that ends at a point along the axis, start a taper somewhere before the start of the axis (possibly combined with a step) and combine it with the negative of that taper, located where the taper is to end.

VIII. Sampling Plans

The ATS makes use of sampling plans to generate data sets from a geometry test definition. A sampling plan specifies the number of points which will be sampled, and the method with which the points are selected. The ATS defines three types of plans:

Equispaced - Points are sampled at regular intervals along the geometry.
Stratified - The geometry is divided into a specified number of bins (in one or two dimensions), and a random point is sampled from each bin.[1]
Random - Points are taken at random on the surface of the geometry.

All sampling plans apply both to edge and surface geometries. (Lines and circles are edge geometries. All others are surfaces.) For surface geometries, the equispaced and stratified sampling plans sample the surfaces in both their u and v (horizontal and vertical) directions. The number of points to be sampled in one direction is independent of the number to be sampled in the other direction. (I.e. the user may specify different numbers of points for each direction.)

For a one-dimensional sampling plan of n points, an equispaced plan would sample a non-cyclic geometry, such as a line, starting at its start and ending at its end, thereby dividing it into n-1 sections. A cyclic geometry, such as a complete circle, would be sampled from its start (its intersection with the local +q, see section IV) to one unit before returning to start. Thus, including the final section from the last sampled point to the circle start, the circle is divided into n sections. The same principle holds true for the cyclic directions of all geometries.

The ATS will avoid generating the same point twice in a sampling plan. This must be avoided when sampling the apex of a cone or the poles of a sphere. If, for example, the ATS had been sampling 5 points along the u direction of a sphere for each v location specified by the sampling plan, at v = -90° and v = 90°, one point only will be sampled.

[1] Bins are equally spaced in u-v parameters. Depending on the geometry, however, they may not all be the same size. To illustrate, although all the bins on a plane will be of equal size, the bins close to the poles of a sphere will be of smaller size than those towards the equator. Similarly, cones and tori will contain bins of unequal sizes.

IX. Measurement Errors

The ATS can simulate measurement errors in data it generates. It supports two types of errors:

Uniform - The magnitude of the error is a uniformly distributed random variable between zero and the measurement error amplitude (supplied by the user).

Normal - The magnitude of the error is a normally distributed random variable with zero mean and standard deviation given by the amplitude.

The actual measurement error generated for each data point will be a random value in the range [0,amp] (or with standard deviation = amp for normally distributed errors). It will be propagated in a random direction.

X. Name Templates

The ATS makes use of user-supplied name templates during processes which generate multiple ATS objects or files. Such processes include data set generation and data set and fit import and export. A name template is a character string with optional embedded counter substitution variables. During the generation process, unique names are created for each generated object through the replacement of the counter variables with actual count values. Each of the different processes has slightly different template requirements. We first list the types of templates needed for the above processes, and then discuss the syntax rules for each one. A complete definition of the syntax of each template is beyond the scope of this section. Appendix B, at the end of this document, provides a formal set of grammar rules in Backus Naur form (BNF).

X.A Types of Templates

The following table lists the different types of templates found within the ATS:

Template Name	Meaning, Usage
dsgen	generated data set template, for the generation of multiple data sets
dsimp	imported data set template, for the importing of multiple data sets
dsimpfile	data set import file template, used to refer to multiple data set files for the importing of data sets from file
dsexpfile	data set export file template, used for the exporting of multiple data sets to file
fitgen	generated fit template, for the generation of multiple fits

Template Name	Meaning, Usage
fitimp	imported fit template, for the importing of multiple fits
fitfile	fit file template, used to refer to multiple fit files both for the importing of fits from file, and the exporting of fits to file
atsfit	ats fit template, for the importing of multiple fits of ATS format[1]
atsfile	file template for fits of ATS format, used to refer to multiple fit files for the import of ATS-format fits from file

X.B General Template Structure

With the exception of the fitgen template, the templates are all quite similar. They differ only in their valid *counter id* lists (explained below). Their general grammar rules are first presented, and then the differences are considered. The fitgen template will be discussed separately.

