EnSight User Manual

for Version 9.2

Table of Contents

Computational Engineering International, Inc.
2166 N. Salem Street, Suite 101, Apex, NC 27523
USA • 919-363-0883 • 919-363-0833 FAX
http://www.ceintl.com or http://www.ensight.com

EN-UM Revision History

EN-UM:5.2-1	October 1994
EN-UM:5.2.2-1	January 1995
EN-UM:5.5-1	September 1995
EN-UM:5.5.1-1	December 1995
EN-UM:5.5.2-1	February 1996
EN-UM:6.0-1	June 1997
EN-UM:6.0-2	August 1997
EN-UM:6.0-3	October 1997
EN-UM:6.0-4	October 1997
EN-UM:6.1-1	March 1998
EN-UM:6.2-1	September 1998
EN-UM:6.2.1-1	November 1998
EN-UM:7.0-1	December 1999
EN-UM:7.1-1	April 2000
EN-UM:7.3-1	March 2001
EN-UM:7.4-1	March 2002
EN-UM:7.4-2	October 2002
EN-UM:7.6-1	May 2003
EN-UM:8.0-1	December 2004
EN-UM:8.2-1	August 2006
EN-UM: 9.0.-0	September 2008
EN-UM: 9.1.-0	December 2009
EN-UM: 9.2.-0	December 2010

This document has been reviewed and approved in accordance with Computational Engineering International, Inc. Documentation Review and Approval Procedures.

This document should be used only for Version 9.2 and greater of the EnSight program.

CEI's World Wide Web addresses:
http://www.ceintl.com
or
http://www.ensight.com

Restricted Rights Legend

Apex, NC 27523 USA

Table of Contents

3 Parts

7 Features

8 Modes

9 Transformation Control

10 Preference and Setup File Formats

11 EnSight Data Formats

12 Utility Programs

13 Parallel and Distributed Rendering

1 Overview

EnSight (for Engineering inSight) provides engineers and scientists with an easy-to-use graphics postprocessing package. EnSight supplies powerful, easy-to-use tools through a user-friendly interface.

The purpose of this chapter is to give you an overview of the EnSight system and its documentation. Because of the power and flexibility of EnSight, the synergy between features provides a great many visualization techniques.

The Overview topics discussed are:

Part Concepts

EnSight processing begins with your model. Usually the elements of your model are grouped into parts. *Within EnSight, nearly all information is associated with parts, and nearly all actions are applied to parts*. The current maximum number of parts is 65000.

Geometry

A part consists of *nodes* and *elements* (elements are sets of nodes connected in a particular geometric shape). Each node, which is shared by its adjoining elements, is defined by its coordinate-location in the model frame of reference.

Variable Values

EnSight-compatible data files provide variable values either at each part's nodes, element centers, or both. When needed (or requested) EnSight will find any variable's value at any point on or within an element by utilizing the element's shape function. The current maximun number of variables is 10000.

Part Attributes

Within EnSight, you can specify additional information about each part. These *part attributes* tell EnSight how to display each part and how the part responds to EnSight controls and display options. Part attributes include:

Category	Includes attributes that control....
General Attributes	Visibility (including visibility per viewport) Susceptibility to Auxiliary Clipping Reference frame Response to changes in time (frozen or active) Fast display representation Visual Symmetry options Coloration (constant, by a palette associated with a variable, or a 2D texture) Shaded Surface susceptibility Surface shading (flat, Gouraud, smooth) Hidden Line susceptibility Fill density (for transparency) Lighting (diffuse, shininess, and highlight intensity)
Node, Element, and Line Attributes	Node visibility Node type (dot, cross, or sphere) Node size (constant or variable) Node detail (for spheres) Element-line visibility Element-line width Element-line style (solid, dotted, or dot-dash) Element representation on client (full, border, 3D border/2D full, 3D feature/2D full, 3D nonvisual/2D full, bounding box, feature angle, or nonvisual) Element-size shrink-factor Polygon Reduction Failed element variable and rules Node and element label visibility
Displacement Attributes	Displacement variable and scale factor
IJK axis display attributes	IJK axis visibility and scale factor

Part Operations

Parts can be copied to show, for example, the same part colored by a different variable. Model parts can be split along an arbitrary plane or any quadric surface, and merged with other model parts. The geometry of parts can be simplified by creating a new part by extracting a simpler representation of an existing part.

Part Representation

Parts can be represented with simpler geometry, both to enhance visualization performance and for special effects. Representation modes include:

Full mode, which represents all the part's elements in the graphics window.

Border mode, which represents 3D elements with their 2D external faces.

Feature angle mode, which represents with 1D elements the "significant" edges of the part (you control what is "significant")

Nonvisual, which does not represent any of the part's elements in the graphics window (but still loads in memory on the server).

Bounding box, which represents the box bounding the coordinate extremes.

(see Section 3.3, Part Editing, and Section 8.1, Part Mode)

Data Types

EnSight supports a number of common data formats as well as interfaces to various simulation packages. If an interface is not available for your data, EnSight includes a library of routines to create your own custom User-Defined Reader. User-Defined Readers have the advantage of not requiring a separate data translation step and thus reduce user effort and disk storage requirements. A number of User-Defined Readers are provided with EnSight; complete documentation and dummy routines may be found in the directory $CEI_HOME/ensight92/src/readers.

There are four different means to get your data into EnSight.

Type 1 - Included Readers - Are accessed by choosing the desired format in the Data Reader dialog. These include common data formats as well as a number of readers for commercial software. These can be internal EnSight readers as well as User-Defined Readers. If the included reader is a User-Defined Reader there may be more info in a README file found in $CEI_HOME/ensight92/src/readers.

Type 2 - Not Included User-Defined readers - A number of User-Defined Readers have been authored by EnSight users, but are not provided with EnSight. See the Comments column below for more information. Also for more info see our website, www.ensight.com and go to the Support page.

Type 3 - Stand - Alone Translators - May be written by the user to convert data into EnSight format files. A complete description of EnSight formats may be found in Chapter 11 of the EnSight Online User Manual. Several translators are provided with EnSight. These are found in the directory $CEI_HOME/ensight92/translators. Translators must first be compiled before they may be used. Some require links to libraries provided by the vendor of the program in question. See the README files found in each translator's directory.

Type 4 - EnSight Format - A growing number of software suppliers support the EnSight format directly, i.e. an option is provided in their products to output data in the EnSight format.

The table that follows summarizes all of the data formats and software packages for which an interface of Type 1-4 exists. As this information changes frequently, please consult our website (www.ensight.com) or your EnSight support representative should you have any questions. If your format or program is not listed here, there is the possibility that an interface does indeed exist. Contact EnSight support for assistance. Should you create a User-Defined Reader or Stand-Alone Translator and wish to allow its distribution with EnSight, please send an email to this effect to support@ensight.com.

Data Format / Program	Type	Comments
ABAQUS_FIL	1	Direct reader for binary or ascii (.fil) files (ABAQUS STANDARD or EXPLICIT)
ABAQUS_ODB	1	Direct reader for binary .odb files (ABAQUS STANDARD or EXPLICIT). **Only available for platforms supported by ABAQUS**
ACUSOLVE	2	Contact vendor for information
ADAGIO	1	Use Exodus II reader
ADINA	3	Use I-DEAS neutral files and translators
ALEGRA	1	Use Exodus II reader
ANSYS	1	Direct reader for binary .rst, .rth, .rmq, .rfl files. (Internal legacy reader = Ansys; single part < 2Gb file reader = Ansys Results (v10); largefile multipart reader = Ansys-Multi-Part)

Data Format / Program	Type	Comments
AVUS	1, 4	Formerly COBALT. User reader - or - Exports EnSight Case Gold format
Case (EnSight6, EnSight Gold)	1	Native EnSight formats, EnSight6 Case and EnSight Gold Case. **(Gold reader handles SOS auto-distribute.)**
CAD	1	Direct Reader for CAD formats as well as STL
CFD++	4	Exports EnSight Case format
CFD-ACE	2	Contact vendor for DTF reader
CFD-FASTRAN	2	Contact vendor for DTF reader
CFDESIGN	1	Uses Tecplot files and reader
CFF	1	User reader for Common File Format from BOEING (WIND/CFF code). Source supplied with EnSight install, executable supplied on some platforms.
CFX4	1, 3	User reader, and translator (useful if results contain massed particles)
CFX5	1, 4	Code exports EnSight Case format, and direct reader available for all platforms except Macintosh
CFX-TASCflow	3	Converts TASCflow output to EnSight format (or use PLOT3D converter from vendor)
CGNS	1	User reader
COBALT	1, 4	See AVUS User reader - or - Exports EnSight Case Gold format
CRAFT	4	Exports EnSight Case Gold format
CRUNCH	4	Exports EnSight Case Gold format
CTH	1	User reader. **(Handles SOS internally)**
DMC	1	Use Exodus II reader
ECLIPSE	3	Contact CEI for details
ENSIGHT (EnSight 5)	1	Original EnSight format (unstructured)
ESTET	1	Direct reader
EXODUS II	1	User reader. **(Handles SOS internally)**
FAST UNSTRUCTURED	1	Direct reader for NASA FAST unstructured format
FEFLO	3	Contact vendor for information
FEMWATER	2	Use GMS reader
FENSAP	4	Contact vendor for information
FIDAP	1	Direct reader for FIDAP neutral (FDNEUT) files
FIELDVIEW	1	User reader for Fieldview 2.5 unstructured data
FINE/Aero	1	Use PLOT3D or CGNS files/reader
FINE/Turbo	1	Use PLOT3D or CGNS files/reader
FIRE	4	Code exports EnSight format
FLOW-3D	1	User reader for FLOW-3D results (flsgrf) files
FLUENT (particle files)	3	Converts Fluent particle file to EnSight format
FLUENT	1, 4	User reader for .cas and .dat files - or - Fluent exports EnSight Casefile format
FOAM	2, 4	Contact developer (Imperial College) for interface details
FPRP	1	User reader imports ASCII .gpp files
GASP	4	Exports EnSight Case format
GMS	2	User reader for GMS groundwater modeling framework, contact CEI for information
GUST	4	Exports EnSight Case format
HDF5	2	Contact CEI for information
HTS	1	LLNL hierarchical triangulated surface format.
INVENTOR	1	User reader for Inventor (.iv) files
I-DEAS	3	Translator for I-DEAS FEA neutral file
IMEX	2	Contact CEI
IO/API	2	User reader for MODELS 3 framework, contact CEI for information
KIVA	2, 3,4	Conversion routines to export EnSight format, contact CEI for info
LS-DYNA	1	User reader for d3plot files
MADYMO	1	Use LS-DYNA reader
MAGMA	2	Contact CEI

Data Format / Program	Type	Comments
MAYA ESC	4	Contact vendor for information
MEDINA BIF/BOF	1	User reader for Medina bif/bof files using Konfig file
MODELS 3	2	Use IO/API reader
MOVIE.BYU	1	Direct reader for MOVIE.BYU format files
MPGS 4.1	1	Direct reader for MPGS, EnSight's predecessor
MSC.ADAMS	4	Plug-in support
MSC.DYTRAN	1	User reader for MSC/Dytran archive (.arc) or data (.dat) files
MSC MARC	1	User reader for t16 and t19 files
MSC.NASTRAN	1	User reader for binary OP2 files
MSC.NASTRAN INPUT	1	User reader for NASTRAN geometry input (.nas, .dat, .bdf) files
MSC.PATRAN	3	Converts PATRAN neutral files to MOVIE.BYU format -or- export to EnSight using PATRAN macro
MUSES/Prism	2	User reader from Thermoanalytics
NCC	2	User reader interface to National Combustion Code, contact CEI for info
N3S	1	Direct reader for the EDF code N3S
NetCDF	1	User reader, contact CEI for information
NIFTI-1/Analyze	1	Analyze format with NIFTI extensions. (http://nifti.nimh.nih.gov)
NSMB	2	User reader developed by CERFACS and CSCS
NSU2D / NSU3D	4	Contact CEI for information
OVERFLOW	1	User reader (modified PLOT3D reader) for OVERFLOW files
PAM-FLOW	2	User-defined reader from ESI for native PAM-FLOW files
PHOENICS	1	Use PLOT3D file/reader, contact CEI for information
PLOT3D	1	Direct reader for PLOT3D and FAST structured formats
POLY-3D	3	Contact vendor for information
POLYFLOW	4	Outputs EnSight Case format
POWERFLOW	3	Contact CEI for information on interfaces available
PERMAS	1	User-defined BIF/BOF reader (see Medina BIF/BOF). **Not available on Mac**
PRESTO	1	Use Exodus II reader
PRONTO	1	Use Exodus II reader
PXI	1	User reader for Parallel Exodus Interface format
RADIOSS	1	User-defined reader
RADTHERM	2	User reader from Thermoanalytics
RESCUE	2	User reader for Schlumberger reservoir modeling framework, contact CEI for information
SCRYU	1	User reader
SC/TETRA	4	Exports EnSight Casefile format.
SILO	1, 3	Reads various formats supported by SILO API. **(Handles SOS internally)**
SPHINX	4	Code exports EnSight format
STAR-CD (Version 3.0.5 & up)	4	Code exports EnSight Casefile format (including particle data)
STL	1	Use the CAD User reader for STL geometry files
SUPERFORGE		Contact CEI
TAHOE	4	Contact CEI for information on interfaces available
TECPLOT	1	User reader for TECPLOT 7 structured and unstructured formats
TECPLOT ASCII	1	User reader for TECPLOT 360 ASCII
Telluride	4	Code exports EnSight format
UNCLE	2	User reader, contact CEI for details
UNIC-CFD	3	Contact vendor for details
USM3D	4	From TetrUSS code. Contact CEI for information
VECTIS	1, 3	User reader
Wavefront OBJ	1	Polygonal model file format.
XDMF 2.0	1	User reader for XDMF format files

Geometry
EnSight reads unstructured and structured geometric data grouped by parts. Data can be 0D, 1D, 2D or 3D.

Analysis Results
EnSight reads scalar, vector, complex scalar, complex vector, and tensor variable values associated with each node and/or element of the geometry. The loading of variable values is optional, and variables can be unloaded to free memory.

Measured Data
EnSight can read measured or computed particles (referred to as discrete particles in EnSight). Particles can have the same variables as the model geometry, or their own variables. Particles can be displayed as points, crosses, or spheres whose size can vary according to a variable value. Sphere smoothness is also controllable. Discrete particles can be time dependent with the geometry, or time dependent with a steady geometry.

(See EnSight Gold Measured/Particle File Format, in Section 11.1)

Cases
EnSight provides the capability to read and manipulate up to 16 datasets or models at a time. Each new “Case” is handled by its own Server process while the Client appropriately deals with merged variables, solution times, etc. This option allows both the recombination of models partitioned for parallel analysis and a number of comparative operations.

Graphical Environment

Parts are visualized in a main Graphics Window. You can create additional viewports and adjust their size to your needs. Each viewport has its own transformations (global, local, look-at, look-from, and Z-clip locations). Part visibility is also controllable in each viewport. The current maximum number of viewports is 16.

A separate “Show Selected Part(s)” window helps in identifying parts.

Hidden Lines and Shaded Surfaces
You can choose to shade surfaces and/or hide hidden lines for realistic views of your model. Visible element edges can be overlaid on shaded solid images.

Clipping
In addition to user-control of the front and back visual clipping planes of your workspace, you can visually cutaway parts or portions of parts along any plane using Auxiliary Clipping. Individual parts can be made immune to the effect, enabling you to look at parts inside of other parts.

Annotations
EnSight can display text-strings, lines, arrows, logos, entity labels, and color-map legends. Text annotations (which may include variables) can be made to automatically update for time-dependent data. The current maximum number of characters in a given annotation string is 1024.

Image Output
Screen images can be saved from within EnSight. Conversion to popular formats is under user control as the image is saved.

Perspective
You have your choice of a perspective view or an orthographic view. The latter is useful for comparing the position of parts and positioning EnSight tools.

Background Color or Image
You can specify a constant or blended color background for the main Graphics Window and independently for any Viewports displayed in the Graphics Window. You can also use an image as the background.

Transformations

The standard transformations of rotate, translate, and scale are available, as well as positioning of the Look-At and Look-From points. An automatic zoom control is available. The transformation-state (the specific view in the Graphics Window and Viewports) can be saved for later recall and use. Transformations can be performed with precision in a dialog, or interactively with the mouse. For the latter case, you can choose to represent the parts with bounding-boxes (or by points only or reduced elements) all the time or only while they are moving. Transformations can individually be reset by type.

(see Chapter 9, Transformation Control)

Frames

Transformations actually apply to frames—the parts attached to the frames transform right along with their frame. You can create new frames and transform them like parts (in a dialog or with the mouse), and change to which frame a part is attached. You control whether and how frames are displayed, enabling you to use them as rulers. Frames can have rectangular, cylindrical, or spherical coordinates.

Frames, and therefore all parts attached to them, can be “periodic”. Rotational or translational periodicity (as well as mirror symmetry) attributes are under user control allowing, for example, an entire pie to be built from one slice of the pie.

(see Section 8.5, Frame Mode and Section 9.3, Frame Transform)

Coloration

Parts can be constant colored, colored according to the value of a variable, or colored by a texture. This feature works for both lines and surfaces. The coloration of each part is an attribute of that part.

Variable Palettes

You control the value-color correspondence with a *palette*. A palette’s scale can be linear, logarithmic, or exponential. Palettes can have a continuous range of colors, or color bands. Off-the-scale parts or portions of parts can be made invisible.

(see Section 4.2, Variable Summary & Palette)

Created Parts

In addition to the *model parts* defined in the dataset, you can (and usually will) define additional *created parts* based on both the geometry *and* variable-values of existing *parent-parts.* Model parts and most kinds of created parts can be used as parent parts. Created parts have their own part attributes, including the *creation attributes* that define them, but *remain dependent upon their parent-parts.* A created part *automatically regenerates* if any of its parent-parts are changed in a way that will affect its representation.

Clips

A clip is a plane, line, box, ijk surface, xyz plane, rtz surface, quadric surface (cylinder, sphere, cone, etc.), or revolution surface passing through specified parent-parts. A clip can either be limited to a specific area (finite), or clip infinitely through the model. You control the location of the various clips with an interactive Tool or appropriate parameter or coefficient input.

A clip line or plane will either be a true clip through the model, or can be made to be a grid where the grid density is under your control.

Clip surfaces can be animated as well as manipulated interactively.

In most cases you will create a clip which is the intersection of the clip tool and the parent parts. This clip can either be a true intersection or all elements that cross the intersection surface (a "crinkly" surface). You can also choose to cut the parent parts into half spaces.

(see Section 7.9, Clip Create/Update)

Contours Contours are created by specifying which parts are to be contoured, and which variable to use. The contour levels can be tied to those of the palette or can be specified independently by the user.

(see Section 7.7, Contour Create/Update)

Developed Surfaces Developed Surfaces can be created from cylindrical, spherical, conical, or revolution clip surfaces. You control the seam location and projection method that will flatten the surface.

(see Section 7.21, Developed Surface Create/Update)

Elevated Surfaces Elevated Surfaces can be displayed using a scalar variable to elevate the displayed surface of specified parts. The elevated surface can have side walls.

(see Section 7.14, Elevated/Offset Surface Create/Update)

Extrusions Parts can be extruded to their next higher order. Namely a line can be extruded into a plane, a 2D surface into a 3D volume, etc. The extrusion can be rotational (such as would be desired for an axi-symmetric part) or translational.

(see Section 7.23, Extrusion Parts Create/Update)

Isosurfaces Isosurfaces can be created using a scalar, vector component, vector magnitude, or coordinate. Isosurfaces can be manipulated interactively or animated by incrementing the isovalue.

(see Section 7.8, Isosurface Create/Update)

Particle Traces Particle traces—both streamlines (steady state) and pathlines (transient)—trace the path of either a massless or massed particle in a vector field. You control which parts the particle trace will be computed through, the duration of the trace, which vector variable to use during the integration, and the integration time-step limits. Like other parts, the resulting particle trace part has nodes at which *all* of the variables are known, and thus it can be colored by a different variable than the one used to create it. Components of the vector field can be eliminated by the user to force the trace to, for example, lie in a plane. The particle trace can either be displayed as a line, a ribbon, or a square tube showing the rotational components of the flow field. Streamlines can be computed upstream, downstream, or both.

Streamline and pathline particle traces originate from *emitters*, which you create. An emitter can be a point, rake, net, or can be the nodes of a part. Each emitter has a particle trace emit time specified which you set, and a re-emit time (if the data case is transient) can also be specified. Point, rake, and net emitters can be interactively positioned with the mouse. For streamlines, the particle trace continues to update as the emitter tool is positioned interactively by the user.

Another form of trace that is available is entitled node tracking. This trace is constructed by connecting the locations of nodes through time. It is useful for changing geometry or transient displacement models (including measured particles) which have node ids.

A further type of trace that is available is a min or max variable track. This trace is constructed by connecting the min or max of a chosen variable (for the selected parts) though time. Thus, on transient models, one can follow where the min or max variable location occurs.

(see Section 7.11, Particle Trace Create/Update)

Profiles

Profile plots can be created by scalar, vector component, or vector magnitude. You control the orientation of the resulting profile plot.

(see Section 7.13, Profile Create/Update)

Subsets

A subset Part can contain node and element ranges of any model Part.

(see Section 7.12, Subset Parts Create/Update)

Vector Arrows

Vector arrows show the direction and magnitude of a vector field. Vector arrows originate from element vertices, element nodes (including mid-side nodes), or from element centers. You specify which parts are to have arrows and which vector variable to use for the arrows, as well as a scale factor. You can eliminate components of the vector, and can also filter the arrows to eliminate high, low, low/high, or banded vector arrow magnitudes. The vector arrows can be either straight or curved, and can have arrow heads. The arrow heads are either proportional to the arrow or can be of fixed size.

(see Section 7.10, Vector Arrow Create/Update)

Tensor Glyphs

Tensor glyphs show the direction of the principal eigenvectors. You specify which eigenvectors you wish to view and how you wish to view compression and tension.

(see Section 7.20, Tensor Glyph Parts Create/Update)

Vortex Cores

Vortex cores show the center of swirling flow in a flow field.

(see Section 7.15, Vortex Core Create/Update)

Shock Surfaces/ Regions

Shock surfaces or regions show the location and extent of shock waves in a 3Dflow field.

(see Section 7.16, Shock Surface/Region Create/Update)

Separation/ Attachment Lines

Separation and attachment lines show where flow abruptly leaves or returns to the 2D surface in 3D fields.

(see Section 7.17, Separation/Attachment Lines Create/Update)

Queries

In addition to visualizing information, you can make numerical queries.

You can query on information for a node, point, element, or a part.

You can query on information for a data set (such as size, no. of elements, etc.)

You can query scalar and vector information for a point or node over time.

You can query scalar and vector information along a line. The line can either be a defined line in space, or a logical line composed of multiple 1D elements for a part (for example query of a variable on a particle trace).

You can query to find the spatial or temporal mean as well as the min/max information for a variable.

Where applicable, query information can be in the form of a Fast Fourier Transform (FFT).

Plotting

The plotter plots Y vs. X curves. The user controls line style, axis control, line thickness and color. All query operations that result in multiple value output in EnSight can be sent to the plotter for display. The user can control which curves to plot. Multiple curve plots are possible. All plotable query information can be saved to a disk file for use with other plotting packages. The current maximum number of plotters at one time is 25.

(see Section 7.5, Query/Plot and Section 7.6, Interactive Probe Query)

Variable Creation

New information can be computed resulting in a constant, a scalar, or a vector. EnSight includes useful built-in functions for computing new variables:

Lambda2	Q_criteria
Area	udmf_sum
Boundary Layer Cf Edge	Boundary Layer Cf Wall
Boundary Layer Cf Wall Shear Stress	Boundary Layer Cf Wall Components
Boundary Layer Displ Thickness	Boundary Layer Scalar
Boundary Layer Momentum Thickness	Boundary Layer Dist to Value fr Wall
Boundary Layer Y1 Plus off Wall	Boundary Layer Thickness
Boundary Layer Recovery Thickness	Boundary Layer Velocity Mag Gradient
Boundary Layer Velocity at Edge	Boundary Layer Shape Parameter
Coefficient	Case Map
Complex Argument	Complex from real and imaginary
Complex Imaginary	Complex Conjugate
Complex Transient Response	Complex Modulus
Curl	Complex Real
Density, Normalized	Density
Density, Normalized Stagnation	Density, Stagnation
Distance Between Part Elements	Density, Log of Normalized
Distance Between 2 Nodes	Divergence
Element to Node	Element Size
Energy, Kinetic	Energy, Total
Enthalpy, Normalized	Enthalpy
Enthalpy, Normalized Stagnation	Enthalpy, Stagnation
Flow	Entropy
Fluid Shear Stress	Flow Rate
Force	Fluid Shear Stress Max
Gradient	Force1D
Gradient Tensor	Gradient Approximation
Helicity Density	Gradient Tensor Approximation
Helicity, Relative Filtered	Helicity, Relative
Integral, Line	Iblanking Values
Integral, Volume	Integral, Surface
Line Vectors	Length
Mach Number	Make Element Scalar
Make Nodal Scalar	Make Vector
Mass Flux Average	Mass Particle Scalar
Material Species	Material Species to Scalar
Max	Min
Moment	Moment Vector
Momentum	Node to Element

Normal	Normal Constraints
Normalize Vector	Offset Field
Offset Variable	Pressure
Pressure Coefficient	Pressure, Dynamic
Pressure, Normalized	Pressure, Log of Normalized
Pressure, Pitot	Pressure, Pitot Ratio
Pressure, Stagnation	Pressure, Normalized Stagnation
Pressure, Stagnation Coefficient	Pressure, Total
Radiograph Grid	Radiograph Mesh
Rectangular to Cylindrical Vector	Server Number
Shock Plot3d	Spatial Mean
Speed	Sonic Speed
SoS Constant	Statistical Moment
Statistical Regression	Statistical Regression 1 or 2 values
Stream Function	Swirl
Temperature	Temperature, Normalized
Temperature, Stagnation	Temperature, Normalized Stagnation
Temperature, Log of Normalized	Temporal Mean, Min, or Max
Tensor Component	Tensor Determinate
Tensor Eigenvalue	Tensor Eigenvector
Tensor Make	Tensor Make Asymmetric
Tensor Tresca	Tensor Von Mises
Velocity	Volume
Vorticity	

A calculator and built-in math functions also are useful for creating variables. Any created variable is available throughout EnSight, and is automatically recomputed if the user changes the current time (in case of transient data).