A template consists of a character string (of zero or more characters) containing zero or more embedded *counter variables* each of which may be placed at any point within the string. Each counter variable has the following format:

/<counter id><counter modifiers>/

(The starting and ending "/" must be entered literally. They are used to distinguish the counter variables from the character strings.) The counter id is a variable which will be replaced by a value at each iteration of the process. The counter modifiers are one or more optional parameters which modify the appearance of the counter id. They modify such aspects as the start value at which the count should start, or the width of the field in which the counter id should be displayed.

During template expansion, the character parts of the template will appear as is in the final string. Each counter id will be replaced by the value of the parameter referred to by that id, and the appearance of each counter id will be adjusted by the counter modifiers.

X.C Counter IDs

Each templates has its own list of legal counter id's. The following is the list of legal dsgen id's.

- **`d`** - the current data set index
- **`g`** - the current nominal geometry index
- **`f`** - the current form error index

[1] The ATS format is discussed in section X.

s - the current sampling plan index
m - the current measurement error index

The dsimp and dsimpfile templates use the same **d** and **g**. However, **f**, **s**, and **m** do not have significance (form errors, sampling plans, and measurement errors of imported files are unknown to the ATS). The meaning of **g** is as follows: Each imported data set receives a nominal geometry definition according to its parent test's nominal geometry schema. The first imported data set will be assigned the first nominal geometry instance, the second will have the second nominal geometry instance and so forth. If there are more imported data sets than there are nominal geometry instances, the schema will restart and the value of **g** will be reset to 1. Template dsexpfile allows **d** only.

The fitimp and fitfile templates are equivalent. They both require counter id's which refer to the number fit we are up to in an import or export command, and the number of the current data set (to which we are importing or exporting fits). The legal id's are:

d - the current data set index
f - the current fit index (per its parent data set)

The **d** modifier is most meaningful for importing/exporting fits from the test level (over a range of data sets). **d** will reflect the current data set index over the range of values. However, when importing/exporting from the data set level, **d** will expand into the same value (the index of the same data set) for all fits.

The atsfit and atsfile templates are equivalent. Both require a single counter id to refer to the number fit the ATS is up to. Its legal id, therefore, is:

f - the current fit index (i.e., the number fit we are currently importing)

It should be noted that **f** has a slightly different connotation here than above for the fitimp and fitfile templates. Above, the meaning was the number fit we are importing/exporting per the current data set. Thus, if we were importing one fit per data set for multiple data sets, **/f/** would expand into "1" for every fit (we might be adding a new "Fit 1" at the end of every data set's fit list). For the ats templates, however, **f** expands into the number fit we are currently importing out of the total number of fits to be imported. Thus **/f/** will expand into 1, 2, 3, ...

All counter indices start at 1. All import or export file name templates have the additional restriction that the resulting string must be a legal DOS file name. (The ATS does not check for DOS legality. If it can't carry out an import or export, it will display an error message.)

X.D Counter Modifiers

The counter modifiers adjust the appearance of the value generated from a counter id. The following chart explains each modifier.

+<*integer*> - The count should be based at <*integer*> rather than at 1.

+A/a - The count should be in characters based at A/a rather than 0. (The count will progress A, B, C, ... Y, Z, AA, AB, ... AY, AZ, BA, BB ....) '+A' will produce a count in uppercase characters, '+a' in lowercase.

:<*integer*> - Use *integer* as the minimum width to use to display this counter value. If *integer* exceeds the width of the entire counter value, the value is right justified in the width. Also, if the count is numeric, it is padded with 0's. If it is a character count, it is padded (by default) with '_'s.

'<*alpha*> - (*alpha* should be a single character.) Use *alpha* as a pad character if our count will be in characters and our minimum width is greater than the size of the data set variable. (The default pad character is '_'.)

X.E Fit Generation Template

The fit generation template has all of the features of the other templates. It allows two counter id's, **/d/** and **/f/**, which refer to the current data set and fit indices respectively. (The fit count refers to the fit index per data set (if we are generating fits for multiple data sets).) Its counter id's allow the same counter modifiers as above. It has an additional feature, the *fit variable*, which allows each fit name to include a string which reflects the type of fit objective used in the fit's generation. A brief description of the fitgen template is provided here. The complete syntax is described in Appendix B.

A fit template consists of a character string (of zero or more characters), zero or more embedded *counter variables*, and zero or more *fit variables*, each of which expands into a character code reflecting the fitting objective used in the generation of the fit. Each counter variable and fit variable may be inserted at any point within the string. The format for a fit variable is:

/t/ or /t:<type strings>/

At fit generation, "t" will be expanded into a character code depending on the type of fit. (The fit objectives themselves are described briefly in section XI.) The type strings are optional modifiers which allow you to change the default character code for each type of fit.

The fit character codes are as follows:

"L2" least squares
"L2_3D" three-dimensional least squares
"MC" minimum circumscribed (not applicable to lines or planes)
"MI" maximum inscribed (not applicable to lines or planes)
"MZ" minimum zone

(Routines "MI" and "MZ" are not implemented as of ATS version 2.0. The syntax for them, however, is already defined.)

The default character codes may be overridden using the longer version of the fit variable. In the <type strings> section, you may specify the replacement string for each code. For example, the template,

```
/t:L2=l-2;MC=min c/
```

would replace code L2 with "l-2" and MC with "min c". As shown in the example, you may override as many codes as you like. Individual type strings are separated by semicolons. Type string may contain any characters except '/' and ';'.

X.F Examples of Name Templates

Examples of generated data set name templates and their expansions are shown in the table. (As above, all other templates (other than the generated fit) are similar, but have fewer counter id's and are therefore simpler.) The expansion examples assume the particular data set is the 120'th data set in the test, corresponding to the first nominal geometry, the second form error, the third sampling plan, and the fourth measurement error.

TEMPLATE	EXPANSION
DS /d/	DS 120
DS /d:2/	DS 120
DS /d:4/	DS 0120
DATASET./f/-/m/	DATASET.2-4
DATASET:/g+A:2/,/s+8/	DATASET:_A,11
DS f=/f+a'#:4/	DS f=###b

Examples of fit name templates and their expansions are shown in the table below. The expansion is for a least-squares fit, which is the third fit for the data set.

TEMPLATE	EXPANSION
Fit /f:2/	Fit 03
/t/ Fit	L2 Fit
/t:L2=L-2;MC=Min C/	L-2

XI. File Formats

This section discusses the different data set file and fit file formats used by the ATS. It deals with both the required format for the basic information contained in a file, and the different types of header information that may be included in a file. This section will therefore be divided into two main parts, a discussion of data set files and a discussion of fit files. Each part will further be subdivided into a discussion of basic information, and of header information.

XI.A Data Set Files

The ATS can import data set files in five distinct formats: Generic, Raw, PTB_A, ATS, and Binary. It can export files in all the above formats except PTB_A. With the exception of Binary, all formats are ASCII file formats.

XI.A.1 Basic Information

All the ASCII file formats are equivalent in terms of the basic file information. They each contain the list of xyz coordinate points. Each value must be separated by one or more white spaces (such as spaces, tabs, or end-of-line characters). Any other separation characters (such as commas) will cause unpredictable results. The values may be expressed in decimal floating point or in exponential notation (using 'E' or 'e' to separate the mantissa from the exponent). Typically, the three coordinate values comprising an xyz coordinate point would be on a single line, but this is not required by the ATS. Binary format contains the list of coordinate points in binary (64-bit IEEE real format).

Two file formats, Generic and ATS, include in their header information the number of points contained in the file (see next section). Files in these formats may contain additional information or comments at the end of the data point list. Since the ATS knows how many points to read, it will know where to stop, and will ignore the remainder of the file. Files in any other format may not include any additional information beyond the data points.

XI.A.2 Header Information

The various data set file formats all differ slightly in header information. The following table summarizes:

Format	Header Information	
	Amount	**Meaning**
Generic	one integer	number of points in data set.
Raw	none	N/A
PTB_A	one line	bookkeeping information used by PTB. Stored in the "comments" of the data set.