(see Section 4.3, Variable Creation)

In addition to the built-in general functions and the calculator options, variables can be derived from user written external functions called User Defined Math Functions (UDMF). The UDMF's appear in EnSight's calculator in the general function list and can be used just as any calculator function.

Another feature of EnSight facilitates the creation of boundary layer variables.

(see Section 7.18, Boundary Layer Variables Create/Update)

Transient Data

EnSight handles transient (time dependent) data, including changing connectivity for the geometry. You can easily change between time steps via the user interface. All parts that are created are updated to reflect the current display time (you can override this feature for individual parts). You can change to a defined time step, or change to a time between two defined steps (EnSight will linearly interpolate between steps), though the "continuous" option is only available for cases without changing connectivity.

Animation

You can animate your model in four ways: particle trace animation, flipbook animation, solution time streaming, and keyframe animation.

Particle Trace Animation

Particle trace animation sends "tracers" down already created particle traces. You control the color, line type, speed and length of the animated traces.

If transient data is being animated at the same time, animated traces will automatically synchronize to the transient data time, unless you specifically indicate otherwise.

Flipbook Animation

A Flipbook animation reads in transient data step by step or moves a part spatially through a series of increments and stores the animation in memory. Playback is much faster as it is from memory rather than disk, but the trade-off is that Flipbook Animation can fill up your client memory. Flipbook animation is simpler to do than keyframe animation, while allowing four common types of animation:

Sequential presentation of transient data
Mode shapes based on a nodal displacement variable
EnSight created parts with an animation delta that recreates the part at a new location (i.e., moving isosurfaces and Clip surfaces).
Sequential displacement by linear interpolation from zero to maximum vector value.

You can specify the display speed, and can step page-by-page through the animation in either direction. You can load some, or all the desired data. If you later load more data, you can choose to keep the already loaded data. With transient data, you can create pages between defined time steps, with EnSight linearly interpolating the data.

Flipbooks can be created in two formats: a) Object animation where new objects are created for each time step. The user can then manipulate the model during animation play back or b) Image animation where a bitmap of the Main View image is created and stored off for each animation page. For large models, image animation can sometimes take less memory - while trading off the capability to manipulate the model during animation.

(see Section 7.2, Flipbook Animation)

Solution Time Streaming

Solution time streaming accomplishes the same result as a flipbook animation of transient data except the data is never loaded into memory: it is streamed directly from disk. While you don't see the animation speed of a flipbook, you only need enough memory to load in one step at a time.

Keyframe Animation

Keyframe animation performs linearly interpolated transformations between specified key frames to create animation frames. Command language can be executed at key frames to script your animation. Some minimal editing is possible by deleting back to defined key frames. Animation key frames can be saved and restored from disk. Animation can be done on transient data and can automatically synchronize with simultaneous flipbook animation and particle trace animation.

"Fly-around", "rotate-objects", and "exploded-view" quick animations are predefined for easy use.

Keyframe animation can be recorded to disk files using a format of your choice.

(see Section 7.3, Keyframe Animation)

Implementation

Interface	EnSight uses the OSF/Motif graphical user interface conventions for the Unix version and Win32 conventions under the Windows 2000/XP operating system. Many aspects of the interface can be customized.
Client-Server	EnSight is a *distributed application*—it runs as separate processes that communicate with each other via a TCP/IP or similar connection. The *Server* performs most CPU-intensive and data-handling functions, while the *Client* performs the graphics-display and user-interface functions. The Client and Server can run together on one host workstation in a "stand-alone" installation or on two host systems with each hardware system performing the functions it does best. When more than one case is loaded the Client communicates with multiple Server processes.
Server-of-Servers	A special server-of-servers (SOS) can be used in place of a normal server if you have partitioned data or utilize the auto-decompose feature. This SOS acts like a normal server to the client, but starts and deals with multiple servers, each of which handle their portion of the dataset. This provides significant parallel advantage for large datasets. (see Section 11.8, Server-of-Server Casefile Format)
Virtual Reality	EnSight is fully capable of running multi-pipe display, virtual reality and distributed rendering modes. (see Section 13, Parallel and Distributed Rendering)
Command Language	Each action performed with the graphical user interface has a corresponding EnSight command. A session file is always being saved to aid in recovery from a mistake or a program crash. The user will be prompted upon restart, after a crash, whether or not to use a recovery file to restore the session. The command language is human-readable and can easily be modified. Command files can be played all the way through, or you can choose to stop the file and step through it line-by-line.
Python	For more powerful scripting, EnSight supports the Python programming language. The EnSight Python implementation includes every EnSight command as well as looping, conditionals, and a large library of standard utilities.
Context Files	You can define a "context" and apply it to similar datasets.
Graphics Hardware	Many graphics functions of EnSight are performed by your workstation's graphics hardware. EnSight uses the OpenGL graphic libraries and is available on a multitude of hardware platforms.
Parallel Computation	EnSight supports shared-memory parallel computation via POSIX threads. Threads are used to accelerate the computation of streamlines, clips, isosurfaces, and other compute-intensive operations. (See How To Setup for Parallel Computation for details on using.)
Distributed Memory Parallel Computation	EnSight supports distributed memory parallel computations (clusters) via server-of-server operations. The data decomposition may either be done by you or can be done "on the fly".
Macros	You can define macros tied to mouse buttons or keyboard keys to automate actions you frequently perform.

Saving and Archiving

You can save the entire current status of EnSight for later use, and can save other entities as well (including the geometry of created parts for use by your analysis software).

(see Section 2.6, Archive Files)

Environment Variables

You can control a number of aspects of EnSight (both client and server) with environment variables. (See How To Use Environment Variables)

Documentation

The printed EnSight documentation consists of the Installation Guide.

The on-line EnSight documentation consists of the EnSight Getting Started Manual, How To Manual, User Manual, Interface Manual, and a Command Language Reference Manual. The online documentation is available via the Help menu.

User Manual

The EnSight User Manual is organized as follows:

User Manual Table of Contents

Chapter 1 - Overview

Chapter 2 - Input/Output. This chapter describes the reading of model data (with internal or user-defined readers), command files, archive files, context files, scenario files, and various other input and output operations.

Chapter 3 - Parts. This chapter describes the various types of Parts, selection, identification, and editing of Parts, and various Part operations,

Chapter 4 - Variables. This chapter describes the selection and activation of variables, color palettes, and the creation of new variables.

Chapter 5 - GUI Overview. This chapter describes the EnSight Graphic User Interface.

Chapter 6 - Main Menu. This chapter describes the features and functions available through the buttons and pull-down menus of the Main Menu of the GUI.

Chapter 7 - Features. This chapter describes the features and functions available through the Icon buttons of the Feature Icon Bar of the GUI.

Chapter 8 - Modes. This chapter describes the features and functions available through the Icon Buttons of the Mode Icon Bar in the six different Modes.

Chapter 9 - Transformation Control. This chapter describes the Global transformation of all Frames and Parts, the transformation of selected Frames and Parts as well as selected Frames alone, the transformation of the various Tools, and the adjustment of the Z-Clip planes and the Look At and Look From Points.

Chapter 10 - Preference File Formats. This chapter describes the format of various preference files which the uses can affect.

Chapter 11 - EnSight Data Formats. This chapter describes in detail the format of the various EnSight data formats.

Chapter 12 - Utility Programs. This chapter describes a number of unsupported utility programs distributed with EnSight.

Chapter 13 - Parallel Rendering and Virtual Reality. This chapter describes how to configure EnSight for various VR configurations and for parallel rendering.

User Manual Index

Cross References in the User Manual will appear similar to:

(see Chapter __ or (see Section __

Clicking on these Cross References will automatically take you to the referenced Chapter or Section.

Command Language Reference Manual

This manual describes each command of EnSight's command language.

How To...

The various How To documents available on-line provide step-by-step, click by click instructions explaining how to perform tasks within EnSight such as creating an isosurface or reading in data.

Interface...

This manual describes the various methods and API's that exist for interfacing with EnSight.

Ordering

To order printed copies of EnSight documentation, go to our website at www.ensight.com and click on support and choose documentation and follow the instructions.

Newsletter

CEI periodically publishes an electronic EnSight newsletter. If you would like to subscribe to the newsletter, see our website:

www.ensight.com.

Contacting CEI

EnSight was created to make your work easier and more productive. If you have any questions about or problems using EnSight, or have suggestions for improvements, please contact CEI support:

Phone:	(800) 551-4448 (USA) (919) 363-0883 (Outside-USA)
Fax:	(919) 363-0833
Email:	support@ensight.com

2 Input

This chapter provides information on data input and output for EnSight.

2.1 Reader Basics provides a detailed description of the basics for reading data. This section is referenced by all formats, in that they all use some or all of these basic procedures. The quick load, as well as the more flexible two step load process is discussed for both unstructured and structured data formats.

2.2 Native EnSight Format Readers describes the specifics for reading the EnSight formats.

2.3 Other Readers describes the specifics for reading many other formats into Ensight. These can be internal or user-defined readers.

2.4 Other External Data Sources describes other ways in which model data can be prepared to be read into EnSight.

2.5 Command Files provides a description of the files that can be saved for operations such as automatic restarting, macro generation, archiving, hardcopy output, etc.

2.6 Archive Files describes options for saving and restoring the entire current state of the program.

2.7 Context Files describes the options for saving and restoring context files.

2.8 Session Files describes the options for saving and restoring session files.

2.9 Scenario Files describes the options for saving scenario files that can be displayed in the EnLiten program.

2.10 Saving Geometry and Results Within EnSight describes how to save model data, from any format which can be read into EnSight, as EnSight gold casefile format.

2.11 Saving and Restoring View States describes options for saving and restoring given view orientations.

2.12 Saving and Printing Graphic Images describes options for saving and printing graphic images.

2.13 Saving and Restoring Animation Frames describes options for saving and restoring flipbook and keyframe animation frames.

2.14 Saving Query Text Information describes options for saving query information to a text file.

2.15 Saving Your EnSight Environment describes options for saving various environment settings which affect EnSight.

Note: Formats for EnSight related files are described in chapters 10 and 11. Formats for the various Analysis codes are not described herein.

2.1 Reader Basics

Dataset Format Basics

EnSight is designed to be an engineering postprocessor, and supports data formats for popular engineering simulation codes and generally used data formats. Yet its many features can be used in other areas as well. EnSight has been used to visualize and animate results from simulations of diesel combustion, cardiovascular flow, petroleum reservoir migration, pollution dispersion, meteorological flow, as well as results from many other disciplines.

EnSight reads node and element definitions from the geometry file and groups elements into an entity called a *Part*. A Part is simply a group of nodes and elements (the Part can contain different element types) which all behave the same way within EnSight and share common display attributes (such as color, line width, etc.).

EnSight allows you to read multiple datasets and work with them individually in the same active session. Each dataset comprises a new "Case" and is handled by its own Server process and can be added by using EnSight's main menu Case > Add... option. Note: if the client and the server are each on different computers, then the data directory path is that seen from the server. Each server process has its own console window and the output from the data read is directed to this console. On Windows it is sometimes helpful to enlarge the default buffer size on the server window to accomodate the sometimes large amount of output. Right-click on the top left of the server window (named at top C:\WINDOWS\System32\cmd.exe) and choose Screen Buffer size to be Width of 120 and Height of 9999, and Window size of Width 120 and Height of 40. Then when you save it, save it for all windows of this name and every time the server window is opened it will have these defaults and to see all of the server console output.

Reading and Loading Data Basics

Reading and then Loading Data into EnSight can be done from "Simple" or "Advanced" interface.

Simple Interface The simple interface allows you to select a dataset which is read by the EnSight server and then have all parts loaded and displayed on the Client. This is quick but it does not allow control of which parts to load, nor does it allow you to control the visual representation. Also, the simple interface only works for files mapped in the `ensight_reader_extension.map` file found in the `$CEI_HOME/ensight92/`

`site_preferences` and/or in the EnSight Defaults Directory which is located

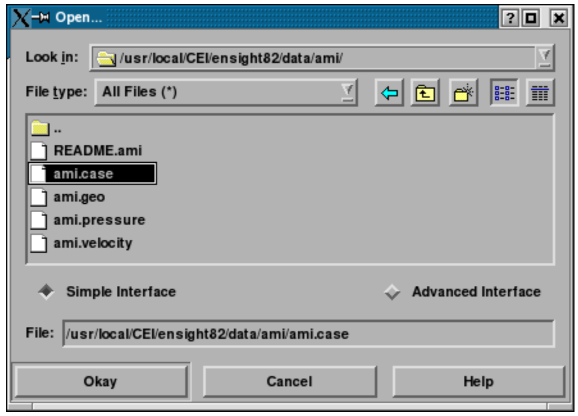


Figure 2-1
File Open Dialog - Simple Interface

at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac) directories.

Look in
This field specifies the directory (or folder) name that is used to list the files and directories in the list below.

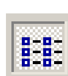
File type
Limits the directory content list to the file type chosen. The default is to show all files.

File/Directory Manipulation Buttons


Changes the Look in directory to the one previously displayed.


Changes the Look in directory to be one up from the current.


Creates a new directory/folder in the current directory. You can rename the new directory/folder by right clicking on it in the file listing and selecting Rename.


Show the content of the Look in directory in list view. In this view the directory and file names are listed in alphabetical order. This is the default.


Show the content of the Look in directory in detail view. This view will show all directories and file names in alphabetical order and also show size, type, date, and read/write attributes.

Content List
Shows the content of the Look in directory/folder. Single click to select a file. This will insert the file name with full path as described in the Look in field in to the File field. If you double click a file name, the file will be inserted into the File field and the Okay button will execute. If you double click on a directory/folder name, you will change the Look in filter.

File
Specifies the file name that will be read once the Okay button is selected. As some file formats require more than one file (geometry and results potentially) any associated files will also be read according to the `ensight_reader_extension.map` file.

Okay	Click to read the file (and associated files) specified in the File field and close the dialog.
Cancel	Click to close the Open... dialog without reading any files. (For a step-by-step tutorial please see How To Read Data).

Advanced Interface

The advanced interface allows you to select a dataset which is read by the EnSight server and then select which parts out of the dataset you wish to load and display on the Client. You can control the format option, extra user interface options that may be defined for your data file format and time settings.

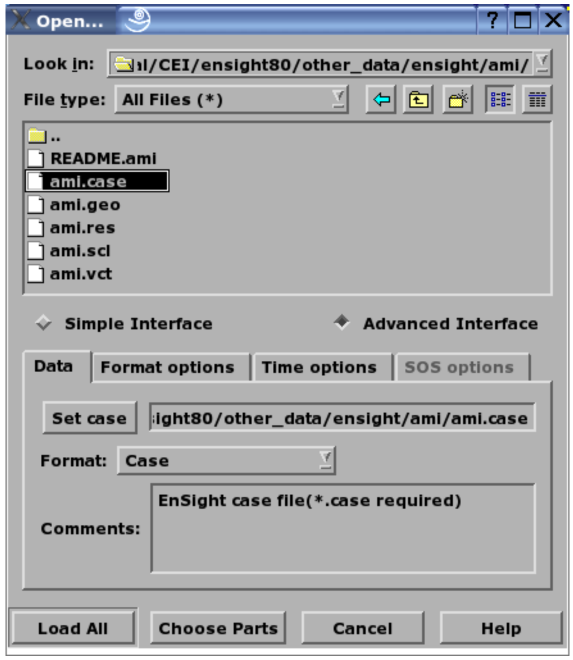


Figure 2-2
File Open Dialog - Advanced Interface - Data Tab

Look in
This field specifies the directory (or folder) name that is used to list the files and directories in the list below.

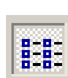
File type
Limits the directory content list to the file type chosen. The default is to show all files.

File/Directory Manipulation Buttons


Changes the Look in directory to the one previously displayed.


Changes the Look in directory to be one up from the current.


Creates a new directory/folder in the current directory. You can rename the new directory/folder by right clicking on it in the file listing and selecting Rename.


Show the content of the Look in directory in list view. In this view the directory and file names are listed in alphabetical order. This is the default.


Show the content of the Look in directory in detail view. This view will show all directories and file names in alphabetical order and also show size, type, date, and read/write attributes.

Content List
Shows the content of the Look in directory/folder. Single click to select a file. This will insert the file name with full path as described in the Look in field in to the File field. If you double click a file name, the file will be inserted into the File field and the Okay

button will execute. If you double click on a directory/folder name, you will change the Look in filter.

Data Tab
: Contains settings for file format and file names.

Set
: The name for this field will depend on the file format. For example, for EnSight it is "Set case" while for CTH it is "Set spcth*". This field describes the file name used to read the dataset. Depending on the file format, there may be two (or possibly more) Set fields. The use of the second (or third) set field depends on the file format and is described in the Comments section of the dialog.

Format
: Specifies the Format of the dataset. This pulldown will vary depending upon what readers are installed at your local site, and what readers are made visible in your preferences. Note: you can start up ensight with the -readerdbg flag to view verbose information on the readers as they are loaded into EnSight.

Comments
: Helpful information that is reader-specific will appear here, such as what file types are entered into what fields.

Format Options Tab
: Contains format specific information.

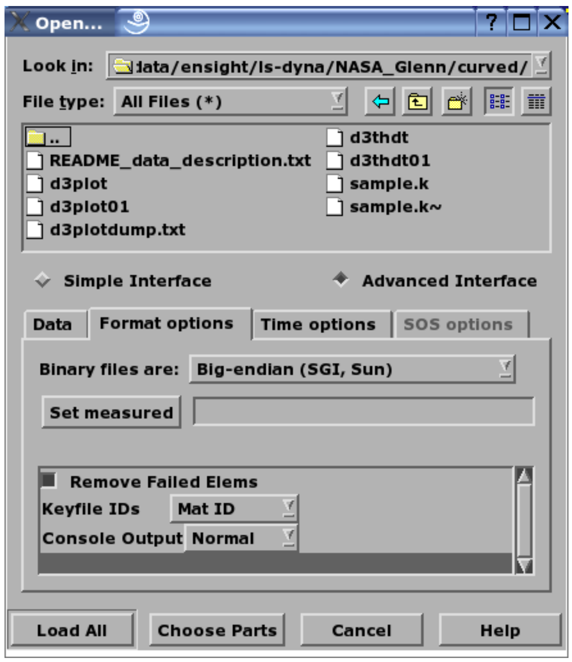


Figure 2-3

File Open Dialog - Advanced Interface - Format options Tab

Binary files are
: This is typically checked automatically by the reader, and thus usually there is no need to use this toggle. If the file is binary, sets the byte order to:
  Big-Endian - byte order used for HP, IBM, SGI, SUN, NEC, and IEEE Cray.
  Little-Endian - byte order used for Intel and alpha based machines.
  Native to Server Machine - sets the byte order to the same as the server machine.

Set measured
: Name of an EnSight 5 format measured results file (typically .mea file). Measured data is read independently of the reader and is entered here for all readers except Case file format. For Case file format, this field is not used and the measured data filename is entered into the Case file. The measured data filename is always optional. Clicking the button inserts the file name shown in Selection field and also inserts path information into Path field. File names can alternatively be typed into the field.

Set boundary
: Name of a boundary file. This field is used only for structured data in Case file, Plot3d, or Special HDF5, or other user-defined formats with structured data.

Extra GUI
: The User-Defined Reader API includes routines to add Toggles, Text Fields, and Pulldown Menus that can allow the user to modify the reader behavior. Only a subset of

the readers make use of this feature.

Time Options Tab

Contains Time specific information.

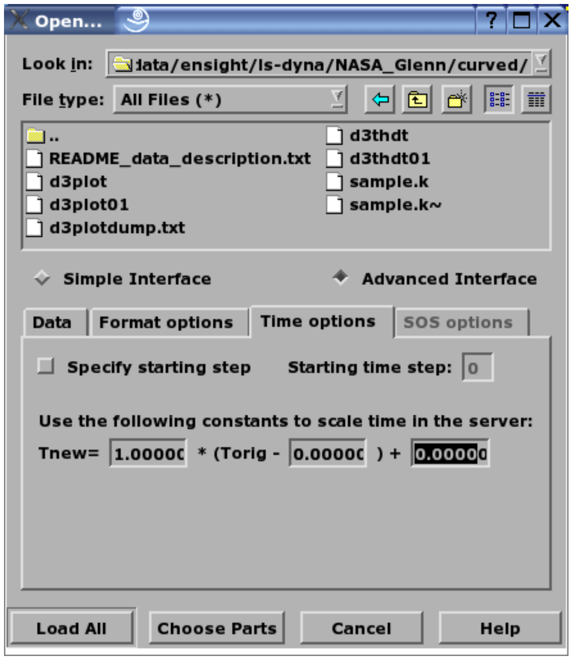


Figure 2-4

File Open Dialog - Advanced Interface - Time options Tab

Time Settings

Specify starting time step. If not specified, EnSight will load the last step (or whatever step you have set in your preferences, see Edit>Preferences>Data). This section also allows you to shift, scale and/or offset the original time values according to the values entered into the equation.

SOS Options Tab

If connected to an SOS server, this tab will be available and controls how the servers will behave when handling data as well as what resources will be used.

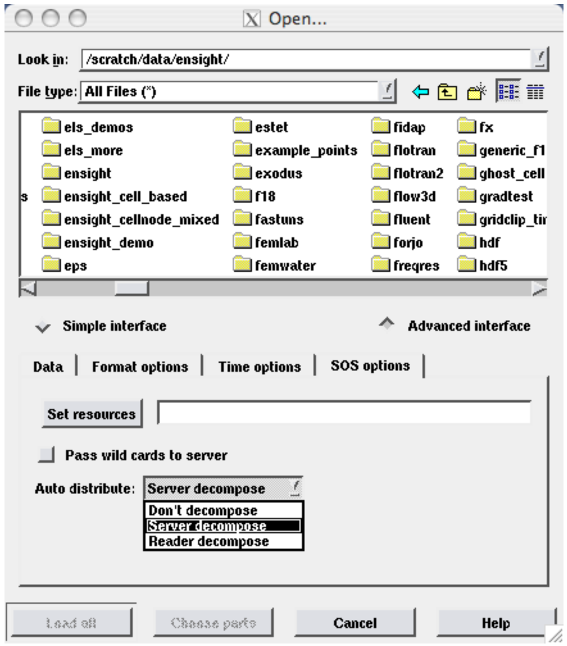


Figure 2-5
File Open Dialog - Advanced Interface - SOS options Tab

Set resources — Sets a filename to be used for SOS and Server resources.

Pass wild cards to server — This toggle will pass wildcard filenames on to the server as opposed to resolving them on the SOS. The usefulness of this toggle is entirely dependent on the specific reader in use.

Auto distribute — How to decompose and distribute the data to each of the servers.

- Don't — Data is already stored on disk decomposed.
- Server — Use the server to automatically partition the data
- Reader — Use the reader to automatically partition the data

Load All — Click to read and load all of the parts associated with the file names specified and close the dialog.

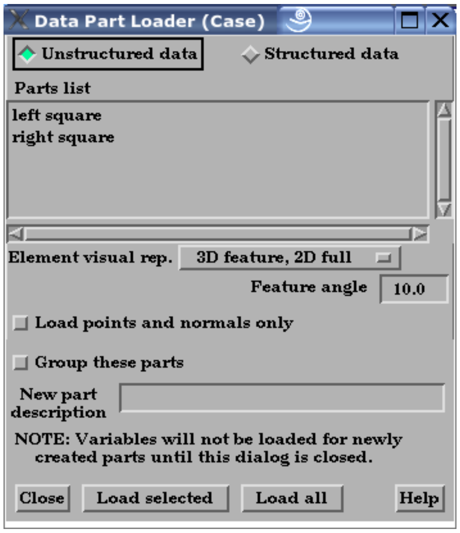
Choose Parts — Click to read the data files specified, close the dialog and bring up the Data Part Loader.

Cancel — Click to close the Open... dialog without reading any files.

(For a step-by-step tutorial please see How To Read Data).

Data Part Loader — The Data Part Loader will allow you to select the parts to be loaded on the server, as well as their new name (if desired) and representation. There are two basic part loader windows. Details of these windows will be discussed below, and variants from these windows will be discussed under each specific reader format.

Unstructured Part Loader Dialog



Structured Part Loader Dialog

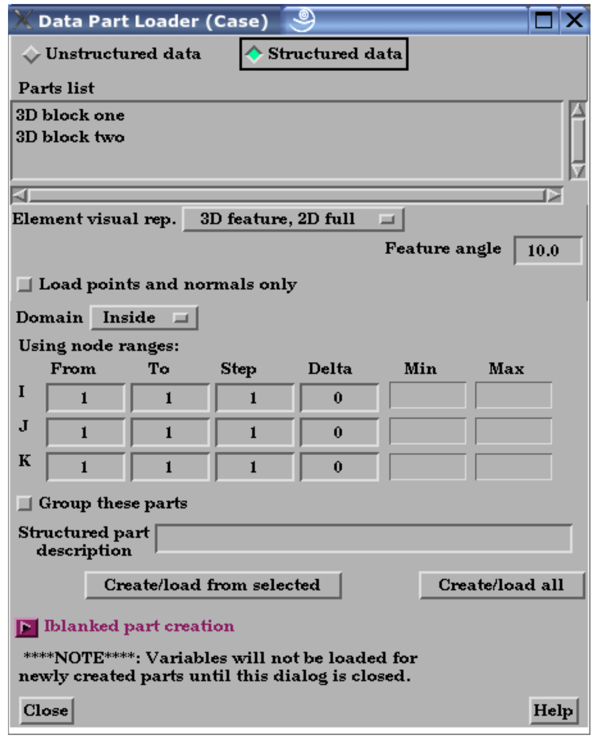


Figure 2-6
Typical File Data Part Loader dialogs

All Parts or some of those available on the Server may be loaded to the Client and their visual representation can be chosen. The Data Part Loader may be reopened at a later time and additional or duplicate parts loaded as desired.

Unstructured Data — This toggle indicates that the Part(s) listed in the Part List is(are) unstructured.

Structured Data — This toggle indicates that the Part(s) listed in the Part List is(are) structured.

Parts List — Lists all Parts in the data files which may be loaded to the Server (and subsequently to the Client). Some formats such as EnSight6 or EnSight Gold data can have unstructured, structured, or both types of Parts.