Format	Header Information	
	Amount	**Meaning**
ATS	two lines	information describing data set's location within the ATS hierarchy and number of points in data set. See below for details.
Binary	none	N/A

Data set files in ATS format contain the following header information and additional information:

1) The first line has the following format:
   ID <geometry name> <date>
   where,

ID - a 32 character alphanumeric string which identifies the location of this data set/fit within the ATS hierarchy.

geometry name - the name of the geometry this data set is attached to.

date - the date of the data set's generation, in the form DD-MMM-YYYY, for example 05-NOV-1994 (my birthday).

2) The second line contains the number of points in the data set.
3) Anywhere after the first line, a ';' may appear which signifies that the remainder of the current line contains a comment. Comments are discarded by the ATS when such a file is read.

XI.B Fit Files

The ATS can import and export fit files containing fit results. It can import fit files in two distinct formats, Raw and ATS. It can export files in Raw format only. Both formats are ASCII file formats.

XI.B.1 Basic Information

A fit file contains a list of fit parameters describing the fitted geometry. The ATS defines the precise file format for fit files. This includes which parameters are required for each geometry, and the order in which the parameters must appear. This information is summed up in the table below. In general, the parameter values must be separated by one or more white spaces (such as spaces, tabs, or end-of-line characters). Any other separation characters (such as commas) will cause unpredictable results. These parameter values may be expressed by decimal floating point or exponential notation. The ATS ignores information or comments in the file beyond the last parameter.

Geometry	File Format	
	Parameter Number	**Parameter Description**
Line	1, 2, 3	coordinates of a point on line
	4, 5, 6	direction numbers of line
Circle	1, 2, 3	coordinates of center
	4, 5, 6	direction numbers of normal to plane of circle
	7	diameter
Plane	1, 2, 3	coordinates of a point on plane
	4, 5, 6	direction numbers of plane normal
Sphere	1, 2, 3	coordinates of center
	4	diameter
Cylinder	1, 2, 3	coordinates of a point on axis
	4, 5, 6	direction numbers of axis
	7	diameter
Cone	1, 2, 3	coordinates of a point on axis
	4, 5, 6	direction numbers of axis
	7	perpendicular distance from surface of cone to point specified by parameters 1-3
	8	full cone angle
Torus	1, 2, 3	coordinates of center
	4, 5, 6	direction numbers of torus normal
	7	major diameter
	8	minor diameter

XI.B.2 Header Information

A fit file in Raw format contain no header information. In ATS format, a file contains the same first line header information as do data set files in ATS format (see section X.A.2). The second

line of the ATS format, which contains the number of points in the data set is not applicable to fit files and is not part of the ATS format for fit files.

XII. Fitting Routines

In order to test geometric fitting software, the ATS provides its own fitting routines. These routines are used to generate *reference fits* for all data sets generated by, or imported into the ATS. To evaluate fitting software, the software is given data sets to fit. The fit result for each data set is then compared to the ATS reference fit for the same data set. The fitting routine used most extensively by the ATS is least squares, orthogonal distance regression. For certain geometries, however, other routines are supported. When evaluating fitting software performance, fits produced using any of the ATS-supported routines may be designated as the reference.

The following is the list of fitting routines supported by the ATS version 2.0, and the geometries which they support.

Least Squares (L2) - orthogonal distance regression. This is available for all geometries. For circles, L2 fitting is handled slightly differently - the plane of the circle is found by an L2 fit, and the points are projected onto the fit plane. The circle is then found as a two-dimensional fitting problem.

Three-Dimensional Least Squares (L2_3D) - full 3-D orthogonal distance regression for circles only. This differs from L2 fitting in that the plane of the circle is found simultaneously to the center and radius. The data points are not projected onto the plane (as with L2 fitting). Rather, residuals are the three-dimensional distance from the points to the circle.

Minimum Circumscribed (MC) - available for circles and spheres. It determines the smallest geometry that encloses all the data points. Circle fitting is done as with L2 fitting: the plane of the circle is determined by an L2 fit and the points are projected onto the plane.

XIII. Analysis

The ATS compares fit results through the performance of an *analysis*. This section discusses the parameters the ATS computes in an analysis and the results it returns. A complete discussion of the parameters calculated by the ATS and their corresponding uncertainties is not provided. A separate paper, has been dedicated to this. (See reference #5 in Introduction, section I.B.)