Element Visual Rep. — Parts are defined on the server as a collection of 0, 1, 2, and 3D elements. EnSight can show you all of the faces and edges of all of these elements, but this is usually a little overwhelming, thus EnSight offers several different *Visual Representations* to simplify the view in the graphics window. Note that the Visual Representation only applies to the

EnSight client—it has no affect on the data for the EnSight server.

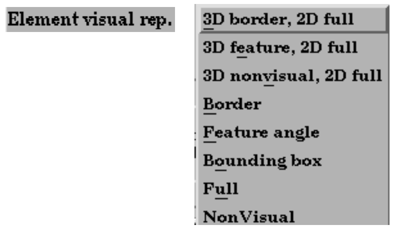


Figure 2-7

Element Visual Representation pulldown

3D Border, 2D Full — In this mode, load the designated parts, show all 1D and 2D elements, but show only the unique (non-shared) faces of 3D elements.

3D Feature, 2D Full — In this mode, load the designated parts, but show the 3D elements in Feature Angle mode (see Feature below), and show all of the 1D and 2D elements.

3D nonvisual, 2D full — In this mode load the 3D parts but do not display them in the graphics window (see Non Visual below) and load all the 1D and 2D elements.

Border — In Border mode all 1D elements will be shown. Only the unique (non-shared) edges of 2D elements and the unique (non-shared) faces of 3D elements will be shown.

Feature Angle — When EnSight is asked to display a Part in this mode it first calculates the 3D Border, 2D Full representation to create a list of 1D and 2D elements. Next it looks at the angle between neighboring 2D elements. If the angle is above the Angle value specified in the Feature Angle Field, the shared edge between the two elements is retained, otherwise it is removed. Only 1D elements remain on the EnSight client after this operation.

Bounding Box — All Part elements are replaced with a bounding box surrounding the Cartesian extent of the elements of the Part.

Full — In Full Representation mode all 1D and 2D elements will be shown. In addition, all faces of all 3D elements will be shown.

Non Visual — This specifies that the loaded Part will not be visible in the Graphics Window because it is only loaded on the Server. Visibility can be turned on later by changing the representation (at which time the elements of the selected representation will be sent to the client).

Load Points and normals only — If toggled on, only the vertices of the element representation, with normals, will be loaded to the client.

Group these parts — If more than one part is selected, they can be grouped into a single entity. The name of the group will be according to the New Part Description filed and the individual parts will receive the names shown in the part list.

New Part Description — This allows the user to name the part. If nothing is entered here, then the part is named from the partlist.

Load Selected — Loads Parts selected in the Parts List to the EnSight Server. The Parts are subsequently loaded to the EnSight Client using the specified Visual Representation. If Non Visual is specified, the selected Parts will be loaded to the Server, but not to the Client.

Load All — Loads all Parts in the Parts List to the EnSight Server. The Parts are subsequently loaded to the EnSight Client using the specified Visual Representation. If Non Visual is specified, the selected Parts will be loaded to the Server, but not to the Client.

Structured Data

Domain

Specifies the general iblanking option to use when creating a structured Part. If the model does not have iblanking, InSide will be specified by default.

Inside	Iblank value = 1 region
Outside	Iblank value = 0 region
All	Ignore iblanking and accept all nodes

Using Node Ranges:

From IJK

Specifies the beginning I,J,K values to use when extracting the structured Part, or a portion of it. Must be >= Min value.

To IJK

Specifies the ending I,J,K values to use when extracting the structured Part, or a portion of it. Must be <= Max value.

Valid values for the From and To fields can be positive or negative. Positive numbers are the natural 1 through Max values. Negative values indicate surfaces back from the max, so -1 would be the max surface, -2 the next to last surface etc. There are therefore two ways to indicate any of the range values; the positive number from the min towards the max, or the negative number from the max toward the min. The negative method is provided for ease of use because of varying max values per part. (Zero will be treated like a -1, thus it is another way to get the max surface)

```
  1, 2, 3,... --->                <--- ...-3, -2 ,-1
   |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
 min                                          max
(always 1)                              (varies per zone)
```

Step IJK

Specifies the step increment through I,J,K. A Step value of 1 extracts all original data. A Step value of 2 extracts every other node, etc. Thus step values greater than 1 give a coarser resolution.

Delta IJK

Specifies the delta to use when creating more than one surface from the same ijk part. Only one of the directions may be non-zero. *Note that an unstructured part is the result of any non-zero delta values.*

Min IJK

Minimum I,J,K values for Part chosen (for reference).

Max IJK

Maximum I,J,K values for Part chosen (for reference).

Part Description

Text field into which you can enter a description for the Part. If left blank a default will be used.

Create And Load from selected

Extracts the data from the data files and creates a Part on the Server (and on the Client unless NonVisual has been specified for Representation) based on all information specified in the dialog.

If only one part is highlighted, the values shown in the From and To fields (as well as the Min and Max fields) are the actual values for the selected part. Using the From and To fields you can control whether an EnSight part will be created using the entire ijk ranges or some subset of them. The Step field allows you to sample at a more coarse resolution. And the Delta field allows for multiple “surfaces” in a given part (like blade rows of a jet engine). Please note that use of a non-zero delta produces an unstructured part instead of a structured one.

If more than one Part is highlighted, the values shown in the From and To fields are the combined bounding maximums of the selected parts. The same basic functionality described for a single part selection applies for multiple part selection, with one part being created for each selected part in the dialog. If the specified ranges for the multiple selection exceed the bounds of a given part, they are modified for that part so that its bounds are not exceeded.

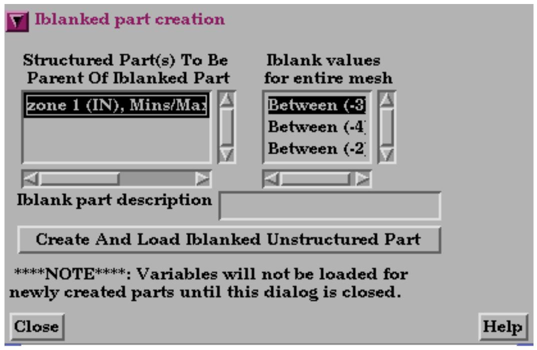


Figure 2-8
Iblanked Part Creation Section of Structured Part Loader dialog

You use this portion of the Part Loader dialog to further extract iblanked regions from structured parts which were created either as inside, outside, or all portions of the model.

Structured Part(s) To Be Parent of Iblanked Part — Lists all structured parts that have been created thus far in the dialog above.

Iblank Values For Entire Mesh — Lists all possible iblank values found in the model. This is a global list and may not apply to all parts.

Iblank Part description — Text field into which you can enter a part name for the iblanked part.

Create And Load Iblanked Unstructured Part — Extracts a new iblanked part from an existing structured part. This new part will actually be an unstructured part.

(For step-by-step instructions see How To Read Data)

Loading Tips

For large datasets, you should try to reduce the amount of information that is being processed in order to minimize required memory. Here are some suggestions:

- When writing out data from your analysis software, consider what information will actually be required for postprocessing. Any filtering operation you can do at this step can greatly reduces the amount of time it takes to perform the postprocessing.
- For each Part you do load to the Client, a *representation* must be chosen. This visual representation can be made very simple (through the use of the Bounding Box or Feature Angle option, for example), or can be made more complex (by using the border or full elements). The more you can reduce the visual representation, the faster the graphics processing will occur on the Client (see Node, Element, and Line Attributes in (see Section 3.3, Part Editing).
- Load to the Client only those Parts that you need to see. For example, if you were postprocessing the air flow around an aircraft you would normally not need to see the flow field itself and could load it non-visual, but you would like to see the aircraft surface and Parts created based on the flow field (which remains available on the Server).
- If you have multiple variables in your result file, activate only those variables you want to work with. When you finish using a variable, consider deactivating it to free up memory and thereby speed processing (see Section 4.1, Variable Selection and Activation).

Troubleshooting Loading Data

Problem	Probable Causes	Solutions
Data loads slowly	Loading more Parts than needed	For some models, especially external fluid flow cases, there is a flow field Part which does not need to be visualized. Try loading this Part non-visual.
	Too many elements	Make sure the default element representation for Model Parts is set to 3D Border/2D Full before loading the data. In some cases it is helpful to set the representation to Feature Angle or 3D Feature 2D Full, before loading.
	Client is swapping because it does not have enough memory to hold all the Parts specified.	Try loading fewer Parts or installing more memory to handle the dataset size.
	Server is swapping because it does not have enough memory to hold all of the Parts contained in the dataset.	Install more memory in your Server host system, reduce the number of variables activated, or somehow reduce the geometry's size. (If you can get the data in, you can cut away any area not now needed. What is left can then be saved as a geometric entity and that new dataset used for future postprocessing.)
Error reading data	Incorrect path or filename	Reenter the correct information. Remember, the Path is on the server.
	Incorrect file permissions	Change the permissions of the relevant directories and files to be readable by you.
	Temporary file space is full	Temporary files are written to the default temporary directory or the directory specified by the environment variable TMPDIR for both the Client and Server. Check file space by using the command "df" and remove unnecessary files from the temporary directory or other full file systems.
	Format of the data is incorrect	Recheck the data against the data format definition. (Can use ens_checker for Ensight6 or EnSight Gold format checking.)

Problem	Probable Causes	Solutions
EnSight format scalar (or vector) data loads, but appears incorrect. Often range of values off by some orders or magnitude.	Scalar (or vector) information not formatted properly in data file	Format the file according to examples listed under EnSight Variable Files (see Chapter 11, EnSight Data Formats) (Can use ens_checker for Ensight6 or EnSight Gold format checking.)
	Extra white space appended to one or more of the records	Check for and remove any extra white space appended to each record

2.2 Native EnSight Format Readers

EnSight's native data format is useful as a general data format for finite elements or structured data. EnSight has three native data formats (from oldest to newest, EnSight5, EnSight6 and EnSight Case Gold) which are well defined and well documented so that they can be easily interfaced to your analysis code. EnSight 5, which is now considered a legacy format, used a global coordinate array and supported unstructured meshes only. EnSight 6 format again used a global coordinate array but added support for structured meshes. EnSight Case Gold (often just called Case format) is the most recent (and recommended) format. Case Gold defines geometry on a part by part basis and uses element index for connectivity. Case Gold format is tuned to the EnSight internal data structure and is the fastest and most memory efficient format available for EnSight.

A dramatic speed up in performance can sometimes be realized simply by reading in data in the given format and saving it back out as Case Gold, then re-reading the data back in using the native Case Gold reader. However, a number of solvers now output data directly into the well-documented Case format. (see Chapter 11, EnSight Data Formats). The application `ens_checker` is included with EnSight to enable error checking of the Case and EnSight 6 formats output by third-party software.

Described below is the process for reading the latest (Case & EnSight 6) and the legacy (EnSight 5) native formats:

EnSight Case Reader

EnSight5 Reader

EnSight Case Reader

In order to use this reader, you must be familiar with the basic data reader and part loader dialogs discussed previously (see Chapter 2.1, Reader Basics).

EnSight6 and EnSight Gold are input using the exact same process. The data consists of the following files:

- Case file (required)
- Geometry file (required)
- Variable files (optional)
- Measured/Particle files (optional)
  - Measured/Particle geometry files
  - Measured/Particle variable files
- Rigid body file (optional)

The Case file is a small ASCII file which points to all other files which pertain to the model. The Case file names the geometry and variable files and records time information. The geometry file is a general finite-element format describing nodes and Parts, each Part being a collection of elements, and/or structured ijk blocks. The variable file contains scalar (one value), vector (three values) or tensor (six or 9 values) data at each node and/or element. Measured/Particle files contain data about discrete Particles in space from the simulation code or information directly from experimental data.

EnSight data is based on Parts. The Parts defined in the data are always available on the Server. However, all Parts do not have to be loaded to the Client for display. Large flow fields for CFD problems, for example, are needed for computation by the Server, but can be loaded non-visual.

EnSight data can be transient. The geometry as well as the variables can change with each timestep. The casefile contains the filenames or filename patterns for the transient data.

Simple Interface Data Load

Load your casefile (typically named with a suffix `.case`) using the Simple Interface method.

Advanced Interface Data Load

Load your casefile (typically named with a suffix `.case`) using the Advanced Interface method.

Data Tab	
Format	Use the **Case** format to read EnSight6 or EnSight Gold data.
Set Case	Select the casefile (typically **.case**) and click this button

(see How To Read Data)

EnSight5 Reader

EnSight5 input data consists of the following files:

- Geometry file (required)
- Result file (optional)
- Variable files (optional)
- Measured Particle Files (optional)
  - Measured/Particle geometry files
  - Measured/Particle results files
  - Measured/Particle variable files

The geometry file is a general finite-element format describing nodes and Parts, each Part being a collection of elements. The result file is a small ASCII file allowing the user to name variables and provide time information. The result file points to variable files which contain the scalar or vector information for each node. Measured/Particle files contain data about discrete Particles in space from the simulation code or information directly from actual experimental tests.

EnSight5 data is based on Parts. The Parts defined in the data are always available on the Server. However, all Parts do not have to be loaded to the Client for display. Large flow fields for CFD problems, for example, are needed for computation by the Server, but do not generally need to be seen graphically.

EnSight5 data can have changing geometry, in which case the changing geometry file names pattern is contained in the results file. However, it is still necessary to specify an initial geometry file name in the (Set) Geometry field.

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.geo`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files (typically named with a suffix `.geo and .res`) using the Advanced Interface method.

Data Tab	
Format	Use the **EnSight 5** format.
Set geometry	Select the geometry file (typically **.geo**) and click this button
Set results	Select the results file (typically **.res**), and click this button.
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

2.3 Other Readers

EnSight includes a number of readers for non-native (non-EnSight) formats. This section includes a description of each of these included readers and includes instruction for their use. Some of the included readers are custom, internal readers, and some of them are written using the standard, User Defined Reader interface.

User Defined Reader Description

A user defined reader capability is included in EnSight which allows otherwise unsupported structured or unstructured data to be read. In other words, the user can create their own data readers. Each user defined reader utilizes a dynamic shared library produced by the user. Once produced, these readers show up in the list of data formats in the File Open Dialog just like the included readers.

User Defined Reader Implementation

The readers are produced by creating the routines documented in the user-defined API. Three versions of the user defined API are available The 1.0 API (which has been available since EnSight version 6) was designed to be friendly to those producing it, but requires more manipulation internally by EnSight and accordingly requires more memory and processing time. The 2.0 API (starting with EnSight 7.2) was designed with efficiency in mind. It requires that all data be provided on a part basis, and as such lends itself closely to the EnSight Gold type format. A few of the advantages of the 2.0 API (Now at version 2.08) are:

* Less memory, more efficient, and faster - as indicated above.
* Model extents can be provided directly, such that EnSight need not read all the coordinate data at load time.
* Tensor and complex variables are supported
* Exit routine provided, for cleanup operations at close of EnSight.
* Geometry and variables can be provided on different time lines (timesets).
* If your data format already provides boundary shell information, you can use it instead of the “border” representation that EnSight would compute.
* Ghost cells (for both structured and unstructured data) are supported
* User specified node and/or element ids for structured parts are supported
* Material handling is supported
* Nsided and Nfaced elements are supported
* Structured ranges can be specified
* Failed elements is supported
* Material Species is supported
* Rigid Body values can be supplied from the reader.
* Reader can be allowed to deal with block min, max, and stride within itself - instead of having EnSight deal with it.

A 3.0 reader API is available in EnSight 9. The 3.0 API aims to provide the flexibility of both of the previous versions while simplifying the reader development processes. Contact CEI for more information on this API.

Creating Your Own Custom User Defined Reader

The process for creating and using a user-defined reader is explained in detail in the Interface Manual. Samples, source code, makefiles, etc can be found in the following location and it's subdirectories:

On the CD: /CDROM/ensight92/src/readers

In installation
directory: $CEI_HOME/ensight92/src/readers

Start EnSight (or EnSight server) with the command line option (-readerdbg), for a step-by-step echo of reader loading progress.

```
ensight92 -readerdbg
```

The actual working user defined readers included in the EnSight distribution may vary by hardware platform.

ABAQUS_FIL Reader

The ABAQUS .fil reader is available to EnSight users, but the preferred method of loading ABAQUS data is to use the .odb reader (see Section , ABAQUS_ODB Reader).

ABAQUS_FIL input data consists of the one file with a .fil extension:

- Geometry/Results file (required). This file (the ABAQUS .fil file) contains both the geometry and any requested results. It can be either ASCII or binary.
- EnSight5 Measured/Particle Files (optional). The measured .res file references the measured geometry and variable files.

EnSight will read ASCII or binary .fil files directly. The element sets in the .fil file will be used for creating parts.

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.fil`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/result file (typically named with a suffix `.fil`) using the Advanced Interface method.

Data Tab	
Format	Use the **ABAQUS fil** format.
Set results	Select the geometry/results file (typically **.fil**), and click this button.
Format Options Tab	
Set measured	Select the measured file and click this button.

Data Part Loader

The Part Loader for this reader is simply an informational message with Okay and Cancel buttons, as shown below. There is no way to load part by part in this reader.

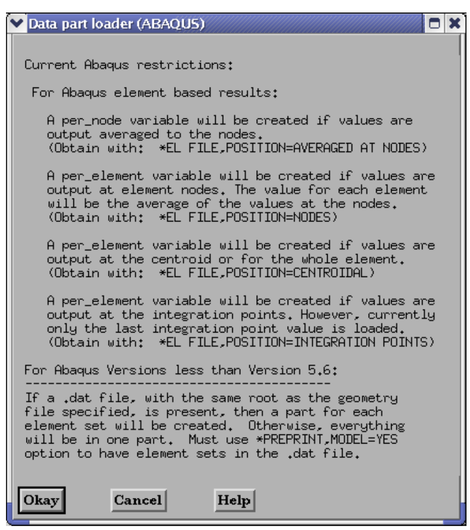


Figure 2-9
ABAQUS FIL Reader Part Loader dialog

ABAQUS_ODB Reader

Overview

Because the reader is dependent upon the ABAQUS libraries, this reader is only available for platforms supported by ABAQUS. See their website for more details.

For updated information please see the file in the following directory:

`$CEI_HOME/ensight92/src/readers/abaqus/README_67.txt` or `_68.txt`

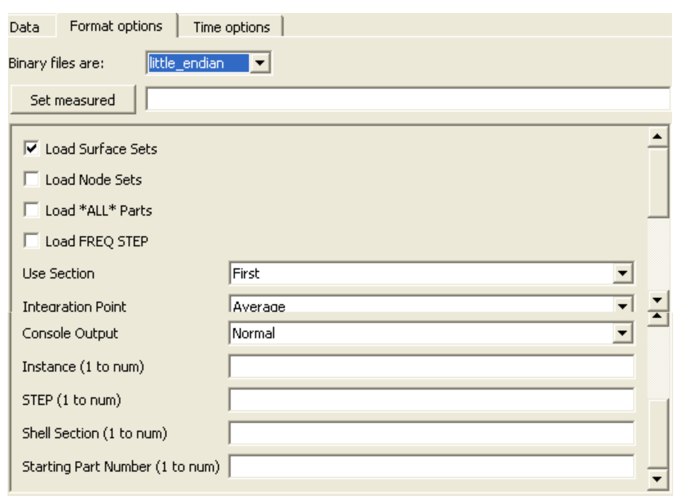
The ABAQUS odb reader is the recommended method of importing ABAQUS data into EnSight.

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.odb`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/result files (typically named with a suffix `.odb`) using the Advanced Interface method.

Data Tab	
Format	Use the **ABAQUS_ODB** format.
Set geometry	Select the geometry file (typically **.odb**) and click this button
Set results	Not used
Format Options Tab	
Set measured	Select the measured file and click this button.
Reader GUI	User controls as shown below are available: 
Load Surface Sets	Toggle ON (default) to load all Surface Sets
Load Node Sets	Toggle ON to load all Node Sets (default OFF).

Load "All" Sets	Often, ABAQUS parts that are simply the global element matrix are redundant (e.g. E_ALL contains all elements). Toggle OFF (default) to skip loading Parts with "all" in their name, saving memory and time.
Load Freq Step	Often, ABAQUS will include multiple steps in an ODB file and the one desired is the modal analysis. Toggle this ON to skip all other steps loading only the frequency step (Default is OFF). If multiple frequencies, then each EnSight timestep now becomes a different frequency.
Load Element Faces	Toggle ON to convert Surface Sets with 3D elements with Face Sets into 2D elements using face specifications (default OFF).
Use Section	Shell elements include variable data for each section. This reader allows the use of the First (default) or the Last section.
Integration Point	Shell elements include variable data for each of several integration points. Choose max, min or average (default).
Console Output	Normal - Informational and error output to the console. Verbose - Detailed informational and error output to the console. Debug - Step by step progress through the reader with detail numerical output for results to the console. None - No console output
Instance	Choose an instance (1 to the number of instances). Default is to load them all.
Step	Choose a step (1 to number of steps). Default is to load all non-frequency steps.
Shell Section	Which shell section to use (1 to number of shell sections).
Starting Part Number	Which part to begin loading (1 to number of parts). Default is 1 (load all parts).

(see How To Read Data)

AIRPAK/ICEPAK Reader

Overview

The current FLUENT Direct Reader also reads AIRPAK and ICEPAK data. The Fluent Direct reader typically loads a Fluent Case (.cas) file and the matching data (.dat) file. However, AIRPAK/ICEPAK writes out a .fdat file which doesn't automatically get recognized by the EnSight Fluent reader and some extra understanding (and sometimes user-intervention) is necessary as described below.

See the following files for latest information on the Fluent reader.

`$CEI_HOME/ensight92/src/readers/fluent/README.txt`

The comments that follow are for the current Fluent reader. The reader loads ASCII, binary single precision, or binary double precision. The files can be uncomrpessed or compressed using gzip.

Data file description

ICEPAK can generate files: `filename.cas`, `filename.dat` just like Fluent, but also if the analysis uses a nonconformal mesh (not available in Fluent) then there will be `filename.fdat`, and `filename.nc.cas` files. The `filename.nc.cas` is a nonconformal mesh geometry and it's matching results file is the `filename.fdat` file.

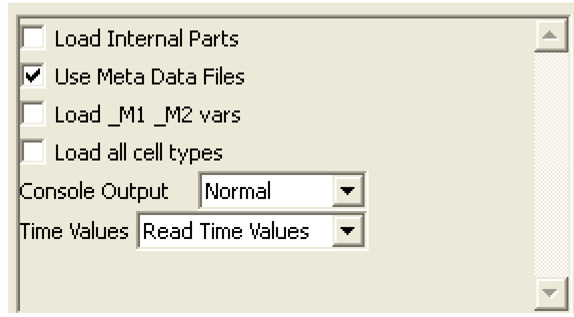
Simple Interface Data Load

Load your geometry file (typically named with a suffix `.cas`) using the Simple Interface method. EnSight will automatically load the matching `.dat` file. However, if you want to load the `filename.nc.cas` data file and it's corresponding `filename.fdat` file, then you will either need to rename it to match exactly and have the .dat extension (`filename.nc.dat`), or go to the advanced data load.

Advanced Interface Data Load.

In EnSight if you don't want to rename your files, then switch to the Advanced Interface toggle in the data reader dialog and manually choose the *.nc.cas and the *.fdat files as described below.

Data Tab	
Format	To use this reader, select the **Fluent** format.
Set cas	Select the geometry file (typically **.cas** or **.cas.gz**) and click this button. For transient data, use *.cas or *.cas.gz .
Set dat	Note that the Fluent reader will automatically select the matching .dat file. If you want to use the .fdat file, then select the results file (typically **.fdat** or **.fdat.gz**), and click this Set Dat button. For transient data, use *.fdat or *.fdat.gz .
Format Options Tab	
Set measured	Select the measured file and click this button.

Other Options using the current Fluent reader	
Load Internal Parts	Toggle this ON to load the Fluent Internal Parts. This will show all the the internal walls forming all the cell volumes. If you do toggle this on, then it is recommended that you click on the 'Choose Parts' button at the bottom of the data reader dialog, rather than 'Load all', as you'll only want to load the interior parts of interest to save memory and time. Default is OFF.
Use Meta Files	Meta files are small summary files that contain highlights of the important locations inside each of the Fluent files. Allowing the EnSight reader to write out Meta Files that map the locations of important data can provide a significant speed up the next time you access that timestep. It is recommended that you leave this toggle ON. If you have write permission in the directory where your data is located, three types of binary Meta Files will be written when you first access each file, with extensions .EFC for the cas file, .EFD for the .dat file and .EFG for the time-history data. They are optional, and if you don't have write permission, the reader will take the extra time to read the entire .CAS and .DAT file to find the relevant data each time you come back to that timestep.
Load _M1 _M2 vars	Variables that end in '_M1' and '_M2' occur in Fluent unsteady flow. They represent the value of the variable at the prior iteration time and the time prior to that respectively. By default this toggle is OFF and these variables are not loaded. Toggle this ON to load these variables.
Load all cell types	Fluent cells have a boundary condition flag. By default (toggle OFF) EnSight loads only the cells with a boundary condition flag equal to 1 (one). Toggle this option ON to load all cells with a non-zero boundary condition. For example, if you have a part with cells of boundary condition 32 (inactive), EnSight will, by default not load this part. Toggle this option ON and EnSight will load this part. Note: parts containing cells with a boundary condition of zero are never loaded.

Console Output	Use this flag to determine the amount of output to the console. Normal - Usually only echo errors to console. Verbose - Normal output plus an echo of every Fluent part that is in the dataset, whether it is interior or not, whether it is skipped, what variables are defined for which parts, and to echo it's Ensight Part number. Debug - Verbose output plus more detailed output and progress through the reader routines often valuable for understanding and reporting problems.
Time Values	Default is 'Calc Const Delta', to read a delta time from one file and calculate the time values from that. If you choose 'Read Time Values' then the reader will open each file and find the exact time value. This will be stored in the EFG file if you've not disabled Meta Files. Finally, the simplest is to 'Use File Steps' which will just use the file step number as the time value. This is quick, but is not a good idea if you need real time for anything such as particle tracing.

(see How To Read Data)

Medina BIF-BOF PERMAS Reader

Overview

Currently this reader is supported only for the following architectures: UNIX 64-bit, Linux 32-bit, IBM 5.3 64-bit, and Win32.

This reader reads in an ASCII configuration file with a .konfig suffix, which lists the geometry and results filenames.

To use this reader, you need to set the HDMcat environment variable to point to the dscat.ds file as described in the README.

`$CEI_HOME/ensight92/src/readers/medina_bifbof/README_single_part.txt`

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.konfig`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/result files (typically named with a suffix `.konfig`) using the Advanced Interface method.