XIII.A Definition

An analysis compares each fit under test to a *reference fit*. A reference fit is the result which is considered "correct" for a given data set, and which all fits under test are compared against. The reference fit will typically be a fit result generated by one of the ATS fitting routines. It can also be any fit designated by the user, or the nominal geometry specification itself. Analyses can be performed at the geometry level of the ATS hierarchy (to evaluate all the fits belonging to all the data sets of all the tests of a geometry), at the test level (all the fits of all its data sets), and at the data set level.

The analysis computes the root mean square (RMS) values of various geometric differences between the fits under test and the corresponding reference fits. It computes corresponding uncertainties for all difference values. It also maintains a list of the worst-case fit for each difference. The bulk of this section discusses the specific geometric differences which are calculated for each geometry type. The discussion is divided according to geometry. The conclusion of this section discusses the results reported by an ATS analysis.

XIII.B Lines

The ATS performs an analysis of each line fit in the following manner: First, the fitted line is bound. (A line fit description defines the location and direction of the line, but not its extent. See section X.B.1.) It is bound by perpendicularly projecting the points of the data set onto the fitted line. Two difference parameters between the fit and the reference are then computed:

Separation - the largest distance from the bounded fit line to the reference line.
Line Tilts - the angle between the reference line and the fit line.

XIII.C Circles

The ATS performs an analysis of each circle fit by calculating the following difference parameters between the fit and the reference: (Bounding is not necessary as circles are naturally bound.)

Separation - the distance between the centers of the test and reference circles (a 3-dimensional distance).
Plane Tilts - the angle between the planes of the test and reference circles.
Radius - the absolute difference between the radii of the two circles.

XIII.D Planes

The ATS performs an analysis of each plane fit in the following manner: First, the fitted plane is bound. (A plane fit description defines the location and direction of the plane, but not its extent. See section X.B.1.) It is bound by perpendicularly projecting the points of the data set onto the fitted plane. Two difference parameters between the fit and the reference are then computed:

Separation - the largest distance from the bounded fit plane to the reference plane.
Plane Tilts - the angle between the test and reference planes.

XIII.E Spheres

The ATS performs an analysis of each sphere fit by calculating the following difference parameters between the fit and the reference: (Bounding is not necessary as spheres are naturally bound.)

Separation - the distance between the centers of the test and reference spheres.
Radius - the absolute difference between the radii of the two spheres.

XIII.F Cylinders

The ATS performs an analysis of each cylinder fit in the following manner: First, the axis of the fitted cylinder is bound. (A cylinder fit description defines the location, direction and diameter of the cylinder, but not the extent of its axis. See section X.B.1.) The cylinder is bound by perpendicularly projecting the data points of the data set onto the cylinder's axis. Three different parameters between the fit and the reference are then computed:

Separation - the largest distance from the bounded fit cylinder axis to the axis of the reference cylinder.
Axis Tilts - the angle between the test and reference axes.
Radius - the absolute difference between the radii of two cylinders.

XIII.G Cones

The ATS performs an analysis of each cone fit in the following manner: First, the axis of the fitted cone is bound. (A cone fit description defines the cone location, direction and angle, but not the extent of its axis. See section X.B.1.) The cone is bound in a two-stage process. First, the data set is projected onto the surface of the fitted cone. Then, the projected points on the surface are perpendicularly projected onto the fitted cone axis. This last projection bounds the cone axis. The following parameters are then computed:

Separation - the largest distance from the bounded fit cone axis to the axis of the reference cone.
Axis Tilts - the angle between the test and reference axes.
Distance - the absolute difference between the test and reference "distances". Cone "distance" means the perpendicular distance from the midpoint of the bounded axis to the surface of the cone.

Angle - the absolute difference between the test and reference cone half angles.

XIII.H Tori

The ATS performs an analysis of each torus fit by calculating the following difference parameters between the fit and the reference: (Bounding is not necessary as tori are naturally bound.)

Separation - the distance between the centers of the test and reference tori.
Axis Tilts - the angle between the normal to the plane of the fitted torus, and the normal to the plane of the reference torus.
Major Radius - the absolute difference between the major radii of the two tori.
Minor Radius - the absolute difference between the minor radii of the two tori.