Data Tab	
Format	Use the **Medina bif/bof** format.
Set file	Select the geometry/results file (typically **.konfig**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

AcuSolve Reader

Overview

Description

This reader from AcuSim will read results from Acusolve. Select the .log file from the simple or advanced interface.

Data Reader

`Main Menu > File > Data (reader)...`

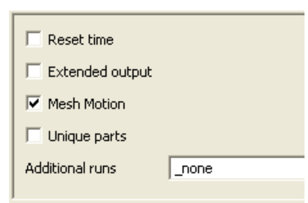
The File Selection dialog is used to specify which files you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Simple Interface Data Load

Load your AcuSolve .log file using the Simple Interface method.

Advanced Interface Data Load

Load your AcuSolve .log file using the Advanced Interface method.

Data Tab	
Format	Use the **AcuSolve** format.
Set file	This field contains the first file name. For the first file you should choose a file with extension .log. Clicking button inserts file name shown into the field. Loading the .log file will load all both geometry and results.
Format Options Tab	
Set measured	Select the measured file and click this button.
Other Options	
Reset time	When toggle is on, time begins at 0.0 (default is off).
Extended output	When toggle is on, console output will be verbose (default is off).
Mesh Motion	When toggle is on, moving meshes are visible (default is on).
Unique parts	When toggle is on, a unique set of surfaces is shown in the part list (default is off).
Additional runs	Enter the comma separated list of runs (3, 5, 10, 10:20:2), or _all. (default is _none).

Note: there is an older AcuSolve (v10 api) reader available from AcuSim.

(see How To Read Data)

ANSYS Reader

Overview

Four Ansys Readers

There are four ANSYS readers available in EnSight: three older, unsupported legacy readers and the supported Ansys Results. Long-term, Ansys Results is the reader of choice. This reader should read the latest Ansys results as well as older versions. The other three, legacy readers will not show up in the reader list by default and will not be documented in this manual.

Legacy Reader Visibility Flag

The older readers, by default, are not loaded into the list of available readers, and are not discussed in the remainder of this document. In the unlikely event you need to enable these readers, go into the Menu, Edit > Preferences and click on Data and toggle on the reader visibility flag. The legacy reader documentation is found in `$CEI_HOME/ensight92/src/readers/ansys/README` and is not included here.

Ansys Results Reader

The Ansys Results reader supports scalar, vector and tensor variables, including the capability to compute several common scalar variables derived from tensors (such as the common failure therories) as well as local element result components (such as axial stress in truss elements) when such element results are available. Additionally, there is some control over the creation of variables from element-based results. For example, they can be averaged to the nodes (with or without geometry weighting) if desired. See the format options below for more details.

Results are always presented in the global coordinate system. Thus, any results in local coordinate systems, or in non-cartesian coordinate systems are transformed as needed into the model system.

For shell elements that have multiple layers (sections), the user can choose the section that will be used. Additionally, the user can choose to have a different variable be created for each section. See format options below for more details.

The user can control how parts are created. Parts can be created according to the part id, the property id, or the material id.

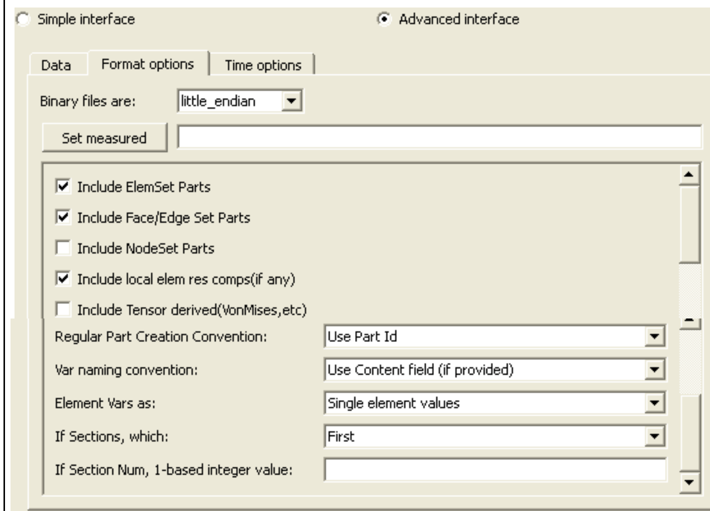
Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.rst`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/result files (typically named with a suffix `.rst`) using the Advanced Interface method.

Data Tab	
Format	Use the **Ansys Results** format.
Set file (or results)	Select the geometry/results file (typically **.rst**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

Other Options	
Include ElemSet Parts	Include any Element sets defined. These are sets of full elements which are generally some logical subset of the total number of elements. Default is on.
Include Face/ Edge Parts	Include any Face or Edge sets defined. These are some logical set of particular faces and/or edges of full elements. Default is on.
Include NodeSet Parts	Include any Node sets defined. These are generally the subset of nodes needed for the Element, Face, or Edge sets above. As such, they are generally not needed as separate parts, but can be created if desired. Default is off.
Include local elem res comps (if any)	Include the local stresses components, etc that are in the element's local system. A simple example is a bar (such as a truss element), which only has tension or compression in the element's axial orientation. Such an element would have an axial stress variable. Other elements would have appropriate result component variables. Default is on

Include Tensor derived (VonMises, etc.)	For tensor results, calculate scalars from the following derived results (principal stress/strains, and common failure theories): Mean VonMises Octahedral Intensity Max Shear Equal Direct Min Principal Mid Principal Max Principal By default, all 9 of these will be derived. You can control which are created by this toggle, with an environment variable. Namely, `setenv ENSIGHT_VKI_DERIVED_FROM_TENSOR_FLAG n` where n = 1for Mean only 2 for VonMises only 4 for Octahedral only 8 for Intensity only 16 for Max Shear only 32 for Equal Direct only 64 for Min Principal only 128 for Mid Principal only 256 for Max Principal only 512 for all or any legal combination. example: for VonMises and Max Shear only, use 18. Default is off
Regular Part Creation Convention	Parts will be created according to the following: Use Part Id - Part Id (this is the default) Use Property Id - Property Id Use Material Id - Material Id
Var naming convention	Use Content Field (if provided) - Variable names will be what is in the Content field, if provided. If not provided, they will be the VKI dataset name. This is the default. Use VKI dataset nameVariable names will be the VKI variable dataset name (which are reasonably descriptive).

Element Vars as	Single element values - Element results (whether centroidal or element nodal) will be presented as a single value per element. Thus will be per_elem variables in EnSight.This is the default. Averaged to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes without using geometry weighting. Thus will be per_node variables in EnSight. Geom weighted average to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes using geometry weighting. Thus will be per_node variables in EnSight
If Sections, which:	Which section will be used to create the variable First - The first section will be used (this is the default) Last - The last section will be used Section Num (below) - The section number entered in the field below will be used Separate Vars per Section - A separate variable will be created for each section.
Section Num	If the previous option is chosen to be Section Num, then the value in this field is the 1-based section number to use to create the variable.

(see How To Read Data)

AUTODYN Reader

Overview

Description

Reads a series of .adres files as a transient solution. Simply select one of the .adres files and the sequence will be detected. Requires that the .adres_base files exists in the same directory.

Data Reader

`Main Menu > File > Data (reader)...`

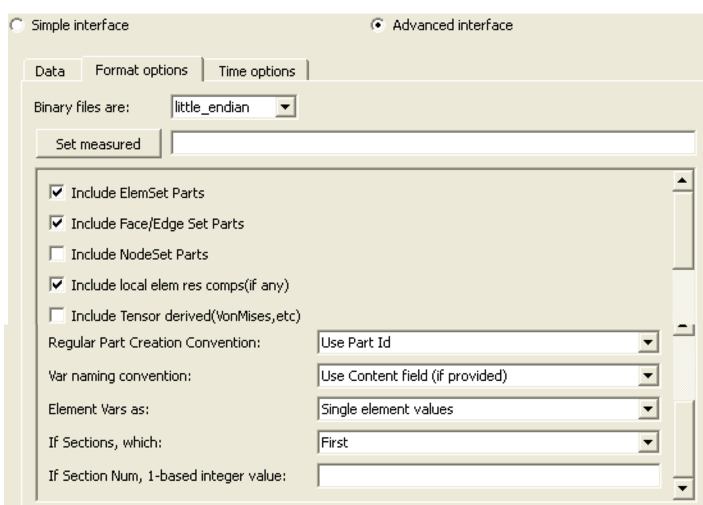
The File Selection dialog is used to specify which files you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Simple Interface Data Load

Load your AUTODYN .adres file using the Simple Interface method.

Advanced Interface Data Load

Load your AUTODYN .adres file using the Advanced Interface method.

<table>
<tr><th colspan="2">Data Tab</th></tr>
<tr><td>Format</td><td>Use the Autodyn format.</td></tr>
<tr><td>Set file</td><td>This field contains the first file name. For the first file you should choose a file with extension .adres. Clicking button inserts file name shown into the field. Loading any .adres file will load all .adres files in the directory which includes both geometry and results.</td></tr>
<tr><th colspan="2">Format Options Tab</th></tr>
<tr><td>Set measured</td><td>Select the measured file and click this button.</td></tr>
<tr><td>Other Options</td><td>
</td></tr>
<tr><td></td><td></td></tr>
</table>

Include ElemSet Parts	Include any Element sets defined. These are sets of full elements which are generally some logical subset of the total number of elements. Default is on.
Include Face/ Edge Parts	Include any Face or Edge sets defined. These are some logical set of particular faces and/or edges of full elements. Default is on.
Include NodeSet Parts	Include any Node sets defined. These are generally the subset of nodes needed for the Element, Face, or Edge sets above. As such, they are generally not needed as separate parts, but can be created if desired. Default is off.
Include local elem res comps (if any)	Include the local stresses components, etc that are in the element's local system. A simple example is a bar (such as a truss element), which only has tension or compression in the element's axial orientation. Such an element would have an axial stress variable. Other elements would have appropriate result component variables. Default is on

Include Tensor derived (VonMises, etc.)	For tensor results, calculate scalars from the following derived results (principal stress/strains, and common failure theories): Mean VonMises Octahedral Intensity Max Shear Equal Direct Min Principal Mid Principal Max Principal By default, all 9 of these will be derived. You can control which are created by this toggle, with an environment variable. Namely, `setenv ENSIGHT_VKI_DERIVED_FROM_TENSOR_FLAG n` where n = 1for Mean only 2 for VonMises only 4 for Octahedral only 8 for Intensity only 16 for Max Shear only 32 for Equal Direct only 64 for Min Principal only 128 for Mid Principal only 256 for Max Principal only 512 for all or any legal combination. example: for VonMises and Max Shear only, use 18. Default is off
Regular Part Creation Convention	Parts will be created according to the following: Use Part Id - Part Id (this is the default) Use Property Id - Property Id Use Material Id - Material Id
Var naming convention	Use Content Field (if provided) - Variable names will be what is in the Content field, if provided. If not provided, they will be the VKI dataset name. This is the default. Use VKI dataset nameVariable names will be the VKI variable dataset name (which are reasonably descriptive).

Element Vars as	Single element values - Element results (whether centroidal or element nodal) will be presented as a single value per element. Thus will be per_elem variables in EnSight.This is the default. Averaged to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes without using geometry weighting. Thus will be per_node variables in EnSight. Geom weighted average to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes using geometry weighting. Thus will be per_node variables in EnSight
If Sections, which:	Which section will be used to create the variable First - The first section will be used (this is the default) Last - The last section will be used Section Num (below) - The section number entered in the field below will be used Separate Vars per Section - A separate variable will be created for each section.
Section Num	If the previous option is chosen to be Section Num, then the value in this field is the 1-based section number to use to create the variable.

(see How To Read Data)

AVUS Reader

Overview

The AVUS reader has been recently renamed, and was formerly called the COBALT reader.

CEI provides the AVUS user-defined-reader on as "as-is" basis, and does not warrant nor support its use.

There are two distinct readers for AVUS data (formerly Cobalt60) -- one for static data, AVUS (formerly Cobalt60), and one for transient solution data, AVUS Case (formerly Cobalt60 Case).

Both readers will read formatted and unformatted (single or double precision) Cobalt60 grids, solution files (pix files), and Cobalt60 restart files. The file format is determined automatically by the reader. The readers also support an enhanced solution (pix) format that contains additional solution data beyond the normal six fields.

See the following README file for current information on this reader and contact the author as listed in the README file for further information.

`$CEI_HOME/ensight92/src/readers/avus_cobalt_2/README`

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.grd`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and restart files (typically named with a suffix `.grd and .pix`) using the Advanced Interface method.

Data Tab	
Format	Use the **AVUS or AVUS Case** format.
Set grid (or file)	Select the geometry file (typically **.grd**) and click this button (or select the .case file for AVUS Case)
Set solution	Select the restart file (typically .pix), and click this button.
Format Options Tab	
Set measured	Select the measured file and click this button.

Limitation

This reader does not support restoring EnSight Context Files.

(see How To Read Data)

CAD Reader

Overview

This reader uses an external translation program to get various CAD files into an STL formatted temporary file which then read into EnSight. With the proper licensing, the following CAD file formats can be read: IGES (.igs), STEP (.STEP), CATIA V4 (.model, .dlv, .exp, .session), CATIA V5 (.CATPart, .CATProduct, or .CATDrawing, on Windows 32/64-bit only), Parasolid (.x_t, .x_b), Pro/Engineer (.prt, .asm), SolidWorks (.sldprt, .sldasm), Unigraphics (.prt), and possibly others.To manually convert this ProE data set into an STL using the CEI CAD translator, run this command:

```
ConvertSTL -i ./asm0002.asm.15 -o rs.stl
```

Additional licensing may be required. Please visit http://cad.ensight.com for more information.

The CAD reader will also load STL files directly (either ASCII or binary) that consist only of surfaces (triangles) and have no associated variables. See the STL Reader.

See the following README file for current information on this reader in the following directory.

```
$CEI_HOME/ensight92/src/readers/stl
```

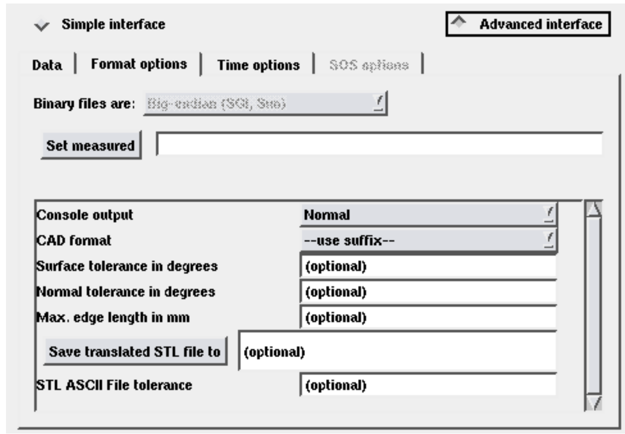
Simple Interface Data Load

Load your geometry file using the Simple Interface method and trust that the translator will recognize the file type using the suffix.

Advanced Interface Data Load

For more options, load your geometry using the Advanced Interface method and click on the Format Options Tab as described below.

Data Tab	
Format	Use the **CAD** format.
Set File	Select the CAD geometry file and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

Reader GUI	
Console Output	Allows the user control of the amount and detail of the console output. The allowable choices are as follows Normal - Typically only error messages displayed (the default) Verbose - Normal messages plus informational messages Debug - Messages indicating progress through the reader and useful for diagnosing problems
CAD Format	A pulldown to specify the format if the file name and extension are not sufficient for automatic selection of the CAD format. Allowable selections include CATIA v4, CATIA v5, IGES, Parasolid, ProE, SolidWorks, STEP, Unigraphics, and STL
Surface Tolerance in degrees	The maximum distance between a facet edge and the true surface. It is a floating point value between 0 and 360 degrees, with a default value: 15.0
Normal Tolerance in degrees	The maximum angle in degrees between two normals on two adjacent facet nodes. Default value: 25. It is a floating point value.
Max edge length in mm	Maximum length of a side of a cell in world space in millimeters. It is a floating point value with default of 0 (no max)
Save translated STL file to	The name of the translated STL output file. If this name is specified, then the STL output file is not automatically deleted after processing. Default value: determined by system call tempnam(). Example: /var/tmp/my_data.stl

STL ASCII file tolerance	A positive floating point value used to round off coordinate values read from ASCII STL files to compensate for the fact that there is often a roundoff error on the last digit which leads to discontinuity in the triangle facets. Example: 1.0E-3 - Sets tolerance value to 1.0E-3 For example: If coordinate value is 147.3247 and Tolerance is 1.0E-3 then new coordinate is 147.324 If Tolerance is 1.0E-2 then new coordinate is 147.32

(see How To Read Data)

CFF Reader

Overview

The CFF Reader is supplied compiled on Linux platforms on as "as-is" basis, and CEI does not warrant nor support its use. A README file as well as the source code is also supplied with the EnSight distribution in the directory below.

```
$CEI_HOME/ensight92/src/readers/cff
```

CFX4 Reader

Overview

Reads a 3D Static Cbinary dump (.dmp) file.

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/cfx4/README.txt`

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.dmp`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with a suffix `.dmp`) using the Advanced Interface method.

Data Tab	
Format	Use the **CFX-4** format.
Set cfx4 dmp	Select the geometry/results file (typically **.dmp**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

CFX5 Reader

Overview

Reads a CFX version 10, 11, or 12 results (.res) file.

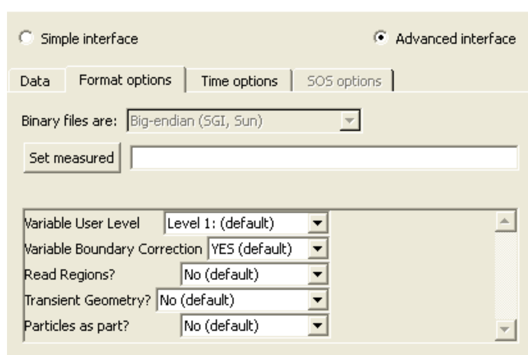
Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.res`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with a suffix `.res`) using the Advanced Interface method to customize the read, for example to read transient geometry (see below).

Data Tab	
Format	Use the **CFX-5** format.
Set file	Select the geometry/results file (**.res**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

Reader GUI	
Variable User Level	Allows the user control of number of variables read based on a call into the CFX API. The allowable choices are as follows: Level 0 - Read in all variables Level 1 - (default) Level 2 - Level 3 -

Variable Boundary Correction	Variable values are corrected using a boundary value correction if this toggle is Yes. YES - (default) Variable values adjusted using a boundary value correction. No - Variable values are not corrected.
Read Regions?	No - (default) - Do not read Regions. YES - Read Regions.
Transient Geometry?	A flag to the reader if the data is transient. Note: by default, a transient .res file will fail to load unless this is changed to Yes. CFx transient data will have a .res file and a series of .trn files (one for each timestep) located in a subdirectory. The res file will have the names of the .trn files, the time value and path. If the data is changing variables only then the .trn will not contain the mesh. If the mesh is moving, then the user must turn on the “Include Mesh” in the Transient Result options so tha tthe solver will write mesh information to each .trn file. Failure to do this results in a static, unmoving mesh over time. No - (default). Yes - Coordinates only.
Particles as Part?	If this is Yes, then EnSight reads in the particle data as a separate EnSight point part. No - (default) Do not read in particles as a separate EnSight Part. Yes - Read in the particles as a separate EnSight part

The CFx solver does export to EnSight Case Gold format.

(see How To Read Data)

CGNS Reader

Overview

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/cgns/README.txt`

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.cgns`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results files (typically named with a suffix `.cgns)` using the Advanced Interface method.

Data Tab	
Format	Use the **CGNS** format.
Set cgns	Select the geometry/results file (typically **.cgns**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

CTH Reader

Overview

Reads a Spymaster .spcth file.

See the following file for current information on this reader.

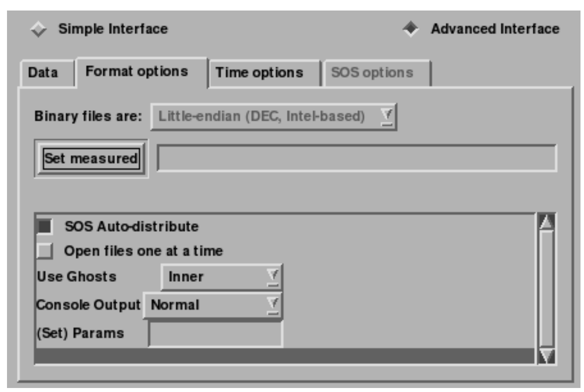
`$CEI_HOME/ensight92/src/readers/cth/README.txt`

Simple Interface Data Load

Load your geometry/results file(s) (typically named with a suffix `.spcth`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results files (typically named with a suffix `.spcth`) using the Advanced Interface method.

<table>
<tr><th colspan="2">Data Tab</th></tr>
<tr><td>Format</td><td>Use the CTH format.</td></tr>
<tr><td>Set spcth*</td><td>To read one Spymaster file, put the CTH Spymaster file (typically named something filename.spcth) into the (Set) Geometry field. To read in multiple, related CTH Spymaster files (for example several files solved in parallel) as follows:
<code>filename.spcth.0</code>
<code>filename.spcth.1</code>
<code>filename.spcth.2</code>
Put an asterisk ‘*’ in the filename (filename.spcth.*).</td></tr>
<tr><th colspan="2">Format Options Tab</th></tr>
<tr><td>Set measured</td><td>Select the measured file and click this button.</td></tr>
<tr><td>Reader GUI</td><td>User controls as shown below are available:

</td></tr>
</table>

(see How To Read Data)

ESTET Reader

Reader Visibility Flag

By default, this reader is not loaded into the list of available readers. To enable this reader go into the Menu, Edit > Preferences and click on Data and toggle on the reader visibility flag.

Overview

ESTET input data consists of one file that contains all geometry and results information. Data is loaded in a multi-step process. The ESTET data is a structured grid. The data file is binary.

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.estet` using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with a suffix `.estet`) using the Advanced Interface method.

Data Tab	
Format	Use the **ESTET** format.
Set geometry	Select the geometry/results file (typically **.estet**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

When reading this data into EnSight, the part loader dialog will be essentially the same as that found in the reader basics for structured data (See Reading and Loading Data Basics). The data can be in rectangular, cylindrical, or curvilinear coordinates. EnSight will interpret and convert properly for any of these types.

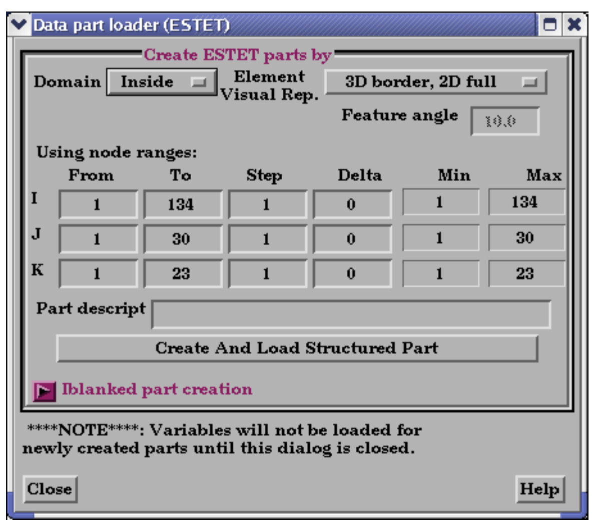


Figure 2-10
ESTET Vector Builder and Data Part Loader dialogs

One difference from the Reader Basics, is that once the desired geometry has been extracted as Parts, you are presented with a list of the results variables contained in the file. There is no way to automatically determine which of the results variables are actually vector components, so you are given the opportunity to build the vectors from the variables. The descriptions usually make this a straightforward process. All variables not used as components to vectors are assumed to be scalar variables.

ESTET Vector Builder and Data Part Loader dialogs

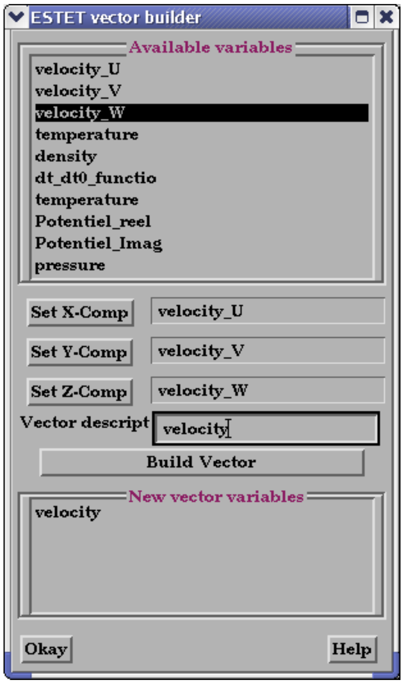


Figure 2-11
ESTET Vector Builder and Data Part Loader dialogs

You use the File Selection dialog to read ESTET data files, the ESTET Vector Builder dialog to build vector variables from scalar components for an ESTET dataset, and the Data Part Loader dialog to extract Parts from an ESTET dataset. The latter two dialogs open in sequence automatically after you click Okay in the File Selection dialog.

`Access: Main Menu > File > Data (Reader)…> ESTET`

(see How To Read Data)

EXODUS II Gold Reader

Overview

Misc Notes

The Exodus reader links to the exodus routines in libexoIIc.a and the netcdf routines in libnetcdf.a. You must have these libraries to compile and run the ExodusII reader.

Variable names that end in "_x", "_y", "_z" will be treated as components of a vector. For example, the variables "vel_x", "vel_y", "vel_z" will be treated as a vector named "vel_vec". Case is ignored in matching variable names.

GUI control of Reader

Note reader behavior can also be controlled in the Data Reader GUI via checkboxes and fields. The environment variables, if set, are used to set the default values for the GUI.

Advanced Multiple File Naming

This reader supports two extensions to the filename fields. The first supports Exodus datasets where the geometry changes at some point in time. In this case, a new Exodus file (or set of files), is used for each set of solution times. To support this feature, insert wildcard characters (e.g. "*" and "?") in the filename that expand to the name of the first files in each timeset.