XIII.I Reported Results

For each of the geometric difference parameters considered in the analysis of a collection of fits, the ATS returns two sets of results, the RMS (root mean square) value of the differences, and the worst case fit for that parameter.

The ATS returns corresponding uncertainty values for each RMS difference returned. The uncertainty of each RMS value is determined by the combined effects of: (1) the statistical uncertainty of estimating the RMS value from a sample of fits and, (2) if applicable, the uncertainty associated with the calculation of the reference fit. These figures are described in much greater detail in Hopp, Levenson, "Performance Measures for Geometric Fitting in the NIST Algorithm Testing and Evaluation Program for Coordinate Measuring Systems".

When comparing fits to an ATS-generated reference fit, the uncertainty associated with the calculation of each parameter of the reference is used in estimating the overall uncertainty of each parameter of the analysis. When, however, comparing fits to the nominal geometry or to an imported fit, the uncertainty associated with the reference fit is assumed to be zero.

For each of the difference parameters computed in an analysis, the ATS stores the value of the worst-case fit. Some of the difference parameters can have both positive and negative values, (i.e., for a given fit, each parameter may be less than or greater than the reference fit value). These are: radius (or distance, for a cone), angle (i.e., cone half angle), and the minor radius of a torus. Thus, two worst-case values for each are stored - the lowest parameter under the reference, and the greatest parameter over. The RMS summary, however, only returns one parameter for each of these differences, since it calculates a single overall RMS value. Thus, the worst-case values provide additional information about the distribution of differences.

APPENDIX A - List of Menu Commands

Appendix A provides a comprehensive listing of all menu commands supported by the ATS. This appendix is divided into six tables. The first describes the menu selections which are available at all levels of the ATS hierarchy. The next five tables correspond to the five levels of the ATS hierarchy (and five panels on the ATS screen), and describe the particular menu selections available at each level. For each level, the following information is provided: (1) the menu sections available, (2) the menu commands contained in each section (and the equivalent hotkey (if any)), (3) the meaning of each command, and (4) the restrictions limiting when each command is available.

A.1 Menu Commands - All Levels

Section	Command	Function	When Selectable
Program	Repaint Display (Ctrl+L)	Redraw display	Always[1]
	Go to DOS	Suspend program, go to DOS[2]	Always
	Quit	Quit program	Always
Help	Topics	Invoke ATS on-line help with display of basic ATS topics	Always
	Context Sen-sitive (F1)	Invoke ATS on-line help for window in which cursor was most recently located	Always
	About the ATS	Display window containing basic ATS information	Always

[1] In general, menu commands are not selectable when a popup window or status window appears. Under such circumstances, the ATS "status" (appearing in the upper-right corner of the ATS screen) will be "Wait", "Enter", or "Prompt". Menu commands are only selectable when the status button is displaying "Ready".

[2] To return to the ATS, type "EXIT" and press ENTER on the keyboard. When the ATS is suspended, it is still in memory. Therefore, your machine may not have enough memory to run large programs. (Expect no more that about 100K of free memory.) Also, be careful not to modify any ATS files while the ATS is suspended.

A.2 Menu Commands - Experiment Level

Section	Command	Function	When Selectable
Disk	Print Experiment (Ctrl+P)	Print experiment on which cursor is positioned	If 1 or more experiments exist[1]
	Print Analysis	Print most recently performed analysis	If an analysis was performed and is still available[2]
	Import ATS Fits	Import fits of ATS format from files[3]	If an experiment is loaded[4]
Analyze	Show Last Analysis	Show most recently performed analysis	If an analysis was performed and is still available[1]

[1] An important distinction is appropriate: A menu command is made selectable as soon as the current panel satisfies certain conditions. Here, once one or more experiments exist in the panel, *"Print Experiment"* becomes selectable. The command itself, however, will only function if the cursor is additionally placed on an experiment within the panel. If, however, the cursor is placed on the blank line of the panel (the line after the experiment list), the *"Print Experiment"* command will be selectable but will not execute.

[2] An analysis is only available if no changes have been made to the ATS which would render it obsolete or unavailable. If you have made further changes to the ATS (such as generating/deleting fits), the analysis may become obsolete. If you have loaded other objects, it may be unavailable.

[3] The fits will be imported to locations within the ATS hierarchy which are specified in the fits files themselves.

[4] This command will function properly if the fits belong to the loaded experiment, they contain the proper information within them, and no tests or data sets have been added to or deleted from the loaded experiment since the data sets the fits were generated from were exported. In truth, there's little way of knowing in advance if all these conditions will be satisfied. Thus, this command is always legal so long as an experiment is loaded. It is up to the user to ensure that all conditions are met. If not, the ATS will flag an error, or the fits will be imported to the wrong locations within the ATS.

A.3 Menu Commands - Geometry Level

Section	Command	Function	When Selectable
Disk	Print Geometry (Ctrl+P)	Print geometry on which cursor is positioned	Always[1]
	Print Analysis	Print most recently performed analysis	If an analysis was performed and is still available[2]
	Import ATS Fits	Import fits of ATS format from files	Always[3]
Analyze	Analyze Fits	Analyze all fits attached to loaded geometry	If a geometry is loaded and has 1 or more test-children[4]
	Show Last Analysis	Show most recently performed analysis	If an analysis was performed and is still available[2]

[1] Geometries always exist: It is possible to reach this panel only if an experiment is defined and loaded. As soon as an experiment is loaded, the ATS automatically loads its seven geometries. They do not have to be inserted by the user since they are predefined. Also, see footnote 1, experiment level.

[2] See footnote 2, experiment table above.

[3] See footnote 3, experiment table.

[4] Ideally, this command should be legal only if we know fits exist to be analyzed (and they're not reference fits, tagged fits etc.), but this is too expensive to determine from the geometry level. If no fits to be tested exist, the menu command will be selectable, but will return an error message upon execution.

A.4 Menu Commands - Test Level

Section	Command	Function	When Selectable
Disk	Print Test (Ctrl+P)	Print test on which cursor is positioned	If 1 or more tests exist[1]
	Print Analysis	Print most recently performed analysis	If an analysis was performed and is still available[2]
	Import Data Sets	Import data sets from files; attach to loaded test[3]	If a test is loaded

Section	Command	Function	When Selectable
Disk (con't)	Export Data Sets	Export dataset-children of loaded test to files	If a test is loaded and has 1 or more dataset-children
	Import Fits	Import fits from files; attach to dataset-children of loaded test[4]	If a test is loaded and has 1 or more dataset-children
	Import ATS Fits	Import fits of ATS format from files	Always[5]
	Export Fits	Export all the fits belonging to all the data sets of loaded test to files	If a test is loaded and has 1 or more dataset-children[6]
Generate	Generate Data Sets	Generate data sets from test definition of loaded test	If a test is loaded and defines 1 or more test cases
	Generate Fits	Generate fits for the dataset-children of loaded test	If a test is loaded and has 1 or more dataset-children
	Copy Test (Ctrl+C)	Create new test with same schema definitions as loaded test (but no children)[7]	If a test is loaded
Analyze	Analyze Fits	Analyze all the fits attached to loaded test	If a test is loaded and has 1 or more dataset-children[6]
	Show Last Analysis	Show most recently performed analysis	If an analysis was performed and is still available[2]
	Tag Test ('*')	Tag test on which cursor is positioned	If 1 or more tests exist[1]

[1] See footnote 1, experiment table.

[2] See footnote 2, experiment table.

[3] In general, whenever importing objects to a level lower than the current, the newly imported objects are placed at the end of the already-existing list of objects. Here, the newly imported data sets are added to the end of the current data set list.

[4] The fits will be added after any existing fit lists for the data sets, see preceding footnote.

[5] See footnote 3, experiment table.

[6] Ideally, this command should be legal only if we know fits exist to be exported or analyzed but this is too expensive to determine from the test level. If no fits to be tested exist, the menu command will be selectable, but will return an error message upon execution.

[7] The new test is placed immediately after the current one.