The second extension allows for multiple files to be read as part of the same timeset (e.g. domain decomposed files). This feature takes the form of a tag (<X:Y>) appended to the filename. The value of "X" is the number of files in the timeset and "Y" is the sprintf() format string on the integer (%d) formatting options) for expanding an integer argument into a string. For example, if a dataset consists of the following files:

Times 0-10	Times 11-15
foo.e.03.00	foo.e-s0002.03.00
foo.e.03.01	foo.e-s0002.03.01
foo.e.03.02	foo.e-s0002.03.02

Where the "foo.e.*" files contain timesteps 0 through 10 and the "foo.e-s0002.*" files contain timesteps 11 through 15. Also, each timeset is spatially decomposed into 3 subfiles, which each contain some portion of the dataset for the given timesteps. For this dataset, use the file pattern "foo.*.03.<3:%0.2d>" to tell EnSight there are multiple timesets and to generate the filenames for each timeset by replacing the "<..>" substring with a number from 0 to 2 as generated using sprintf() and "%0.2d". Note that the "<...>" marker must be the last part of the filename. The '*' will wildcard the timeset number and the "<...>" specifies the spatial decomposition.

When using SoS casefiles, the servers will automatically distribute the subfiles over all of the specified servers. There can be no more servers than the number of sub-files however (but fewer servers are legal and in most cases, recommended). Finally, if there are sub-files, the number and the naming convention must be the same over all timesets.

Autodetection of Spatially and Temporally Decomposed Datasets

The Exodus reader includes a 'Autodetect Spatial Decomp' option. In the 'MultiExodusII' reader, this option is off by default, but in the 'MultiExodusIIng' reader, it is enabled by default. When using this option, simply select one of the files in the dataset and EnSight will attempt to figure out the spatial and temporal decomposition scheme by parsing the filename and scanning the directory.

If your dataset naming conventions follow the example above, this option is a much simpler way of opening the entire dataset. In the previous example, the user need only select to load the file 'foo.e.03.00' to load the entire dataset. For must users who follow the general filenaming guidelines, it is strongly suggested they use the 'MultiExodusIIng' reader (and its dataset autodetection scheme) to load their decompoased data.

The "autodetection" scheme parses the name, looking for a trailing .X.Y scheme, where X and Y are zero padded integers, to detect that the file is spatially decomposed. If the name does not end in this form, it is assumed that the dataset is not spatially decomposed. The scheme also parses the filename looking for the string '.e' and checks the current directory for other files in the same directory that have other text between the '.e' and the spatial '.X.Y' (if it exists). If multiple files are found (checked using patial chunk 0), then the autodetection assumes that the dataset is temporally decomposed as well. The reader will load all of the Exodus II files matching the autotected scheme as a single temporal and/or spatially decomposed dataset.

Note that the option effectively prevents you from being able to open an individual Exodus temporal or spatial "chunk", but for most situations, that is not an issue.

Environmental variables

The following Environmental variables can be se to modify the behavior of the Exodus reader. Many of these are available as well in the reader dialog. When you choose Exodus reader, under the format options tab, will be a number of options for controlling the reader behavior. The advantage of setting the Environmental variables is that you can control the default behavior.

ENSIGHT_EXODUS_SCALE - set to a scaling factor (default 1.0)

ENSIGHT_EPSILON - set to a temporal epsilon (default 1.0)

ENSIGHT_EXODUS_DF (or EXODUS_DF) - if nonzero, support distribution factors (off by default)

ENSIGHT_EXODUS_NO_SIDESETS - if nonzero, disable sideset support (enabled by default)

ENSIGHT_EXODUS_NO_NODESETS - if nonzero, disable nodeset support (enabled by default)

ENSIGHT_EXODUS_VERBOSE - if nonzero, enable verbose mode (off by default)

ENSIGHT_EXODUS_USE_NODEMAPS - if nonzero, use node maps (default true)

ENSIGHT_EXODUS_USE_HIGHERORDERELEMENTS - if non-zero, read element order will be presered in EnSight. if zero, return first order elements for higher order elements in the file (e.g. convert a HEX20 into a HEX08). (default true)

ENSIGHT_EXODUS_IGNORE_CONSTANTS - if non-zero, constant variables will not be read (default false)

ENSIGHT_EXODUS_CHECK_NANS - if non-zero, all read floats will be checked for IEEE nans and set to zero if nan.

ENSIGHT_EXODUS_CLIP_TIMESTEP_OVERLAP - if non-zero, when multiple timesets result in overlapping time ranges, the ones in the first file read are dropped.

ENSIGHT_EXODUS_AUTODETECT_DECOMP - if set and the filename passed ended in the numbering scheme used in spatial decomposition of the form .XX.00 the reader will convert the filename to the brace form: .XX.<XX:%0.2d>. This allows the user to select the first file in the decomposition and have the proper template automatically generated.

ENSIGHT_EXODUS_USE_UNDEF_VALUE - if non-zero (the default), the reader will return the server 'Undefined' float value when a variable is not defined on a part. if set to 0, the value returned is actually 0.0 (this is the behavior for the reader prior to 2.72).

ENSIGHT_EXODUS_NO_ELEMENT_ATTRS - if non-zero, the reader will not attempt to read/use any element attribute variables.

ENSIGHT_EXODUS_USE_FULL_NAMES - By default, this option is set off (0) and names are "reduced" to the original 19 character limits. If this option is enabled (non-zero), the limit is updated to the current EnSight limit of 49 characters.

ENSIGHT_EXODUS_USE_DTA_FILE - By defualt, if there is a .dta file present, the reader will pass its contents on the to the client. If this is set to 0, .dta files will be ignored.

ENSIGHT_EXODUS_AUTOGEN_DTA - if non-zero, and there is no .dta file, a "stub" .dta file will be automatically generated and passed to the client. This is disabled by default.

Names file format

The .names file can be used to name parts in a Exodus file. Note that the names for these files are generated by clipping off the input Exodus filename at the rightmost '.' and adding the suffix ".materials" or ".names". External part names and materials files have the same syntax.

```
*    File containing block, sideset, nodeset names
*    ---------------------------------------------
*    (Must have the same root as the .exo file, and reside in the same
*     directory, and have a .names extension)
*
*    Note:  1. Comments have a # in first column of the line.
*
*           2. Three types of sections (must be exactly as indicated):
*
*                 blocks
*                 sidesets
*                 nodesets
*
*               (Presence of any of the sections is optional - for example,
*                 if you only have blocks, you don't need the sidesets or
*                 nodesets sections.)
*
*           3. There must be a number, white space, then name
*                on lines within the sections.
*
*                numbers must be 1-based (relates to the block number)
*
*                names must be a single token - no spaces (Underscores or
```

```
*                    dashes are okay.)
*
*            4. Line length must be less than 80 chars - which should be
*                plenty since EnSight truncates names at 49 chars.
*
*     Thus, general format is:
*
*     blocks
*     1      name_for_block_1
*     2      name_for_block_2
*     .         .
*     .         .
*     .         .
*     n      name_for_block_n
*     sidesets
*     1      name_for_sideset_1
*     2      name_for_sideset_2
*     .         .
*     .         .
*     .         .
*     m      name_for_sideset_m
*     nodesets
*     1      name_for_nodeset_1
*     2      name_for_nodeset_2
*     .         .
*     .         .
*     .         .
*     p      name_for_nodeset_p
*
*
*     Here is an Example:
*     -------------------
*     blocks
*     1               Strongback
*     2               bolt_1
*     3               bolt_2
*     4               bolt_3
*     5               nut_1
*     6               nut_2
*     7               nut_3
*     8               Tab
*     9               screw_1
*     10              screw_2
*     11              screw_3
*     12              Block
*     sidesets
*     1               Strongback_end
*     2               Strongback_front
*     3               Strongback_back
*     nodesets
*     1               Tab_nodes
```

For the previous example, the files would be named "foo.1_03.names" and "foo.1_03.materials". Only one .names and one .materials file is required.

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/exodus_gold/README.exodus`

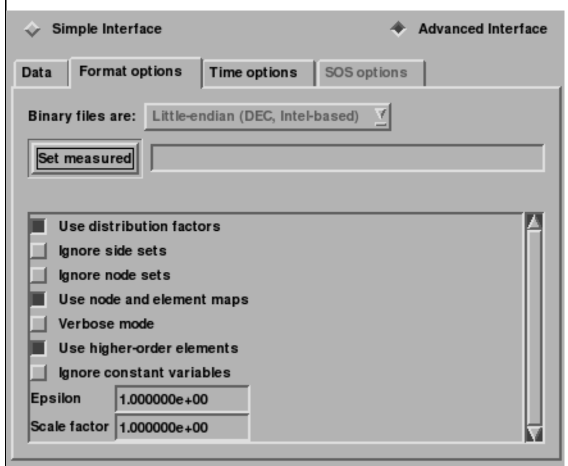
Data Reader

Simple Interface Data Load

Load your geometry/results file (typically named with suffix .ex, or .ex2, or .exo) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with suffix .ex, or .ex2, or .exo) using the Advanced Interface method.

<table>
<tr><th colspan="3">Data Tab</th></tr>
<tr><td>Format</td><td colspan="2">Use the MultiExodusII or MultiExodusIIng format.</td></tr>
<tr><td>Set exo</td><td colspan="2">Select the geometry/results file (typically .ex, or .ex2, or .exo) and click this button. If there are multiple Exodus files in the solution set, you may enter a filename with wildcard characters, for example, "mydata*.exo2" or "mydata1?.exo2"</td></tr>
<tr><th colspan="3">Format Options Tab</th></tr>
<tr><td>Set measured</td><td colspan="2">Select the measured file and click this button.</td></tr>
<tr><td rowspan="2">Extra GUI Parameters</td><td colspan="2">These toggles and fields below are customized for the Exodus reader.

</td></tr>
<tr><td>Use Distribution Factors</td><td>If nonzero, use support distribution factors (off by default). Node sets and side sets in Exodus can have distribution factor weights associated with them. If this option is set, these distribution factors will be read by EnSight as additional scalar variables on the node or side set parts. If the Environment Variable ENSIGHT_EXODUS_DF (or EXODUS_DF) is set, then the reader will use it to set the default value.</td></tr>
</table>

	Ignore Side Sets	If nonzero, disable sideset support (enabled by default). If the Environment Variable ENSIGHT_EXODUS_NO_SIDESETS is set then the reader will use it to set the default value.
	Ignore Node Sets	If nonzero, disable nodeset support (enabled by default). If the Environment Variable ENSIGHT_EXODUS_NO_NODESETS is set then the reader will use it to set the default value.
	Use Node and Element Maps	If nonzero, use node maps (default true). Exodus files may contain element and nodal labels, referred to as nodemaps. If this option is set, EnSight will use the labels in the Exodus files. There is no guarantee (particularly when using spatially decomposed Exodus files) that these numbers are unique over all the nodes and elements. If this option is not set, EnSight will ignore the labels in the Exodus file and use an internal numbering scheme guaranteed to generate unique node and element labels. If the Environment Variable ENSIGHT_EXODUS_USE_NODEMAPS is set then the reader will use it to set the default value.
	Verbose Mode	If nonzero, enable verbose mode (off by default). See the console for the verbose output. If the Environment Variable ENSIGHT_EXODUS_VERBOSE is set then the reader will use it to set the default value.
	Use higher-order elements	Higher order elements will be preserved and not down converted to simpler elements.
	Ignore constant variables	Does not load single-value variables (called Constants in EnSight).
	NaN filter input data	Check all floats for validity.
	Clip overlapping timesteps	Restarts datasets that overlap in time use the later datasets to construct the timeline
	Autodetect spatial decomp	Automatically recognize all the files using one of the files.
	Use Undef value for missing vars	Use EnSight's Undef value to indicate missing variables
	Ignore element attribute vars	Do not read/use element attribute variables.

	Use detected DTA XML file	Use a specific XML formatted file to name parts and assign attributes
	Auto generate DTA XML file	Auto generate this XML file and use it for part naming.
	Use full object names	Long variable names.
	Epsilon	Set to a temporal epsilon (default 1.0). This number is used to adjust non-monotonically increasing solution times. If the reader detects consecutive solution times (as floats) that do not progress forward in time, the later solution time value will be set to the earlier time plus this value. Note: this can have a cascading effect shifting other solution times later as well. This feature is generally useful when the solution times in the Exodus II file are all the same value. If the Environment Variable ENSIGHT_EPSILON is set, then the reader will use it to set the default value.
	Scale Factor	Set to a scaling factor (default 1.0). This number is multiplied by each of the x, y, and z geometry coordinates in order to scale the geometry. If the Environment Variable ENSIGHT_EXODUS_SCALE is set, then the reader will use it to set the default value.

(see How To Read Data)

FAST UNSTRUCTURED Reader

Overview

FAST UNSTRUCTURED is a format containing triangle and/or tetrahedron elements. The triangles have tags indicating a grouping for specific purposes. EnSight will read the unstructured single zone grid format for this data type, placing all tetrahedral elements into the first Part, and the various triangle element groupings into their own Parts.

Simple Interface Data Load

Load your grid file using the Simple Interface method.

Advanced Interface Data Load

Load your grid and solution files using the Advanced Interface method.

Data Tab	
Format	Use the **FAST Unstructured** format.
Set grid	Select the grid file and click this button. This is the FAST UNSTRUCTURED single zone grid file. Defines the geometry as unstructured triangles and/or tetrahedrons.
Set solution	Select the solution file and click this button. The Results file can either be a Modified Result file which utilizes a modified EnSight results file format, or can be variable files (optional) which are either a PLOT3D solution file (Q-file) or FAST function file with I = number of points and J=K=1. The modified EnSight results file provides access to multiple solution files that are produced by time dependent simulations. FAST UNSTRUCTURED data can have changing geometry. When this is the case, the changing geometry file names are contained in the results file. However, it is still necessary to specify an initial geometry file name. WARNING: Do not use your solution file (e.g. file.q) here. You must create a special results file to handle FAST variable files.
Format Options Tab	
Set measured	Select the measured file and click this button.

Part Loader

The FAST UNSTRUCTURED reader uses a simplified Part Loader as follows:

- All Parts: all parts are loaded to the client in the default visual representation (typically 3D Border, 2D Full).
- Part 1 Only: Only the first part is loaded to the client in the default visual representation. The other parts will have the NonVisual representation.
- All But Part 1: All parts other than part 1 are loaded to the client in the default visual representation. Part 1 will be NonVisual.
- No Parts: No parts are loaded to the client (i.e.the representation of all parts is set to NonVisual).

(see How To Read Data)

FIDAP NEUTRAL Reader

Overview

A FIDAP Neutral file contains all of the necessary geometry and result information for use with EnSight.

A neutral file is produced by a separate procedure defined in the FIDAP documentation. If the data is time dependent this information is also defined here.

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.fdneut`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with a suffix `.fdneut`) using the Advanced Interface method.

Data Tab	
Format	Use the **FIDAP Neutral** format.
Set geometry	Select the geometry/results file (typically **.fdneut**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

Part Loader

The FIDAP reader uses a simplified Part Loader as follows:

- All Parts: all parts are loaded to the client in the default visual representation (typically 3D Border, 2D Full).
- Part 1 Only: Only the first part is loaded to the client in the default visual representation. The other parts will have the NonVisual representation.
- All But Part 1: All parts other than part 1 are loaded to the client in the default visual representation. Part 1 will be NonVisual.
- No Parts: No parts are loaded to the client (i.e.the representation of all parts is set to NonVisual).

(see How To Read Data)

FLOW3D-MULTIBLOCK Reader

Overview

This EnSight reader uses the FLSGRF READER API LIBRARY from FlowScience in order to read data from a FLOW3D "flsgrf" output data file, which ends in `.dat` or in `.fgz` if compressed in their API.

Requirements

The FLSGRF API requires that the user have **write** permissions in the directory where the flsgrf file resides.

THE FLSGRF API is only available for Windows 32-bit, 64-bit, Linux 32-bit, and Linux 64-bit platforms.

The FLSGRF API is for FLOW3D version 9.2, however the FLSGRF API should read flsgrf files written by older versions of FLOW3D.

Data Types

There are several kinds of data available in a FLOW3D flsgrf output file, each with it's own EnSight timeset: Restart, Selected, Fixed, and Particle data.

Restart data

By default, there are 11 restart timesteps per solution: t=0 and an additional ten each spaced at 1/10th of the total simulation time. The user can change this frequency in FLOW3D using the plotting interval (PLTDT) in the PREPIN input file.

Selected data

This consists of selected variables output at a higher number of timesteps. Restart and selected data can both be available in the data file.

Fixed time group

This data does not change over time. It consists of simulation parameters such as binary flags used to activate physical models as well as mesh data.

Particle data

If particles are present in the simulation they will be present in the Restart data. If the user requests particle information in the Selected data (from their project file where anmtyp(i)='part') then particle information will also be available in the Selected data timesets.

Particle data is not imported into EnSight by default. To import this data, Choose the Multipart Flow3d reader then click on the Format Options Tab and select the type of Particle data that you wish to import.

Geometry is STATIC in EnSight unless Particle data is imported and then the geometry is CHANGING CONNECTIVITY because the number of particles can change with each timestep.

Technical notes

All block part variable data is cell-based (per element data). All particle variable data is node-based (per node data).

To visualize the fluid in the block try creating an isosurface of the Fluid Fraction using an isosurface value of 0.51. Now to see the surrounding structure, make another isosurface with a value of 0.51 using the Cell_Volume_Fraction_Fixed variable. Click on the paint can icon and change your shading to smooth to improve the isosurface look. To see the fluid as an isovolume, File>command and type in 'test: simple_isovolume_off' then make an isovolume from 0.5001 to 1.0 using the Fluid Fraction and again change it's shading to smooth.

To see detailed information about variables, blocks, boundary conditions, etc. choose Console Output Debug as described below.

This reader makes use of Timesets. Each of the variable types has it's own Timeset timeline and EnSight merges them all together. For details on using these timesets see the advanced section of the Change Time Steps in the How To Manual.

Ghost cells

Ghost cells are invisible elements that help EnSight to interpolate variable values. For example, Ghost cells between blocks allow for a smooth transition of the isosurface of the fluid surface at part block boundaries. Ghosts that are not used have a zero value for the variable, and must be removed: removal of ghosts at a symmetry surface allows for smooth mirroring of the part(s).

`Updated info`

`$CEI_HOME/ensight92/src/readers/multi_flow3d/README.txt`

Simple Interface Data Load

Load your Flow3d file (typically named `flsgrf.dat`) using the Simple Interface method.

Advanced Interface Data Load

Load your Flow3d file (typically named `flsgrf.dat`) using the Advanced Interface method.

Table 1:

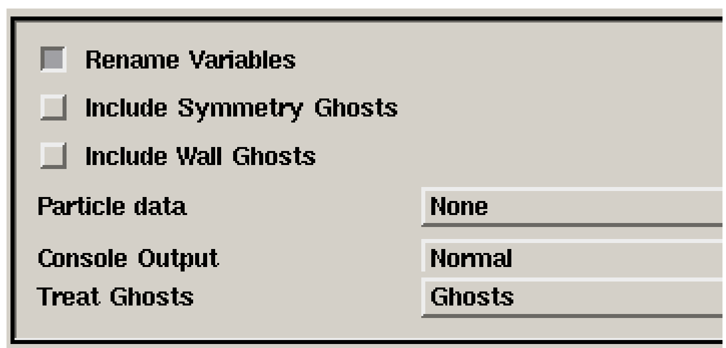
Data Tab	
Format	Use the **Flow3d-Multiblock** format.
Set flsgrf	This field should be an flsgrf file.
Format Options Tab	
Format Options	
Rename Variables	Variables are renamed by default to be easier to interpret for Flow3D users.

Table 1:

Include Symmetry Ghosts	Flow3d includes the following boundary conditions: symmetry, wall, continuative, periodic, specified pressure, specified velocity, grid overlay, outflow and interblock. Ghost elements are always included in periodic, specified pressure, and inter-block boundary conditions. Ghosts are always removed for continuative, specified velocity, grid overlay and outflow boundary conditions. Ghosts are optional for symmetry and wall boundary conditions. By default, symmetry ghosts are removed so that mirroring in EnSight has no seam. However toggling ON this flag will include the symmetry ghosts for example for inviscid flow using the symmetry condition to simulate a wall.
Include Wall Ghosts	Flow3d includes the following boundary conditions: symmetry, wall, continuative, periodic, specified pressure, specified velocity, grid overlay, outflow and interblock. Ghosts are always included in periodic, specified pressure, and inter-block boundary conditions. Ghosts are always removed for continuative, specified velocity, grid overlay and outflow boundary conditions. Ghosts are optional for symmetry and wall boundary conditions. By default, wall ghosts are removed so that the EnSight fluid/wall interface does not show up using the isosurface of the fluid fraction. However toggling ON this flag will cause the fluid/wall interface to show up using the isosurface of the fluid fraction.
Particle Data	By default, particle data is not read into EnSight. Toggle this ON to read in the Particle data.
Console Output	Use this flag to determine the amount of output to the console. Normal - Usually only echo errors to console. Verbose - Normal output plus an echo of every Fluent part that is in the dataset, whether it is interior or not, whether it is skipped, what variables are defined for which parts, and to echo it's Ensight Part number. Debug - Verbose output plus more detailed output and progress through the reader routines often valuable for understanding and reporting problems. Also detailed block information, variable information, boundary condition information, and timeset information

Table 1:

Treat Ghosts	Ghost elements are invisible elements that help EnSight to interpolate variable values. Ghosts can be read in as Ghost cells, as normal (visible) cells, or they can be not read in at all. 1. Ghosts - include invisible ghost elements according to the Flow3d boundary conditions (default) and user settings. 2. None - Use NO ghost elements. This will especially be apparent in the gaps in the isosurface between data blocks due to the lack of these invisible interpolating elements. 3. Normal - Use ALL ghost elements as NORMAL, visible elements. This is useful for understanding boundary conditions around part blocks.
Set measured	Select the measured file and click this button.

(see How To Read Data)

FLUENT Direct Reader

Overview

There are three methods to get Fluent data into EnSight. The first is to use the current Fluent reader. This loads a Fluent Case (.cas) file and data (.dat) file. The second method to get data into EnSight is to use Fluent's EnSight Case Gold Export option and read it directly into EnSight using the Case reader.The last method is a little-used legacy reader for Fluent Universal file and is described later under the FLUENT UNIVERSAL Reader.

See the following files for current information on the Fluent direct reader.

`$CEI_HOME/ensight92/src/readers/fluent/README.txt`

The comments that follow are for the current Fluent reader. The reader loads ASCII, binary single precision, or binary double precision. The files can be uncompressed or compressed using gzip. Note also, this reader is used to load AIRPAK/ICEPAK .fdat data files (see AIRPAK/ICEPAK Reader) and .cdat ('lite' data file, or 'extra CFD variables' file).

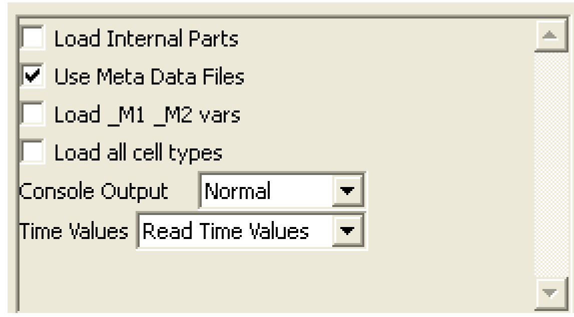
Simple Interface Data Load

Load your geometry file (typically named with a suffix `.cas`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files (typically named with a suffix `.cas and .dat`) using the Advanced Interface method.

Data Tab	
Format	To use this reader, select the **Fluent** format.
Set cas	Select the geometry file (typically **.cas** or **.cas.gz**) and click this button. For transient data, use *.cas or *.cas.gz. For the old Fluent reader the asterisk must replace a 4-digit number.
Set dat	Select the results file (typically **.dat** or **.dat.gz**), and click this button. For transient data, use ***.dat** or ***.dat.gz** . Note that **.cdat** files ('lite' dat file, or an 'extra CFD variables' file) are also useable in place of the .dat file. Finally **.fdat** files (see Airpak / Icepak reader) are also useable in place of the .dat file. Because the .dat file is the automatic selection, if a .dat file is colocated with a .cdat or a .fdat file, the user will have to manually select the .cdat or .fdat file and then click on the Set dat button to use these .dat file variants.
Format Options Tab	
Set measured	Select the measured file (typically a .mea suffix) and click this button. If you have fluent particle data you can translate it into EnSight's measured data format and import the particles as measured data.

Other Options using the current Fluent reader	
Load Internal Parts	Toggle this ON to load the Fluent Internal Parts. This will show all the the internal walls forming all the cell volumes. If you do toggle this on, then it is recommended that you click on the 'Choose Parts' button at the bottom of the data reader dialog, rather than 'Load all', as you'll only want to load the interior parts of interest to save memory and time. Default is OFF.
Use Meta Files	Meta files are small summary files that contain highlights of the important locations inside each of the Fluent files. Allowing the EnSight reader to write out Meta Files that map the locations of important data can provide a significant speed up the next time you access that timestep. It is recommended that you leave this toggle ON. If you have write permission in the directory where your data is located, three types of binary Meta Files will be written when you first access each file, with extensions .EFC for the cas file, .EFD for the .dat file and .EFG for the time-history data. They are optional, and if you don't have write permission, the reader will take the extra time to read the entire .CAS and .DAT file to find the relevant data each time you come back to that timestep.
Load _M1 _M2 vars	Variables that end in '_M1' and '_M2' occur in Fluent unsteady flow. They represent the value of the variable at the prior iteration time and the time prior to that respectively. By default this toggle is OFF and these variables are not loaded. Toggle this ON to load these variables.
Load all cell types	Fluent cells have a boundary condition flag. By default (toggle OFF) EnSight loads only the cells with a boundary condition flag equal to 1 (one). Toggle this option ON to load all cells with a non-zero boundary condition. For example, if you have a part with cells of boundary condition 32 (inactive), EnSight will, by default not load this part. Toggle this option ON and EnSight will load this part. Note: parts containing cells with a boundary condition of zero are never loaded.

Poly to Regular Cell	Fluent polyhedral cells when composed of the correct kind and number of regular faces can be converted to regular cells (tetrahedrons, hexahedrons, pyramids, or wedges) boosting EnSight speed and decreasing memory requirements. Toggle ON and reader checks each polyhedron to see if it can be converted to a regular cell (default) and OFF to not convert any polyhedral cells. There is very little slowdown during the read to do this and a big payoff for some datasets with large numbers of convertible polyhedra. Leave it on.
Poly faced Hex to Poly	Fluent hex cells that transition to a more refined hex mesh will sometimes have one or more of the quad4 faces subdivided into four quad4 faces. For example a hexahedral cell with one transition face will have the six full faces, and four subdivided faces for a total of 10 faces. Toggle ON and the the subdivided faces are kept rather than the full face and the cell is changed into a polyhedral which will slow down EnSight performance, and greatly increase memory. The polyhedral also will have hanging nodes (see the next toggle). Toggle OFF (default) to convert the hex element to a six-sided hex which will be adjacent to four smaller cells rather than completely connected to them slowing down EnSight's adjacency searching. The default is thought to be the lesser of two evils.
Fix Hanging Nodes	Some Fluent polyhedral cells and all transition hex cells converted to polyhedrals will have hanging nodes. A hanging node not shared by at least 3 faces. A polyhedral element with hanging nodes is not water tight and can cause real problems in EnSight, so it is best to leave this toggle ON (default) and only turn it OFF for experimental purposes.
Console Output	Use this flag to determine the amount of output to the console. Normal - Usually only echo errors to console. Verbose - Normal output plus an echo of every Fluent part that is in the dataset, whether it is interior or not, whether it is skipped, what variables are defined for which parts, and to echo it's Ensight Part number. Debug - Verbose output plus more detailed output and progress through the reader routines often valuable for understanding and reporting problems.
Time Values	Default is 'Calc Const Delta', to read a delta time from one file and calculate the time values from that. If you choose 'Read Time Values' then the reader will open each file and find the exact time value. This will be stored in the EFG file if you've not disabled Meta Files. Finally, the simplest is to 'Use File Steps' which will just use the file step number as the time value. This is quick, but is not a good idea if you need real time for anything such as particle tracing.

Node and Elem IDs

Parts have node and element ids to enable querying your data. Node ids are created from the coordinate global node. Element ids are created as follows. Face part elements are uniquely numbered according to their zone index. Cell part elements are uniquely numbered using their zone index added to the total number of faces. So a dataset with 100 face elements and 300 cell elements would have the face elements number 1-100 and the cell elements numbered 101-400. In Verbose mode, the element id range for each part are written to the console.

Variable Location

Variables that are cell-centered remain where they are found in the .dat file, that is the reader does not interpolate the cell-centered variables to the nodes. Fluent can export variable data to EnSight's Case Gold format at either the nodes (default) or at the elements (same as .dat file). Unfortunately older versions of Fluent export variables averaged to the nodes leading to flow into and out of walls which causes particles to stop prematurely and skews mass flow calculations. Later versions of Fluent consider boundary conditions prior to averaging the data to the nodes, yielding a much more realistic representation of the physics.

Variables Undefined

Not all variables exist on all parts. If you select a part and color by a variable and get undefined, then load the data using Verbose mode and take a look at the console. Your variable is probably not defined for this part The EnSight reader does not currently extrapolate data from the cells to faces. Create a clip on the location of an interior part on a volume part if you want to see a plane with the values from the volume.

Variable Names

Many variables that formerly were named improperly have been fixed. However if you see a variable that is not named properly, send the variable number and what it should be named to support@ensight.com and we will ask Fluent the proper name and fix it.

Extra Variables

EnSight will try to calculate extra CFD variables given the existing DAT variables for your convenience.

CAS Constants

Extra CAS Single Value Variables - A number of single value variables are read from the CAS file(s). These will show up in the EnSight Calculator named as follows.

'PRESSURE_ABS' - operating pressure (absolute)

'PRESSURE_ABS_INIT'- initial operating pressure (absolute)

'GAMMA_REF' - reference gamma, ratio of specific heats

'VISCOSITY_REF' - reference viscosity

'TEMPERATURE_REF' - reference temperature

'PRESSURE_REF' - reference pressure

'DENSITY_REF' - reference density

'SPEED_SOUND_FAR' - far field speed of sound

'PRESSURE_FAR' - far field relative pressure

'DENSITY_FAR' - far field density

'R_ref' - Calculated reference gas constant = PRESSURE_ABS / (DENSITY_REF * TEMPERATURE_REF)

'V_def' - Calculated default velocity magnitude from x, y and z-velocity default values

'M_def' - Calculated from V_def / SPEED_SOUND_FAR

UDS, UDM Variables

UDM and UDS variables now read in as UDM_0, UDM_1, UDM_2... and UDS_0, UDS_1, ... Fluent differentiates between UDS (User defined scalar) and UDM (user defined memory) as follows.

A UDS is a scalar variable for which a transport equation can be solved (e.g. transport of a red color from an injection nozzle into the volume; convective terms, diffusive terms,..) The single terms of this transport equation are programmed via UDF (user defined functions) in C and are run time libraries.

A UDM is a node-based value which also is calculated using a UDF (e.g.viscosity against local temperature and density). For UDMs transport equations are not solved. Thus they require less memory compared to UDS.

UDMs and UDSs are available for additional physics which are not available in Fluent (for example one can use a UDM for the calculation of dust concentration in filter elements.

Polyhedral elements

There are two methods to import polyhedral elements into EnSight. The first is to try using the direct reader. This has the advantage that the direct reader will attempt to convert polyhedral elements back to regular elements, saving memory and speeding up EnSight. The second is to export EnSight Case Gold from Fluent. Case Gold has the advantage of currently supporting automatic Server of Server decomposition, which can distribute the many tasks to multiple servers and speed up post processing.

Periodic elements

The reader now supports rotational symmetry to provide continous boundaries.

Particles

Included with EnSight is a Fluent particle file translator to translate the Fluent `.part` file into an EnSight measured (`.mea`) data file. To get help with this translator, type

```
$CEI_HOME/ensight92/machines/$CEI_ARCH/flupart -h
```

where $CEI_ARCH is your hardware/OS architecture (e.g. linux_2.6_64 or apple_10.5, win32, etc.).

Source code and README for this translator are located

```
$CEI_HOME/ensight92/translators/fluent/Particles/
```

This measured data file is entered in the the measured data field under the Format Options tab of the data reader dialog.

(see How To Read Data)

FLUENT UNIVERSAL Reader

Reader Visibility Flag

This is not the preferred reader for Fluent data. Therefore, by default, this reader is not loaded into the list of available readers. The preferred reader for Fluent data is the Fluent Direct Reader. If you have Universal file data, you can enable this reader as follows: go into the Menu, Edit > Preferences and click on Data and toggle on the reader visibility flag.

The FLUENT Universal file contains all of the necessary geometry and result information for use with EnSight for a steady-state case. If the case is transient, EnSight needs a Universal file for each time step of the analysis and a modified version of the EnSight results file.

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.univ or .fluniv`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with a suffix `.univ or .fluniv`) using the Advanced Interface method.

Data Tab	
Format	Use the **Fluent Universal** format.
Set universal	Select the universal file (typically **.univ or .fluniv**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

Part Loader

The Fluent Universal reader uses a simplified Part Loader as follows:

- All Parts: all parts are loaded to the client in the default visual representation (typically 3D Border, 2D Full).
- Part 1 Only: Only the first part is loaded to the client in the default visual representation. The other parts will have the NonVisual representation.
- All But Part 1: All parts other than part 1 are loaded to the client in the default visual representation. Part 1 will be NonVisual.
- No Parts: No parts are loaded to the client (i.e.the representation of all parts is set to NonVisual).

(see How To Read Data)

Inventor Reader

Overview

Reads inventor (.iv) datasets for which there is a one-to-one correspondence in Part, coordinate index, geometry index and element record. That is, there is one set of coordinates, one geometry, one set of elements per inventor node. To read in one inventor file, enter the .iv filename.

To read in multiple .iv files us a Case Inventor file (.civ). The user has the option in the reader Format Options Tab to toggle off one part per file option (default on). The .civ file is an ASCII file with the number of files on the first line, and the filenames on the remaining. The filenames must be in quotes, and if they don't include the path, the .civ file must be in the directory where the files are located. A '.civ' filename cannot be in a '.civ' file: only '.iv' filenames are allowed.

Example Format:

```
numfiles: 2
"filename1.iv"
"filename2.iv"
```

For more info, see

`$CEI_HOME/ensight92/src/readers/inventor`

Data Reader

`Main Menu > File > Data (reader)...`

The File Selection dialog is used to specify which files you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.iv` or `.civ`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with a suffix `.iv` or `.civ`) using the Advanced Interface method.

Data Tab	
Format	Use the **Inventor** format.
Set .iv file	Select the inventor file (typically **.iv** or **.civ**) and click this button
Format Options Tab	
One Part per file	Makes one part out of each file.

Console Output	Use this flag to determine the amount of output to the console. Normal - Usually only echo errors echoed to console. Verbose - Normal plus high level output describing dataset and progress while reading Debug - Detailed output and progress through the reader routines often valuable for understanding and reporting problems.
Set measured	Select the measured file and click this button.

(see How To Read Data).

LS-DYNA Reader

Overview

The LS-DYNA reader reads in a single or multiple unstructured C-binary d3plot files. It supports bars, quads, bricks and thick shell elements.

Key File for Part Naming

Can use of Key file to name parts as follows:

a. Works only if Material IDs are in the d3plot file
b. Put key file name into Params field
c. Looks for *PART keyword in keyfile
d. '$' in first column is a comment in keyfile
e. First non-comment line after *PART is used as partname if alpha or digit
f. Second non-comment line after *PART, 3rd integer is Material ID
g. If Mat'l ID in d3plot matches Mat'l ID in keyfile partname from key file is substituted for Material ID in EnSight name.

Limitations

The reader has the following limitations.

Doesn't support PACKED data (3 integers per word)

Skips over Smooth Particle Hydrodynamics Node data

Skips over Rigid Road Surface Data.

Skips over Computational Fluid Dynamics Data.

Does not read in LS-DYNA time-history plots

Coordinate System: Global vs. Local: Beam stresses and strains are always output in the local r,s,t system. Per LSTC manual, stresses and strains of the other elements are generally in the global system. However, shells & thick shells have an option to output in local system (see LS-DYNA 960 keyword users manual page 9.18. flag CMPFLG). The reader has no way of knowing whether stresses and strains are output in the global or local system and just shows the values contained in the files.

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/ls-dyna3d/README`

Data Reader

Simple Interface Data Load

Load your d3plot file using the Simple Interface method.

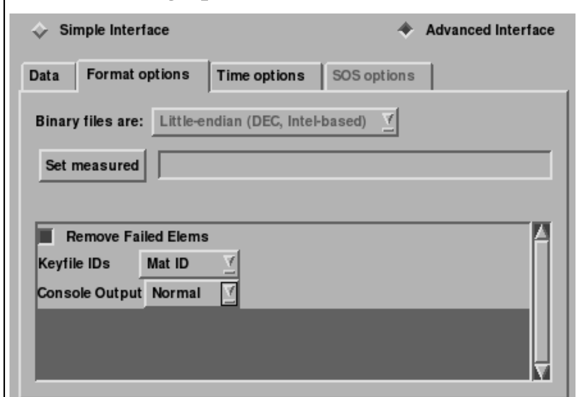
Advanced Interface Data Load

Load your d3plot file using the Advanced Interface method.

<table>
<tr><th colspan="2">Data Tab</th></tr>
<tr><td>Format</td><td>Use the LS-DYNA3D format.</td></tr>
<tr><td>Set d3plot</td><td>This field should have the first d3plot file name. All of the d3plot files will be loaded starting with this first one</td></tr>
<tr><td>Set key</td><td>This field can be used to import parameters that modify the behavior of the reader, or the Extra GUI section can be used to choose which ids to use for naming and to name the keyfile respectively.

keyfilename - Type the keyfile name into this field
<code>-mid - use material id in keyfile to name parts in d3plot file.</code>
<code>-pid - use part id in keyfile to name parts in d3plot file.</code>

example:
file.key -mid

This will use the material ids in keyfile named file.key to name parts.</td></tr>
<tr><th colspan="2">Format Options Tab</th></tr>
<tr><td>Set measured</td><td>Select the measured file and click this button.</td></tr>
<tr><td rowspan="4">Format Options</td><td>The following options are customized for the reader:


</td></tr>
<tr><td>Remove Failed Elems - Toggle on to remove failed elements</td></tr>
<tr><td>Keyfile IDs - This pulldown provides the choice of either Material IDs or Part IDs from the keyfile to be used for part naming. Alternatively, the ID can be specified in the (Set) Params field as described above.</td></tr>
<tr><td>Console Output - Can control amount of output that comes to the console. Options are: Normal, Verbose, or Debug</td></tr>
<tr><td></td><td>ASCII File Inpt - Can input glstat, abstat, matsum, rcforc, rwforc, nodout, secforc, sleout, elout xy ascii files.</td></tr>
</table>

(see How To Read Data)

Movie.BYU Reader

Reader Visibility Flag

By default, this reader is not loaded into the list of available readers. To enable this reader go into the Menu, Edit > Preferences and click on Data and toggle on the reader visibility flag.

Overview

Movie.BYU has a general n-sided polygon data format. In translating this format to the element-based EnSight data format, not all elements possible in the Movie.BYU format can be converted to EnSight format. However, for most practical cases there are no problems.

Movie.BYU data can have changing geometry. When this is the case, the changing geometry file name patterns are found in the results file. However, it is still necessary to specify an initial geometry file name in the (Set) Geometry field.

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.geo`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files (typically named with a suffix `.geo and .res`) using the Advanced Interface method.

Data Tab	
Format	Use the **Movie BYU** format.
Set geometry	Select the geometry file (typically **.geo**) and click this button
Set results	Select the results file (typically **.res**), and click this button.
Format Options Tab	
Set measured	Select the measured file and click this button.

Part Loader

The reader uses a simplified Part Loader as follows:

- All Parts: all parts are loaded to the client in the default visual representation (typically 3D Border, 2D Full).
- Part 1 Only: Only the first part is loaded to the client in the default visual representation. The other parts will have the NonVisual representation.
- All But Part 1: All parts other than part 1 are loaded to the client in the default visual representation. Part 1 will be NonVisual.
- No Parts: No parts are loaded to the client (i.e.the representation of all parts is set to NonVisual).

(see How To Read Data)

MPGS 4.1 Reader

Reader Visibility Flag

MPGS is considered a legacy format. Therefore, by default, this reader is not loaded into the list of available readers. To enable this reader go into the Menu, Edit > Preferences and click on Data and toggle on the reader visibility flag.

Overview

MPGS4.x uses a general n-sided polygon, n-faced polyhedral data format. In going from this format to the specific element data format of EnSight, you encounter the problem associated with translating from a general format to a specific format. Not all elements possible in MPGS4.x can be converted to EnSight format. However, there will not be a problem in most situations.

MPGS4.x models of modest size can be read directly into EnSight. Size can become an issue since the amount of memory needed to do the conversion in EnSight to the internal data format in a reasonable length of time can become excessive for large models.

MPGS4.x data can have changing geometry. When this is the case, the changing geometry file name patterns are contained in the results file. However, it is still necessary to specify an initial geometry file name in the (Set) Geometry field.

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.geo`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files (typically named with a suffix `.geo and .res`) using the Advanced Interface method.

Data Tab	
Format	Use the **MPGS 4.1** format.
Set geometry	Select the geometry file (typically **.geo**) and click this button. The MPGS Geometry file defines all geometric model Parts in a general n-sided polygon format.
Set results	Select the results file (typically **.res**), and click this button.
Format Options Tab	
Set measured	Select the measured file and click this button.

Part Loader

The MPGS reader uses a simplified Part Loader as follows:

- All Parts: all parts are loaded to the client in the default visual representation (typically 3D Border, 2D Full).
- Part 1 Only: Only the first part is loaded to the client in the default visual representation. The other parts will have the NonVisual representation.
- All But Part 1: All parts other than part 1 are loaded to the client in the default visual representation. Part 1 will be NonVisual.
- No Parts: No parts are loaded to the client (i.e.the representation of all parts is set to NonVisual).

(see How To Read Data)

MSC.DYTRAN Reader

Overview

Reads archive files ".ARC" directly, a modified case file, or a .dat file (which is the preferred method).

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/dytran/README`

Simple Interface Data Load

Load your dytran file (typically named with a suffix `.dat or .arc`) using the Simple Interface method.

Advanced Interface Data Load

Load your dytran file (typically named with a suffix `.dat or .arc`) using the Advanced Interface method.

Data Tab	
Format	Use the **MSC/Dytran** format.
Set dytran	This field should have the .dat or .arc suffix. Selecting any one file will read only that file. Inserting wild card characters, such as '*' and '?', will cause all files of a set to be read. For example, if I enter CONTAINER_EUL_*.ARC, the program will matches the pattern to all the files below.
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

MSC.MARC Reader

Overview

Reads a t16 or t19 file (which is the preferred method).

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/marc/README`

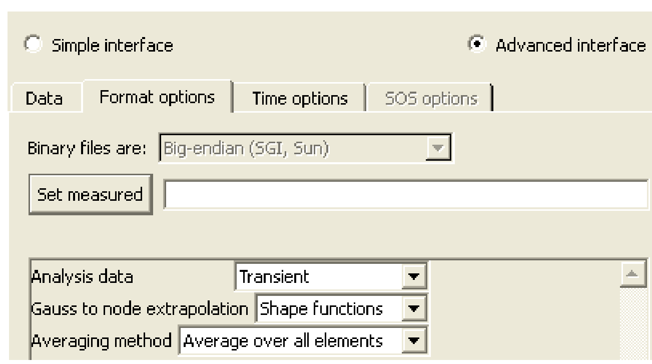
Simple Interface Data Load

Load your marc file (typically named with a suffix `.t16 or .t19`) using the Simple Interface method.

Advanced Interface Data Load

Load your marc file (typically named with a suffix `.t16 or .t19`) using the Advanced Interface method.

Table 2:

<table>
<tr><th colspan="2">Data Tab</th></tr>
<tr><td>Format</td><td>Use the MSC.Marc format.</td></tr>
<tr><td>Set t16/t19</td><td>This field should have the .t16 or .t19 suffix file.</td></tr>
<tr><th colspan="2">Format Options Tab</th></tr>
<tr><td>Format Options</td><td>


</td></tr>
<tr><td></td><td>Analysis data: Transient, Modal, Buckling. Please choose the type of data to be read. The "Transient" option supports any time-dependent result types (static, quasi-static and transient).</td></tr>
<tr><td></td><td>Gauss to node extrapolation: Shape functions. Currently only one choice.</td></tr>
<tr><td></td><td>Averaging method: Average over all elements. Currently only one choice.</td></tr>
<tr><td>Set measured</td><td>Select the measured file and click this button.</td></tr>
</table>

Features

1. Can read both t16 binary result files and t19 formatted result files
2. Supports most analysis types (structural, thermal, magnetic, etc.)
3. Supports results for nodes (scalar / vector) and gauss-points (scalar, vector and tensor). Gauss point results are extrapolated to nodes and averaged at the nodes.
4. Handles remeshing (global and local) and element activation / de-activation.

5. Can read results for DMP runs: If the main result file is selected, all domains are imported. Domain result files can be selected individually

6. Can read static / transient results, modal results and buckling results.

7. Works on all Marc versions from Marc 2000 up to 2005r2.

8. Little-endian and Big-endian results are handled transparently. A message "Not a native format file. May not work correctly!" will appear if the reader suspects it is not in the native machine format, and automatic conversion will start.

9. For Buckling and Modal results, a variable called "Load_Factor" or "Frequency" is created respectively that gives the mode frequency or buckling load factor.

10. Ensight does not support the ability for the time to move backwards as the Arc-length methods will make happen. When the reader detects time that moves backwards at any point in the results file, the times will be reset to be 0., 1., 2. etc., and an extra variable "Time" will be created that contains the actual time.

11. Reader supplies Stress as a Tensor (when available in the data)

a. To calculate Principal Stress, use EnSight's TensorEigenvalue calculator function

b. To calculate VonMises Stress, use EnSight's TensorVonMises calc function

Limitations

1. Cannot read pre Marc 2000 results.

2. Rigid contact bodies and their results are not read.

3. Flow line data is not used.

4. Springs and Tyings are not used.

5. Only time-dependent results (static and / or transient), modal or buckling results can be read at a time. If more than one of these exist in a single result file, only the one selected on the options form will be read.

6. No Mentat sets (or Patran groups) are imported.

7. The t19 (formatted) results files read much slower than t16 (binary) result files. It is faster to convert a t19 file to t16 (by using the "pldump2000" executable that is always installed with Marc) than read t19 files directly in Ensight.

(see How To Read Data)

MSC.NASTRAN Reader

Overview

Reads .op2 suffix files including most PDA Patran (PARAM POST = -1) and SDRC I-DEAS (PARAM POST = -2) files.

Limitations

a) Binary format only. (If you need to read ASCII, convert using the Nastran utility that will do this.)

b) Some non-linear and composite element types have not yet been implemented.

Recent Enhancements

1. Extra GUI options were added to allow the user control over part creation and variable extraction.

2. Location and displacement coordinate systems are now recognized and applied.

3. Multiple op2 files can be read by the reader, and thus will appear in the same EnSight case. This is controlled by a simple ascii file (.mop file).

   The format of the .mop file is:
   ```
   ----------------------------------
   line 1:          The word mop, in quotes
   line 2:          The number of files
   line 3 and up:   Each op2 filename, in quotes
   ```

   example:
   ```
   ----------
   mop
   3
   “boom.op2”
   “bucket.op2”
   “hframe_side.op2”
   ```

   NOTE: The .mop file extension has been added to the Nastran reader section of the ensight_reader_extension.map file (in site_preferences).

4. Rigid Body euler parameters are being read and passed to EnSight. (EnSight has also been modified to apply these rigid body parameters to the geometry and vector variables.) The specification of the rigid body file and the registration of which parameters apply to what - is also done in the .mop format.

   The format of the .mop file, with rigid body information as well, is:
   ```
   ------------------------------------------------------------------------------
   line 1:          The word mop, in quotes
   line 2:          The number of files
   line 3 and up:  Each op2 filename, the euler parameter filename, the title of the
                   rigid body transformation in the euler parameter file that apply to
                   this .op2 file, and a unit conversion scale factor (if needed).  All
                   on one line per file, and all in quotes.
   ```

   example:
   ```
   ----------
   mop
   3
   “boom.op2”      “rigid.eet”  “BOOM”    “1000.0”
   “bucket.op2”    “rigid.eet”  “BUCKET”  “1000.0”
   ```

“hframe_side.op2” “motion.eet” “HFRAME” “1000.0”

NOTE: Since an euler parameter file contains the transformation information for many different “parts”, the same file will generally be indicated for each .op2 file. However, this can be a different file for each .op2 file.

Also, the last column is not required - but is provided in the case that unit conversion is needed between the .op2 system and the euler parameter system. In our example, the .op2 system was in millimeters, while the translations values in the euler parameter file were given in meters.

NOTE: If there is an additional offset to the CG that is needed (other than that specified in the euler parameter file), these offsets can also be placed in the .mop file. Simply add three more columns containing the x, y, z offsets, like the following:

```
example:
----------
mop
3
“boom.op2”        “rigid.eet”  “BOOM”    “1000.0”  “883.7”  “207.4”  “0.0”
“bucket.op2”      “rigid.eet”  “BUCKET”  “1000.0”  “-10.5”  “67.2”   “7.89”
“hframe_side.op2” “rigid.eet”  “HFRAME”  “1000.0”  “367.5”  “-12.45” “0.0”
```

5. You can also add a rotation order and yaw, pitch, and roll values on each of the file lines if the coordinate system needs to be re-oriented. These additional columns follow the same format as those in the EnSight Rigid Body (.erb) file. (see Section 11.13, EnSight Rigid Body File Format)

6. The reader deals with timelines and needed interpolations between them. Generally, EnSight readers need only provide data at the given timesteps of a model. EnSight takes care of getting both ends of a time span and interpolating between them if needed. However, if rigid body motion is provided, the controlling timeline will be the rigid body timeline. Thus, for a given rigid body timestep, we may fall between timesteps for the nastran model. This reader can interpolate properly for this situation.

   Also, if not using rigid body, but are using multiple files - with different timelines - a combined timeline will be created and sent to EnSight. This also can require interpolation within the different files - and this is handled as well.

7. How variables are handled has been completely redone. The old reader simply presented the values of whatever was in the file. This lead to many different variables, depending especially on which element types were used. It also did not assure that some of the standard variables were available.

   This reader now presents a standard list of the component and principal stresses/strains, and the useful failure theories. These values are obtained either by reading them from the file (if provided), or computing them from the data that is provided. We believe it is It is much more friendly and useful!.

8. Because of the way that element variable values now may lead to nodal variables - requiring averaging, and because the way data is stored in a .op2 file is not always conducive to being used efficiently by EnSight - various caching schemes have been implemented to attempt to improve the efficiency of the reader. Hopefully appropriate trade-offs between memory and speed have been

utilized. As such, it should be pointed out that one can color all parts by a variable about as quickly as coloring only one.

9. Static models with multiple loadcases use the Solution Time dialog to switch between loadcases. Thus, a “change of timestep” in EnSight will actually change between loadcases.

   NOTE: The preference within EnSight to have the Color Palette update at each time step is especially nice to have set for this situation.

README

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/nastran/README.txt`

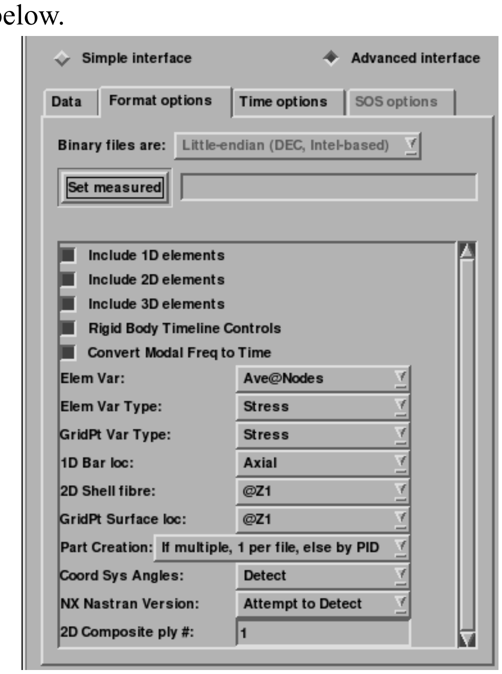
Simple Interface Data Load

Load your geometry/results file (typically named with a suffix `.op2`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry/results file (typically named with a suffix `.op2`) using the Advanced Interface method.

<table>
<tr><th colspan="2">Data Tab</th></tr>
<tr><td>Format</td><td>Use the Nastran OP2 format.</td></tr>
<tr><td>Set op2</td><td>Enter the .op2 filename if reading a single NASTRAN .op2 file, or a .mop filename if reading multiple .op2 files. The .mop file is an ASCII file listing .op2 filenames. See the description above or the README file indicated above for more details.</td></tr>
<tr><th colspan="2">Format Options Tab</th></tr>
<tr><td>Set measured</td><td>Select the measured file and click this button.</td></tr>
<tr><td>Extra GUI</td><td>The following parameters are available. They are described below.