A.5 Menu Commands - Data Set Level

Section	Command	Function	When Selectable
Disk	Print Data Set (Ctrl+P)	Print data set on which cursor is positioned	If 1 or more data sets exist[1]
	Print Analysis	Print most recently performed analysis	If an analysis was performed and is still available[2]
	Import Data Sets	Import data sets from files; place at present cursor location	Always
	Export Data Set	Export loaded data set to file	If a data set is loaded
	Import Fits	Import fits from files; attach to loaded dataset[3]	If a data set is loaded
	Import ATS Fits	Import fits of ATS format from files	Always[4]
	Export Fits	Export fit-children of loaded data set to files	If a data set is loaded and has 1 or more fit-children
Fit	Compute Fit	Compute fit(s) for loaded data set	If a data set is loaded
Analyze	Analyze Fits	Analyze all the fits attached to loaded data set	If a data set is loaded and has 1 or more fit-children[5]
	Show Last Analysis	Show most recently performed analysis	If an analysis was performed and is still available[2]
	Tag Data Set ('*')	Tag data set on which cursor is positioned	If 1 or more data sets exist[1]

[1] See footnote 1, experiment table.

[2] See footnote 2, experiment table.

[3] The fits are added to the end of the data set's current fit list. See footnote 2, test table.

[4] See footnote 3, experiment table.

[5] Even so, the analysis may fail for a number of reasons. For example, if the loaded data set has only one fit, the fit may be tagged or the reference fit. This, however, is too costly to check. If such an error occurs, the analysis will return with an error message.

A.6 Menu Commands - Fit Level

Section	Command	Function	When Selectable
Disk	Print Fit (Ctrl+P)	Print fit on which cursor is positioned	If 1 or more fits exist[1]
	Print Analysis	Print most recently performed analysis	If an analysis was performed and is still available[2]
	Import Fits	Import fits from files; place at present cursor location	Always
	Import ATS Fits	Import fits of ATS format from files	Always[3]
	Export Fit	Export loaded fit to file	If a fit is loaded
Analyze	Show Last Analysis	Show most recently performed analysis	If an analysis was performed and is still available[2]
	Tag Fit ('*')	Tag fit on which cursor is positioned	If 1 or more fits exist[1]

[1] See footnote 1, experiment table.

[2] See footnote 2, experiment table.

[3] See footnote 3, experiment table.

APPENDIX B - Formal Syntax for ATS Templates

Appendix B formally defines the proper syntax for the different types of templates supported by the ATS. Templates were discussed in full above in Section X. This appendix provides the formal rules only. Section B.1 presents the rules for the generated data set template. (Recall that most of the other templates are similar in structure to those for generating data sets. The differences lie primarily in the fact that fewer counter id's are valid in the other templates. Thus, most of the other templates have rules similar to, but simpler than, the ones presented here.) Section B.2 presents the rules for the generated fit template. All rules are presented in Backus Naur form (BNF), a standard method for notating formal grammars.

B.1 Syntax for the Generated Data Set Template

In the syntax rules below, the generated data set template is referred to as the "ATS template" since similar rules apply to nearly all ATS templates.

<ATS template> :: <*character*><ATS template> | <*character*>
| <ATS variable><ATS template> | <ATS variable>

<ATS variable> :: **/**<ATS counter id>**/**
| **/**<ATS counter id><counter modifiers>**/**

<ATS counter id> :: **d** | **g** | **f** | **s** | **m**

<counter modifiers> :: <counter modifier><counter modifiers> | <counter modifier>

<counter modifier> :: **+**<*integer*> | **+A** | **+a** | **:**<*integer*> | **'**<*alphanumeric*>

B.2 Syntax for the Generated Fit Template

The grammar rules for the generated fit template are defined below. "Counter modifiers" refer to same "counter modifiers" defined above in section 1.

<fit template> :: <*character*><fit template> | <*character*>
| <fit variable><fit template> | <fit variable>

<fit variable> :: **/**<fit counter id><counter modifiers>**/**
| **/t:**<type strings>**/** | **/t/**

<fit counter id> :: **d** | **f**

<type strings> :: <type string>**;**<type strings> | <type string>

<type string> :: **L2=**<*string*> | **L2_3D=**<*string*> | **MC=**<*string*>
| **MI=**<*string*> | **MZ=**<*string*>