</td></tr>
</table>

Extra GUI Parameters

The toggles and fields below are customized for the Nastran OP2 reader. They allow the user to specify basic options before the data is read. They may not all apply to any given model.

Include 1D elements

Toggle on to include any 1D (bar, rod, etc.) elements.

Include 2D elements

Toggle on to include any 2D (tri, quad, etc.) elements.

Include 3D elements

Toggle on to include any 3D (tet, hex, etc.) elements.

Rigid Body Timeline Controls

Toggle on to have the geometry timeline controlled by the rigid body times (if present). If off, the flex body times in the .op2 file will control.

Convert Modal Freq to Time

Toggle on to compute solution time from the "frequency" field in the file for eigen analysis (default is on). The is done with Time = (sqrt(freguency))/(2*PI). Toggle off to no perform this computation and instead read the "frequency" value as the time.

Elem Var:

This pulldown provides control over the way Nastran element values will be presented as variables within EnSight.

Centroidal	produces per_elem variables from the value at each element centroid
Ave@Nodes	produces per_node variables, by averaging all vertex values at a given node
Max@Nodes	produces per_node variables, by taking the maximum vertex value at a given node
Min@Nodes	produces per_node variables, by taking the minimum vertex value at a given node

Elem Var Type:

This pulldown provides the choice of extracting either Strain or Stress from Nastran element values.

GridPt Var Type:

This pulldown provides the choice of extracting either Strain or Stress from Nastran Grid Point values.

1D Bar loc:

This pulldown provides the choice of where along the bar (EndA, EndB) or across the cross-section (Pts 1-4), and what type of stress or strain to extract from Nastran bar elements.

Axial	
Bend, EndA, Pt1	Combined, EndA, Pt1
Bend, EndA, Pt2	Combined, EndA, Pt2
Bend, EndA, Pt3	Combined, EndA, Pt3
Bend, EndA, Pt4	Combined, EndA, Pt4
Bend, EndB, Pt1	Combined, EndB, Pt1
Bend, EndB, Pt2	Combined, EndB, Pt2
Bend, EndB, Pt3	Combined, EndB, Pt3
Bend, EndB, Pt4	Combined, EndB, Pt4

2D Shell Fibre: This pulldown provides the choice of which cross-sectional fibre (@Z1 or @Z2) to extract from Nastran 2D Shell elements.

GridPt Surface loc: This pulldown provides the choice of which cross-sectional fibre (@Z1, @Z2, or @MID) to extract from Nastran 2D Grid Point surfaces.

Part Creation: This pulldown provides part creation choices (which are most useful when a .mop file is used to bring in multiple OP2 files together:

One Per File	One part for each file will be created. If a single OP2 file is being read, all elements will be placed in a single part. If multiple OP2 files are being read, one part per file will be produced.
By Property id	Parts will be created by property id. According to how property ids were used in the OP2 file, this will generally create several parts per file.

NX Nastran Version: This pulldown provides the user with some control over the changes that occured in the element record length at NX Nastran version 4.0. This is needed because there is not a good way to determine the version used from the .op2 file itself.

Attempt to Detect	Attempts to divine the version number, but may not always work correctly. If it can't tell, will default to less than version 4.
Declare as >= 4.0	Declares the version to be 4.0 or greater, so doesn't go through the detection process.

2D Composite ply: This field allows the user to specify from which ply number to extract values from Nastran 2D composite elements

(see How To Read Data)

Nastran Input Deck Reader

Overview

Description

This reader will load Nastran input deck or bulk data files (typically .nas, .bdf, .dat). These files contain the geometry for a Nastran run.

Usefulness

Being able to read this format allows for the display of the original Nastran geometry for verification as well as for use with rigid body motion.

Usage

The Nastran input deck reader can read in and individual .nas/.dat/.bdf file, or it can read in an exec file so that more than one file can be included in the same case.

Limitations

The current reader does not deal with local coordinate systems and only recognizes the following elements:

1D Elements	2D Elements	3D Elements
CBAR	CTRIA	CTETRA
CBEAM	CTRIAR	CPENTA
CROD	CTRIA6	CHEXA
CGAP	CTRIAX	
CTUBE	CTRIA6X	
CVISC	CQUAD	
CONROD	CQUAD4	
PLOTEL	CQUADR	
RROD	CQUAD8	
RBAR	CQUADX	
CELAS1	CSHEAR	
CELAS2		

Simple Exec file format

An exec file is used to read in multiple Nastran input deck files into one case. This exec file is a very simple ascii file that must conform to the following:

1. All lines must begin in column 1
2. No blank or comment lines allowed
3. If the stl filenames begin with a "/", it will be treated as absolute path. Otherwise, the path for the exec file will be prepended to the name given in the file. (Thus, relative paths should work).

```
line 0:     numfiles: N    (where N is the no. of files)
[line 1:    version #]     (optional line containing the
                            version number)
next N lines:  nasfilename1
               . . .
               . . .
               nasfilenameN
```

Example Simple Exec file (without version number)

```
numfiles: 3
CASTLE.DAT
bincastle.bdf
test.nas
```

Example Simple Exec file (with version number)

```
numfiles: 3
version 1.1
CASTLE.DAT
bincastle.bdf
test.nas
```

Rigid Body Motion Exec file

The reader includes the capability to link each input deck file with a rigid body transformation file to allow the parts in each file to rigidly translate and rotate over time. The rigid body motion Exec file has additional columns that contain the Euler Parameter filename (see Section 11.14, Euler Parameter File Format), the transformation title in the Euler Parameter file, and a units scale factor. The rigid body version of this Exec file requires quotes as shown around the strings and values of the file lines.

example:

```
numfiles: 3
"CASTLE.DAT"      "motion.dat" "CASTLE"  "1000.0"
"bincastle.bdf"   "motion.dat" "BCASTLE" "1000.0"
"test.nas"        "motion.dat" "TEST"    "1000.0"
```

And if an additional offset is needed to the CG, add these in 3 more columns

example:

```
numfiles: 3
"CASTLE.DAT"    "motion.dat" "CASTLE"  "1000.0" "1.35"  "2.66" "0.0"
"bincastle.bdf" "motion.dat" "BCASTLE" "1000.0" "-2.45" "1.0"  "-2.0"
"test.nas"      "motion.dat" "TEST"    "1000.0" "60.2"  "23.4" "0.0"
```

You can also add a rotation order and yaw, pitch, and roll values on each of the file lines if the coordinate system needs to be re-oriented. These additional columns follow the same format as those in the EnSight Rigid Body (.erb) file.
(see Section 11.13, EnSight Rigid Body File Format)

README

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/nas_input/README.txt`

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.nas, .bdf, or .dat`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry file (typically named with a suffix `.nas, .bdf, or .dat`) using the Advanced Interface method.

Data Tab	
Format	Use the **Nastran Input Deck** format.
Set geometry	Select the geometry file (typically **.nas, .bdf, or .dat**) and click this button
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

N3S Reader

Reader Visibility Flag

By default, this reader is not loaded into the list of available readers. To enable this reader go into the Menu, Edit > Preferences and click on Data and toggle on the reader visibility flag.

Overview

Description

N3S is a data format developed by Electricité de France (EDF) consisting of a geometry file and a results file. For this data format, both files are always required. Versions 3.0 and 3.1 are both supported.

When reading N3S data into EnSight, you extract Parts from the mesh interactively based on different color numbers or boundary conditions. The available color numbers and boundary conditions for the model are presented.

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.geo`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files (typically named with a suffix `.geo and .res`) using the Advanced Interface method.

Data Tab	
Format	Use the **N3S**format.
Set geometry	Select the geometry file (typically **.geo**) and click this button
Set results	Select the results file (typically **.res**), and click this button.
Format Options Tab	
Set measured	Select the measured file and click this button.

N3S Part Creator dialog

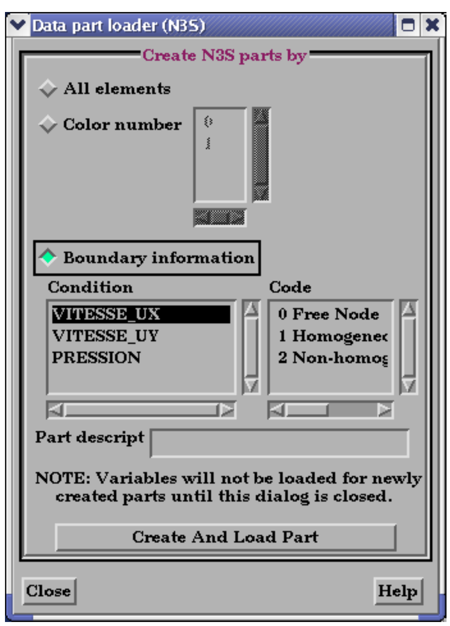


Figure 2-12
N3S Part Creator dialog

You use the File Selection dialog to read in N3S dataset files. You use the N3S Part

Creator dialog to extract Parts from a N3S dataset.
`Access: Main Menu > File > Data (Reader)...> N3S`

Create N3S Parts By

All Elements — Selection to create a Part using all of the elements available within the data file.

Color Number — Selection to create a Part according to the color number associated with each element.

Boundary Information — Selection to create a Part according to specified conditions and codes.

Condition — Select boundary condition to use for Part creation.

Code — Select Code to use for boundary condition.

Part Descript — Specify name for Part.

Create Part — Click to create a Part. The Part is listed in the main Parts list of the Parts & Frames dialog and is displayed in the Main View window.

(see How To Read Data)

OpenFOAM Reader

Overview

Description

Reads OpenFOAM controlDict file found in modelname/system/controlDict.

Data Reader

`Main Menu > File > Data (reader)...`

The File Selection dialog is used to specify which file you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Handles steady state geometry with either steady state or transient variables. Steady state variables with multiple iterations will use each iteration as an EnSight timestep.

Not yet supported:
a. Changing coordinates or changing connectivity. Utilize the "foamToEnsight" translation routine from OpenFOAM to convert the OpenFOAM data into EnSight Case Gold format.
b. Parallel files. Reconstruct the parallel files into a single, serial file using the OpenFOAM utility 'reconstructPar' OpenFOAM.
c. Ongoing solution. The reader cannot handle newly available timesteps or iterations (for example from an ongoing OpenFOAM solution) after the model has been read into EnSight the first time. Should new iteration or timesteps become available after the model was originally read into EnSight, the user must reload the dataset.

Command Line Data Load

To automatically start EnSight and load the current directory's OpenFOAM dataset from the command line, type 'ensight92 -Eensfoam'. This will trigger EnSight to start up, look for the current directory's "system/controlDict" file, and automatically load the dataset into EnSight (using the default reader settings). This reduces the number of steps to load the file into EnSight and thus the real time required to load the data, and provides a level of integration with this data format.

Sample Data

A sample OpenFOAM dataset is included as a sample session with your install. To access the welcome screen, at the top menu choose Window>Welcome To... and load the Dam Break example session. Or, to load the same dataset manually, find the controlDict file in $CEI_HOME/ensight/other_data/openfoam .

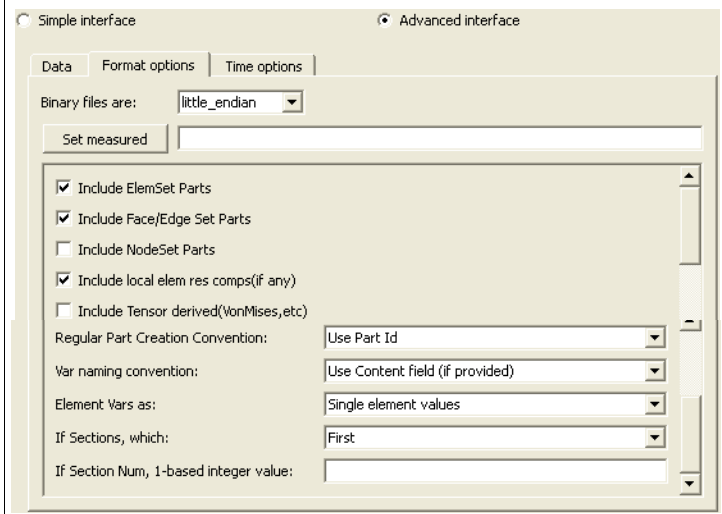
Simple Interface Data Load

Load your OpenFOAM controlDict file using the Simple Interface method. Or from the command line, simply run ensfoam.

Advanced Interface Data Load

Load your OpenFOAM file using the Advanced Interface method.

Data Tab	
Format	Use the **OpenFOAM** format.
Set file	This field contains the controlDict file. Clicking button inserts the file name shown into the field.
Format Options Tab	
Set measured	Select the measured file and click this button.

Other Options	
Include ElemSet Parts	Include any Element sets defined. These are sets of full elements which are generally some logical subset of the total number of elements. Default is on.
Include Face/ Edge Parts	Include any Face or Edge sets defined. These are some logical set of particular faces and/or edges of full elements. Default is on.
Include NodeSet Parts	Include any Node sets defined. These are generally the subset of nodes needed for the Element, Face, or Edge sets above. As such, they are generally not needed as separate parts, but can be created if desired. Default is off.
Include local elem res comps (if any)	Include the local stresses components, etc that are in the elemen Default is on
Include Tensor derived (VonMises, etc.)	For tensor results, calculate scalars from the tensors. Default is off
Regular Part Creation Convention	Parts will be created according to the following: Use Part Id - Part Id (this is the default) Use Property Id - Property Id Use Material Id - Material Id

Var naming convention	Use DataSource field - By default variables are named using the variable filename. For example, "U" is velocity, "p" is pressure, etc. Use Content Field (if provided) - Known variables are given full, meaningful names, for example, "Velocity" or "Pressure". Use VKI dataset name - Long, hybrid variable name that is guaranteed to be unique, but perhaps cryptic.
Element Vars as	Single element values - Element results (whether centroidal or element nodal) will be presented as a single value per element. Thus will be per_elem variables in EnSight.This is the default. Averaged to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes without using geometry weighting. Thus will be per_node variables in EnSight. Geom weighted average to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes using geometry weighting. Thus will be per_node variables in EnSight
If Sections, which:	Not used
Section Num	Not used

(see How To Read Data)

OVERFLOW Reader

Simple Interface Data Load

Load your geometry file using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files using the Advanced Interface method.

Data Tab	
Format	Use the **OVERFLOW** format.
Set geometry	Select the grid file (grid.in or single 'x.' file, e.g 'x.14200', see details below) and click this button. This file is a structured GRID file with FAST enhancements.
Set results	Select the results file and click this button. The Results File is either a modified EnSight Results file (q.res) or standard plot3d Q-file (q.save or single 'q.' file, e.g. 'q.14200'). The standard plot3d Q-file is a variable file for a single timestep and is optional. The modified EnSight results file directs the reader to handle multiple grid (x.) files and/or multiple variable (q.) files from transient simulations. If using a .res file, then enter only the first 'x.' file into the set geometry field, and the '.res' into the set results field. Note: The Q-file(s) (and result file) may be located in a different directory than the grid file.
Format Options Tab	
Set bounds	The optional boundary file defines boundary portions within and/or across structured blocks. (Note: this can be EnSight's boundary file format or a .fvbnd file.)
Set measured	Select the measured file and click this button.

Extra Information

OVERFLOW reader extracts structured parts and iblanked unstructured parts essentially the same way as PLOT3D, and as described in the basic structured reader (see Reading and Loading Data Basics).

Additionally to successfully read OVERFLOW data, the following information must be known about the data:

1. format - Fortran binary
2. whether single or multizone
3. dimension - 3D, 2D, or 1D
4. whether iblanked or not

5. precision - single or double

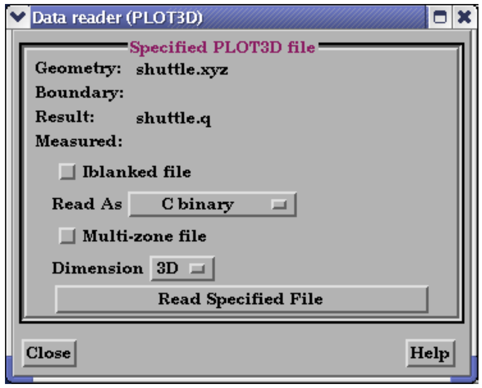


Figure 2-13

Part Data Loader (PLOT3D) used for OVERFLOW

EnSight attempts to determine these five settings automatically from the grid file. The settings that were determined (for the first four) are shown in the Part Builder dialog, where you can override them manually if needed. Note that OVERFLOW data is restricted to only FORTRAN Binary format.

The precision setting is not reflected in the dialog, but is echoed in the Server shell window. The q file precision will by default be set the same as that of the grid file. In the rare case where the automatic detection is wrong for the grid file or the precision is different for the q file than for the grid file, commands can be entered into the Command dialog to manually set the precision.

test:	`plot3d_grid_single`	to read grid file as single precision
test:	`plot3d_grid_double`	to read grid file as double precision
test:	`plot3d_qr_single`	to read q (or function) file as single precision
test:	`plot3d_qr_double`	to read q (or function) file as double precision

Limitation

In order to automatically recognize the data as Overflow, the files must have the format ‘`x.`’ and ‘`q.`’ format (For example `x.14000`, `x.14200`, `q.14000`, and `q.14200` would be appropriate filenames).

Note that the overflow reader can read in transient geometry files but these files must have the same number of zones (EnSight parts) at each timestep. Each zone can change in size (changing connectivity), but the total number of zones must remain constant throughout time.

Example .res file

For example, if you have files `x.14400` to `x.16000` and `q.14400` to `q.16000`, then an example `q.res` file would be as follows. Then, put `x.14400` into the set geometry and `q.res` into set results field and you will have transient geometry and variables.

```
2 1 1
10
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
14400 200
x.*****
q.***** S 1 Density
q.***** S 5 Energy
q.***** S 2 3 4 Momentum
```

OVERFLOW Q FIle Variables

The variables of the flow information read by the OVERFLOW reader basically conforms to those read by the PLOT3D reader, and includes additional flow constants as well as additional Q variables such as γ and possible turbulance field and species densities variables.

The 'Constant Variables' include (where the first 4 are the standard PLOT3D constants):

FSMACH	=	freestream Mach number M_{inf}
ALPHA	=	angle-of-attack α
RE	=	Reynolds number Re
TIME	=	iteration (file) number (in OVERFLOW; in PLOT3D, time value)
GAMinf	=	freestream gamma γ_{inf}
BETA	=	sideslip angle β
Tinf	=	freestream temperature T_{inf} (in degrees Rankine)
IGAM	=	variable gamma option where: 0 = use constant γ value of GAMinf 1 = Single gas with variation of γ with temperature computed using LT_A0-4, UT_A0-4 below 2 = Two gases, with variation of γ with temperature computed using LT_A0-4, UT_A0-4 below all gas 1 below HT1, all gas 2 above HT2, linear mix in between.
HTinf	=	freestream stagnation enthalpy $h_0{}^*{}_{inf}$
RefMACH	=	reference mach number (Note: in OVERFLOW “restart” files only)
Tvref	=	actual simulation time (Note: in OVERFLOW “restart” files only)
DTvref	=	delta simulation time (Note: in OVERFLOW “restart” files only)
RGAS1	=	species gas constant 1
RGAS1_SMW	=	species gas constant 2

The 'Q-Field Scalars' include (where the first 4 are the standard PLOT3D Q-variables):

Density	=	Q1-field variable = dimensionless density, ρ^*
Momentum	=	dimensionless momentum vector with: Momentum[X] = Q2-field variable = x component of Momentum $\rho^* u^*$ Momentum[Y] = Q3-field variable = y component of Momentum $\rho^* v^*$ Momentum[Z] = Q4-field variable = z component of Momentum $\rho^* w^*$

Energy	=	Q5-field variable = dimensionless total energy $\rho^* e_0^*$
Gamma_Q6_field	=	Q6-field variable = gamma γ (constant field, unless you use the gamma option of the code)

And for SA model:

Q7_field	=	Q7-field variable = turbulence variable

And for k-e model:

Q7_field, Q8_field	=	Q7-field and Q8-field variables which are the k and epsilons

Assigning Analysis_Time

By default, the Analysis_Time constant variable value is assigned the time values listed in the q.res file. (see Section 11.7, PLOT3D Results File Format). In order to use the TIME (or Tvref - if using an OVERFLOW restart q.file) value located in the header of the q-file(s), edit the q.res file:
a) change the total number of time steps to a negative value, and
b) remove the list of time values in the q.res file.

(see How To Read Data, and Section 11.7, PLOT3D Results File Format)

PLOT3D Reader

Example Source

See the following directory for an example User-defined source code implementation this reader.

`$CEI_HOME/ensight92/src/readers/plot3d/`

Simple Interface Data Load

Load your geometry file using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files using the Advanced Interface method.

Data Tab	
Format	Use the **PLOT3D** format.
Set geometry	Select the grid file and click this button. This file is a structured GRID file with FAST enhancements.
Set results	Select the results file and click this button. The Results File is either a modified EnSight Results file or standard plot3d Q-file. Variable files (optional) are solution (PLOT3D Q-files) or function (FAST) files. The modified EnSight results file provides access to multiple solution files that are produced by time dependent simulations.
Format Options Tab	
Set bounds	The optional boundary file defines boundary portions within and/or across structured blocks. (Note: this can be EnSight's boundary file format or a .fvbnd file.)
Set measured	Select the measured file and click this button.

Extra Information

PLOT3D reader extracts structured parts and iblanked unstructured parts essentially the same way as described in the basic structured reader (see Reading and Loading Data Basics).

Additionally to successfully read PLOT3D data, the following information must be known about the data:

1. format - ASCII, C binary, or Fortran binary
2. whether single or multizone
3. dimension - 3D, 2D, or 1D
4. whether iblanked or not
5. precision - single or double

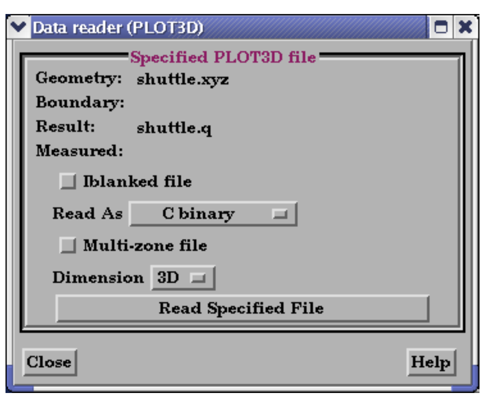


Figure 2-14
Part Data Loader (PLOT3D)

EnSight attempts to determine these five settings automatically from the grid file. The settings that were determined (for the first four) are shown in the Part Builder dialog, where you can override them manually if needed.

The precision setting is not reflected in the dialog, but is echoed in the Server shell window. The q (or function) file precision will by default be set the same as that of the grid file. In the rare case where the automatic detection is wrong for the grid file or the precision is different for the q (or function) file than for the grid file, commands can be entered into the Command dialog to manually set the precision.

test:	`plot3d_grid_single`	to read grid file as single precision
test:	`plot3d_grid_double`	to read grid file as double precision
test:	`plot3d_qr_single`	to read q (or function) file as single precision
test:	`plot3d_qr_double`	to read q (or function) file as double precision

Assigning Analysis_Time

By default, the Analysis_Time constant variable value is assigned the time values listed in the q.res file. (see Section 11.7, PLOT3D Results File Format). In order to use the TIME value located in the header of the q-file(s), edit the q.res file:
a) change the total number of time steps to a negative value, and
b) remove the list of time values in the q.res file.

(see How To Read Data)

RADIOSS Reader

Overview

Description Reads Radioss 4.x ANIM files.

Data Reader

`Main Menu > File > Data (reader)...`

The File Selection dialog is used to specify which files you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Simple Interface Data Load Load your radioss file using the Simple Interface method.

Advanced Interface Data Load Load your radioss file using the Advanced Interface method.

Data Tab	
Format	Use the **RADIOSS_4.x** format.
Set anim	This field contains the first Radios file name in the series. Clicking button inserts file name shown into the field. File name can then be modified with an asterisk “*” or question mark “??” to indicate the unique identifiers in the file series.
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

POLYFLOW Reader

Overview

Description Reads Polyflow .msh and .res files.

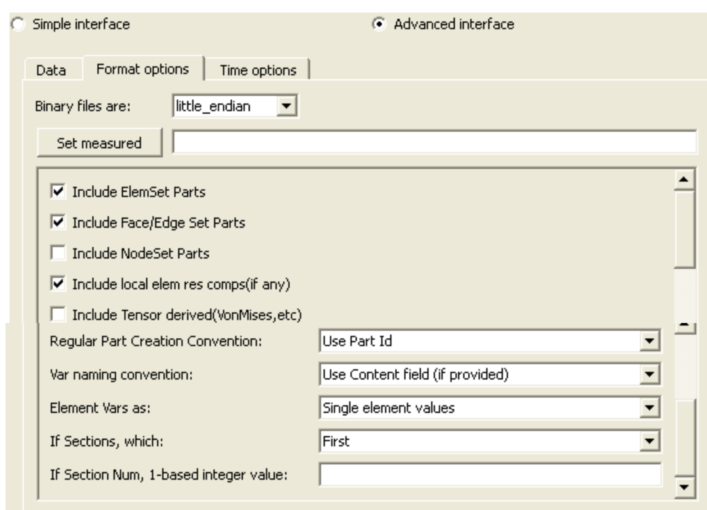
Data Reader

`Main Menu > File > Data (reader)...`

The File Selection dialog is used to specify which files you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Simple Interface Data Load Load your Polyflow file using the Simple Interface method.

Advanced Interface Data Load Load your Polyflow file using the Advanced Interface method.

<table>
<tr><td colspan="2">Data Tab</td></tr>
<tr><td>Format</td><td>Use the Polyflow format.</td></tr>
<tr><td>Set .msh</td><td>This field contains the mesh file. Clicking button inserts .msh file name shown into the field.</td></tr>
<tr><td>Set .res</td><td>This field contains the result file.</td></tr>
<tr><td colspan="2">Format Options Tab</td></tr>
<tr><td>Set measured</td><td>Select the measured file and click this button.</td></tr>
<tr><td>Other Options</td><td>
</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Include ElemSet Parts</td><td>Include any Element sets defined. These are sets of full elements which are generally some logical subset of the total number of elements. Default is on.</td></tr>
</table>

Include Face/ Edge Parts	Include any Face or Edge sets defined. These are some logical set of particular faces and/or edges of full elements. Default is on.
Include NodeSet Parts	Include any Node sets defined. These are generally the subset of nodes needed for the Element, Face, or Edge sets above. As such, they are generally not needed as separate parts, but can be created if desired. Default is off.
Include local elem res comps (if any)	Include the local stresses components, etc that are in the elemen Default is on
Include Tensor derived (VonMises, etc.)	For tensor results, calculate scalars from the tensors. Default is off
Regular Part Creation Convention	Parts will be created according to the following: Use Part Id - Part Id (this is the default) Use Property Id - Property Id Use Material Id - Material Id
Var naming convention	Use Content Field (if provided) - Variable names will be what is in the Content field, if provided. If not provided, they will be the VKI dataset name. This is the default. Use VKI dataset name - Variable names will be the VKI variable dataset name (which are reasonably descriptive).
Element Vars as	Single element values - Element results (whether centroidal or element nodal) will be presented as a single value per element. Thus will be per_elem variables in EnSight.This is the default. Averaged to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes without using geometry weighting. Thus will be per_node variables in EnSight. Geom weighted average to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes using geometry weighting. Thus will be per_node variables in EnSight
If Sections, which:	Not used
Section Num	Not used

(see How To Read Data)

SDRC Ideas Reader

Overview

Description

Reads SDRC/Ideas Universal files in Ascii and Binary format. Select the .unv file and select "SDRC/Ideas" in the format pulldown on the Advanced reader tab.

Data Reader

`Main Menu > File > Data (reader)...`

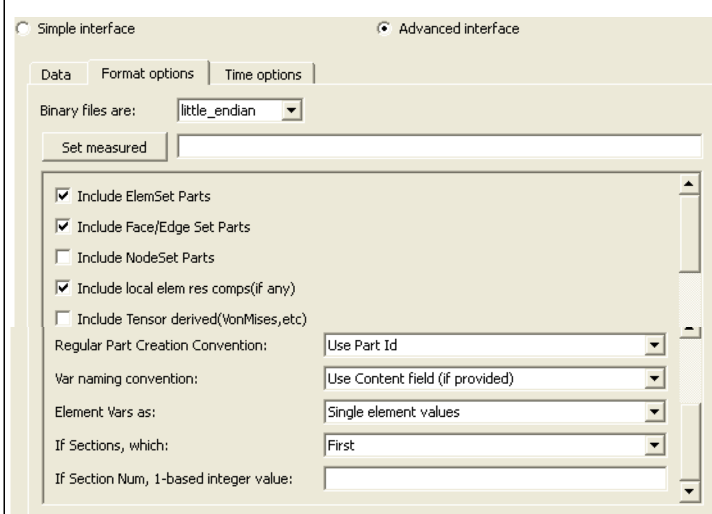
The File Selection dialog is used to specify which files you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Simple Interface Data Load

You cannot use the Simple Interface method to load your SDRC Ideas data because the .unv extension is used by other formats.

Advanced Interface Data Load

You must load your SDRC Ideas .unv file using the Advanced Interface method.

Data Tab	
Format	Use the **SDRC Ideas** format.
Set file	This field contains the first file name. For the first file you should choose a file with extension .unv. Clicking button inserts file name shown into the field. Loading the .unv file will load both geometry and results.
Format Options Tab	
Set measured	Select the measured file and click this button.
Other Options	
Include ElemSet Parts	Include any Element sets defined. These are sets of full elements which are generally some logical subset of the total number of elements. Default is on.

Include Face/ Edge Parts	Include any Face or Edge sets defined. These are some logical set of particular faces and/or edges of full elements. Default is on.
Include NodeSet Parts	Include any Node sets defined. These are generally the subset of nodes needed for the Element, Face, or Edge sets above. As such, they are generally not needed as separate parts, but can be created if desired. Default is off.
Include local elem res comps (if any)	Include the local stresses components, etc that are in the element's local system. A simple example is a bar (such as a truss element), which only has tension or compression in the element's axial orientation. Such an element would have an axial stress variable. Other elements would have appropriate result component variables. Default is on
Include Tensor derived (VonMises, etc.)	For tensor results, calculate scalars from the following derived results (principal stress/strains, and common failure theories): Mean VonMises Octahedral Intensity Max Shear Equal Direct Min Principal Mid Principal Max Principal By default, all 9 of these will be derived. You can control which are created by this toggle, with an environment variable. Namely, `setenv ENSIGHT_VKI_DERIVED_FROM_TENSOR_FLAG n` where n = 1for Mean only 2 for VonMises only 4 for Octahedral only 8 for Intensity only 16 for Max Shear only 32 for Equal Direct only 64 for Min Principal only 128 for Mid Principal only 256 for Max Principal only 512 for all or any legal combination. example: for VonMises and Max Shear only, use 18. Default is off

Regular Part Creation Convention	Parts will be created according to the following: Use Part Id - Part Id (this is the default) Use Property Id - Property Id Use Material Id - Material Id
Var naming convention	Use Content Field (if provided) - Variable names will be what is in the Content field, if provided. If not provided, they will be the VKI dataset name. This is the default. Use VKI dataset nameVariable names will be the VKI variable dataset name (which are reasonably descriptive).
Element Vars as	Single element values - Element results (whether centroidal or element nodal) will be presented as a single value per element. Thus will be per_elem variables in EnSight.This is the default. Averaged to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes without using geometry weighting. Thus will be per_node variables in EnSight. Geom weighted average to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes using geometry weighting. Thus will be per_node variables in EnSight
If Sections, which:	Which section will be used to create the variable First - The first section will be used (this is the default) Last - The last section will be used Section Num (below) - The section number entered in the field below will be used Separate Vars per Section - A separate variable will be created for each section.
Section Num	If the previous option is chosen to be Section Num, then the value in this field is the 1-based section number to use to create the variable.

(see How To Read Data)

SILO Reader

Overview

Description

The Silo reader can read .silo files directly or can read them using a Casefile which lists the geometry variable filenames, the timesteps and the constants all in one ASCII file. The .silo file contains both the geometry and the results.

Library

The Silo reader requires the Silo library version 4.2 or later. For information on Silo please see the following website:

http://www.llnl.gov/bdiv/meshtv/software.html

SILO Casefile format

The User Defined SILO Reader reads a restricted version of the EnSight Gold ASCII casefile as described below.

```
1. FORMAT
   type:  - "silo"  required
2. GEOMETRY
   model:
3. VARIABLE
   constant per case:
4. TIME    (But only one of these!!!!)
     number of steps:  - required
     time values:      - required
     # Use the following if transient and
     #   evenly spaced values
     filename start number:
     filename increment:
     # Use the following if transient and
     #   list all values
     filename numbers:
 5. All commands and options must start in first
     column. However, A space, newline, or # can
    be used in first column to indicate a comment
     line.
```

The following examples could be read by the user defined ensight gold reader

```
Example 1: A static model
-------------------------
FORMAT
type:            silo
GEOMETRY
model:           example1.silo
VARIABLE
constant per case:      Density      .5
TIME
number of steps:        1
time values:            0.0
```

The following files would be needed for Example 1:

example1.silo

```
Example 2: A transient model
----------------------------
FORMAT
type:           silo
GEOMETRY
model:          example2.*.silo    change_coords_only
VARIABLE
constant per case:      Density      .5
constant per case:     Modifier    1.0 1.01 1.025 1.04
1.055
TIME
number of steps:        5
time values:            .1 .2 .3 .4 .5
filename start number:  1
filename increment:     2

The following files would be needed for Example 2:
        example2.1.silo
        example2.3.silo
        example2.5.silo
        example2.7.silo
        example2.9.silo
```

README

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/silo/README.txt`

Simple Interface Data Load

Load your silo file (typically named with a suffix `.silo or .pdb or .case`) using the Simple Interface method.

Advanced Interface Data Load

Load your silo files (typically named with a suffix `.silo or .pdb or .case`) using the Advanced Interface method.

Data Tab	
Format	Use the **Silo** format.
Set file	This field should have the Silo Case file name or the .silo file.
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

STAR-CD and STAR-CCM+ Reader

Overview

Description

This reader reads .ccm, .ccmg (static geometry), .ccmp (variable), .ccmt (transient variable) data exported from STAR-CD version 4.x or STAR-CCM+.

Export Case Gold

STAR-CD version 3.x, STAR-CD version 4.x and STAR-CCM+ all export the native format of EnSight (EnSight Case Gold). Prostar exports to EnSight Case Gold Format. Use the 'automatic' export found with the NavCenter to export all parts and all primary variables. Use the Prostar command line to export separate parts, and/or any variable or combination of calculated variables. Both Steady State and Transient Models can be exported in a similar manner, with options of "automatic", or user controlled. The Case Gold format is likely to be more efficient, robust and faster in EnSight.

Read .sim file

This reader does *not* support .sim files written by STAR-CCM+. The .sim file should be translated using STAR-CCM+ into EnSight Case Gold.

Read Version 3 file

This reader does not support output from STAR-CD version 3.x. This data should be translated using Prostar into EnSight Case Gold.

Transient geometry

This reader does not support transient geometry. Transient geometry data must be exported to EnSight Case Gold.

Particle data (.trk)

Particle data is contained in the .trk file, which was formerly a File.33 file. The EnSight install includes a source file which can be compiled and run to translate the .trk (or File.33) file into EnSight's measured data format, which can be loaded together with the .ccm file (as described below), or the EnSight .case file can be edited to include the measured file name and it will automatically load.

The source file to the translator is found in the following location:

$CEI_HOME/ensight92/translators/starcd_file33

There is a README that guides you through compiling and using the translator. If you have difficulty with this, contact support@ensight.com and we will supply you with a compiled version for your hardware/OS. If you are using the translator and have a case gold format file, the translator will automatically edit the case file so that input of the measured data is automatic when your case file is loaded into EnSight. If you are using this reader and a .ccm file, then choose the EnSight 5 option and you will get a .res file that you can use to load in the measured data field described below.

Data Reader

`Main Menu > File > Data (reader)...`

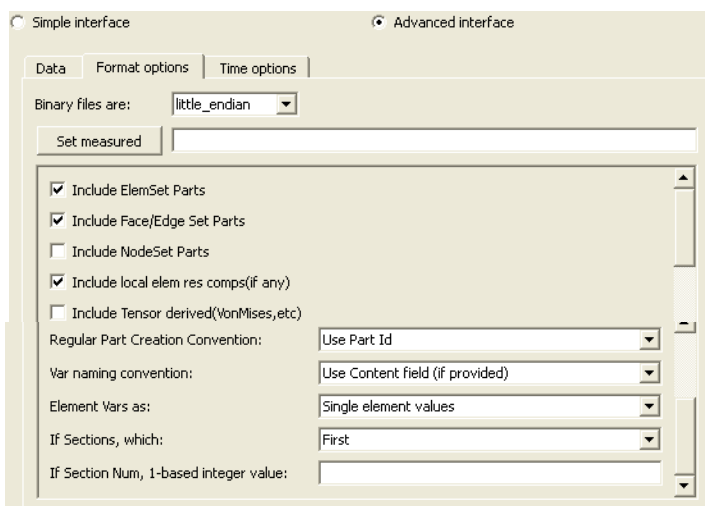
The File Selection dialog is used to specify which files you wish to read.
`Access: Main Menu > File > Data (Reader)...`

Simple Interface Data Load

Load your .ccm file using the Simple Interface method.

Advanced Interface Data Load

Load your STAR-CCM file using the Advanced Interface method.

Data Tab	
Format	Use the **STAR-CD CCM** format.
Set file	This field contains the first file name. For the first file you should choose the file with extension `.ccm`, `.ccmg`, or `.ccmp`. Clicking button inserts file name shown into the field. The `.ccmg` file contains geometry only and the geometry must be static. Loading a `.ccm` or `.ccmp` file will load both geometry and results. .
Set .ccmt	For a time varying variable data, set the `.ccmt` file using this field. This is only transient variable data.
Format Options Tab	
Set measured	Select the measured file and click this button. This can be the measured file obtained from the `File.33` or the `.trk` file using the EnSight translator (see above).
Other Options	
Include ElemSet Parts	Include any Element sets defined. These are sets of full elements which are generally some logical subset of the total number of elements. Default is on.
Include Face/ Edge Parts	Include any Face or Edge sets defined. These are some logical set of particular faces and/or edges of full elements. Default is on.
Include NodeSet Parts	Include any Node sets defined. These are generally the subset of nodes needed for the Element, Face, or Edge sets above. As such, they are generally not needed as separate parts, but can be created if desired. Default is off.

Include local elem res comps (if any)	Include the local stresses components, etc that are in the elemen Default is on
Include Tensor derived (VonMises, etc.)	For tensor results, calculate scalars from the tensors. Default is off
Regular Part Creation Convention	Parts will be created according to the following: Use Part Id - Part Id (this is the default) Use Property Id - Property Id Use Material Id - Material Id
Var naming convention	Use Content Field (if provided) - Variable names will be what is in the Content field, if provided. If not provided, they will be the VKI dataset name. This is the default. Use VKI dataset name - Variable names will be the VKI variable dataset name (which are reasonably descriptive).
Element Vars as	Single element values - Element results (whether centroidal or element nodal) will be presented as a single value per element. Thus will be per_elem variables in EnSight.This is the default. Averaged to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes without using geometry weighting. Thus will be per_node variables in EnSight. Geom weighted average to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes using geometry weighting. Thus will be per_node variables in EnSight
If Sections, which:	Not used
Section Num	Not used

(see How To Read Data)

STL Reader

Overview

Description

Reads .ccm files exported from STAR-CCM and STAR-CCM+.

Note: There is no longer an EnSight STL reader. this format is now read using the CAD reader.

Overview

Description

This reader will load STL files (either ASCII or binary). Note that STL files consist only of surfaces (triangles) and have no associated variables.

Usefulness

STL geometry format is widely compatible with a number of codes. Multiple STL files geometries can be created to represent scenery or background, then read in and scaled (using the -scaleg option) as a separate case to add to the presentation of your existing model results in EnSight.

Usage

The STL reader can read in an individual .stl file, or it can read in an exec file so that more than one stl file can be included in the same case.

Limitations

The current reader does not allow the coloring of each facet, nor does it allow coloring of each part, and just skips over color statements in the file.

STL binary file format

If the file is a binary STL (.stl) file, then it must contain exactly one part.

STL ASCII file format

If the file is an ASCII STL (.stl) file, then it can contain one or multiple parts. If you wish to read in multiple files

Single file multipart ASCII format is as follows:

```
solid part1
    ...
endsolid part1
solid part2
   ...
endsolid part2
```

Simple Exec file format

An exec file (.xct) is used to read in multiple STL files into one case. Because binary STL can contain only one part, if you wish to read in more than one binary STL file into a single case, then you must use an exec file. ASCII STL files with one or multiple parts can be read in to a single case using the exec file. An exec file can read in binary and ASCII files together into a single case. This exec file is a very simple ascii file that must conform to the following:

1. All lines must begin in column 1
2. No blank or comment lines allowed
3. If the stl filenames begin with a "/", it will be treated as absolute path. Otherwise, the path for the exec file will be prepended to the name given in the file. (Thus, relative paths should work).

```
line 0:    numfiles: N   (where N is the no. of files)
line 1-n:  stlfilename1
. . .
. . .
stlfilenameN
```

Example Simple Exec file

```
numfiles: 3
CASTLE.STL
bincastle.stl
test.slp
```

Rigid Body Motion Exec file

Release 2.1 of the STL reader includes the added ability to link each STL part with a rigid body transformation file to allow the STL part to rigidly translate and rotate over time. The rigid body motion Exec file has additional columns that contain the Euler Parameter filename (see Section 11.14, Euler Parameter File Format), the transformation title in the Euler Parameter file, and a units scale factor (This is used to scale the translations, not the geometry. Scaling of the geometry is accomplished in the Part Feature Detail Editor.). The rigid body version of this Exec file requires quotes as shown around the strings and values of the file lines.

example:

```
numfiles: 3
"CASTLE.STL"      "motion.dat" "CASTLE"  "1000.0"
"bincastle.stl"   "motion.dat" "BCASTLE" "1000.0"
"test.slp"        "motion.dat" "TEST"    "1000.0"
```

And if an additional offset is needed to the CG, add these in 3 more columns

example:

```
numfiles: 3
"CASTLE.STL"    "motion.dat" "CASTLE"  "1000.0" "1.35"  "2.66" "0.0"
"bincastle.stl" "motion.dat" "BCASTLE" "1000.0" "-2.45" "1.0"  "-2.0"
"test.slp"      "motion.dat" "TEST"    "1000.0" "60.2"  "23.4" "0.0"
```

You can also add a rotation order and yaw, pitch, and roll values on each of the file lines if the coordinate system needs to be re-oriented. These additional columns follow the same format as those in the EnSight Rigid Body (.erb) file.

(see Section 11.13, EnSight Rigid Body File Format)

README

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/stl/README`

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.stl or .xct`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry file (typically named with a suffix `.stl or .xct`) using the Advanced Interface method.

Data Tab	
Format	Use the **STL** format.
Set geometry	Select the geometry file (typically **.stl or .xct**) and click this button

Set results	As of version 8.0.7(h) this field is activated to allow flags to change reader behavior. In order to truncate the float values put in a tolerance value and the reader will retain only to the designated significant digit. This can be used to eliminate duplicate node problems due to roundoff error. Put in the keyword -tol 1e-3 to eliminate the fourth and smaller decimal point values in the nodal coordinates during data file read. See the README file for more details.
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

Tecplot Reader

Overview

Description

There are two Tecplot readers included with EnSight: **Tecplot Binary** and **Tecplot_ASCII** which read binary and ASCII Tecplot data.

TECPLOT Binary Reader Usage

The TECPLOT binary file format uses a Tecplot plt file.

Tecplot ASCII Reader

A subset of the Tecplot 360 ASCII format is read using the Tecplot_ASCII reader. In the format options tab of the data reader dialog, choose Debug to get extra output to the console if EnSight fails to read your ASCII file.

README

See the following directory for current information on these readers.

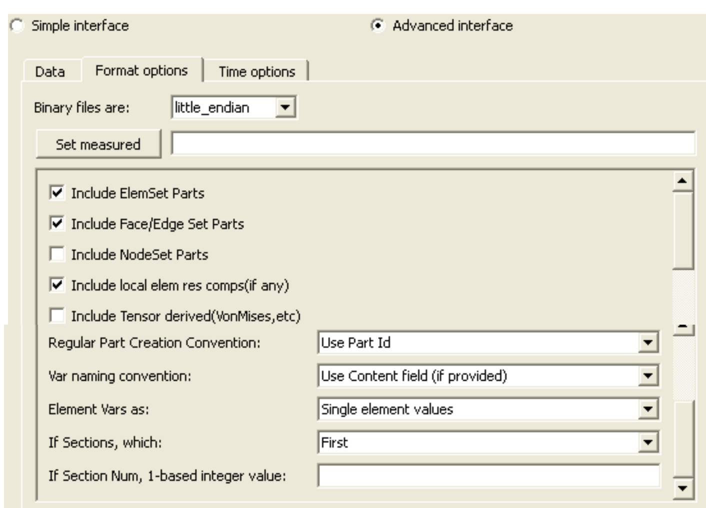
`$CEI_HOME/ensight92/src/readers/tecplot/`

Simple Interface Data Load

Load your Tecplot file (typically named with a suffix `.plt or .plot or .dat`) using the Simple Interface method.

Advanced Interface Data Load

Load your Tecplot file (typically named with a suffix `.plt or .plot or .dat`) using the Advanced Interface method.

Data Tab	
Format	Use the **or Tecplot_ASCII**, **Tecplot Binary**, or the legacy **TECPLOT 7.x** format.
Set plot (or dat)	This field should have the .dat file name for ASCII data and the .plt name for binary data.
Format Options Tab	
Set measured	Select the measured file and click this button.
Tecplot Binary Other Options	

Include ElemSet Parts	Include any Element sets defined. These are sets of full elements which are generally some logical subset of the total number of elements. Default is on.
Include Face/ Edge Parts	Include any Face or Edge sets defined. These are some logical set of particular faces and/or edges of full elements. Default is on.
Include NodeSet Parts	Include any Node sets defined. These are generally the subset of nodes needed for the Element, Face, or Edge sets above. As such, they are generally not needed as separate parts, but can be created if desired. Default is off.
Include local elem res comps (if any)	Include the local stresses components, etc that are in the element's local system. A simple example is a bar (such as a truss element), which only has tension or compression in the element's axial orientation. Such an element would have an axial stress variable. Other elements would have appropriate result component variables. Default is on

Include Tensor derived (VonMises, etc.)	For tensor results, calculate scalars from the following derived results (principal stress/strains, and common failure theories): Mean VonMises Octahedral Intensity Max Shear Equal Direct Min Principal Mid Principal Max Principal By default, all 9 of these will be derived. You can control which are created by this toggle, with an environment variable. Namely, `setenv ENSIGHT_VKI_DERIVED_FROM_TENSOR_FLAG n` where n = 1for Mean only 2 for VonMises only 4 for Octahedral only 8 for Intensity only 16 for Max Shear only 32 for Equal Direct only 64 for Min Principal only 128 for Mid Principal only 256 for Max Principal only 512 for all or any legal combination. example: for VonMises and Max Shear only, use 18. Default is off
Regular Part Creation Convention	Parts will be created according to the following: Use Part Id - Part Id (this is the default) Use Property Id - Property Id Use Material Id - Material Id
Var naming convention	Use Content Field (if provided) - Variable names will be what is in the Content field, if provided. If not provided, they will be the VKI dataset name. This is the default. Use VKI dataset nameVariable names will be the VKI variable dataset name (which are reasonably descriptive).

Element Vars as	Single element values - Element results (whether centroidal or element nodal) will be presented as a single value per element. Thus will be per_elem variables in EnSight.This is the default. Averaged to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes without using geometry weighting. Thus will be per_node variables in EnSight. Geom weighted average to node values - Element results (whether centroidal or element nodal) will be averaged to the nodes using geometry weighting. Thus will be per_node variables in EnSight
If Sections, which:	Which section will be used to create the variable First - The first section will be used (this is the default) Last - The last section will be used Section Num (below) - The section number entered in the field below will be used Separate Vars per Section - A separate variable will be created for each section.
Section Num	If the previous option is chosen to be Section Num, then the value in this field is the 1-based section number to use to create the variable.

(see How To Read Data)

Vectis Reader

Overview

Reader Visibility Flag

By default, this reader is not loaded into the list of available readers. To enable this reader go into the Menu, Edit > Preferences and click on Data and toggle on the reader visibility flag.

Reader vs. Translator

This reader is designed for files written before Vectis 3.6. For versions 3.6 or later, we recommend using the Ricardo v2e translator to convert the Vectis POST file to the Ensight format (for more details, see our FAQ on our website `www.ensight.com/FAQ/faq.0024.html`).

Pre-version 3.6 Description

This reader inputs either .TRI or .POS datasets as follows

Single TRI file - Gives the CAD geometry, but no variables (If you must see this along with your POST data, will have to read it as a second case), for example, CYLINDER.TRI

Single POST file WITH NO *'s in the name - Gives the geometry and variables in the post file, including surface patches and particles.

Multiple POST files - Enter a filename WITH *'s in the name

Gives the geometry and variables in the post files, which match the asterisks in a sequentially increasing pattern (starts at 1, increases by 1). Note: If your naming/numbering scheme is different than this, we require you to rename/renumber.

```
ex1) CYLINDER.POS.**      matches:
CYLINDER.POS.01     CYLINDER.POS.02     CYLINDER.POS.03
```

```
ex2) myfile***.pos        matches:
myfile001.pos    myfile002.pos
```

Query over time

Query node over time operation within EnSight will only work for cell variables on the cell part. Patch and drop variables will currently return all zeros.

Cell Variables

You may request cell variables on patch or droplet parts. The cell variable will be mapped onto them. BUT, be aware that any portions of the patches which are actually in the "external" cells will have zero values, because VECTIS doesn't contain that info directly. This leads to slightly "streaked" or "blotched" models which basically show the variable, but are probably not presentation quality. In order to eliminate this effect, neighboring cell information will need to be accessed - and at this time that work has not been done. Consider using Ensight's Offset Variable capability - it might be useful for certain models.

README

See the following file for current information on this reader.

`$CEI_HOME/ensight92/src/readers/vectis/README.txt`

Simple Interface Data Load

Load your Vectis file (typically named with a suffix `.TRI or .POS`) using the Simple Interface method.

Advanced Interface Data Load

Load your Vectis file (typically named with a suffix `.TRI or .POS`) using the Advanced Interface method.

Data Tab	
Format	Use the **Vectis** format.

Set tri/pos	Select the vectis file (typically **.TRI or .POS**) and click this button. This field should written by a Vectis version earlier than 3.6
Format Options Tab	
Set measured	Select the measured file and click this button.

(see How To Read Data)

XDMF Reader

Overview

Description

Reads eXtensible Data Model and Format files (.xdmf files).

This reader is based on the xdmf library from:

`pserver:anonymous@public.kitware.com:/cvsroot/Xdmf`

The reader can handle all the element types in Xdmf except:

`XDMF_MIXED`

`XDMF_POLYGON`

Structured meshes are converted to unstructured form automatically by the reader. The reader supports variables of type:

`XDMF_ATTRIBUTE_TYPE_SCALAR`

`XDMF_ATTRIBUTE_TYPE_VECTOR`

`XDMF_ATTRIBUTE_TYPE_TENSOR`

With centering:

`XDMF_ATTRIBUTE_CENTER_CELL`

`XDMF_ATTRIBUTE_CENTER_NODE`

The reader can handle 'Tree' grids. The reader does automatically decompose datasets for server of server mode (SOS) based on 'Tree' grids. The various grid blocks are distributed round-robin over the servers. Grids that are not 'Tree' grids will all be read on the first server.

The reader allows the user to pass a filename using a wildcard (* or ?) to select a collection of .xmf files. The reader assumes that each .xmf file contains a separate timestep. Some checks are made to verify that each file has the same structure/ variables, but the checks are not complete. Likewise, some basic checks are made for grids defined by reference. If all but one file has its grids geometry/topology by reference, the reader will assume that they can be reused for other timesteps.

The reader can also be passed a file in the schema:

```
<?xml version="1.0" ?>
<Xdmf>
<Dataset>
<File Name=".xmf file" [Time="unknown"]/>
<Template Name="sprintf for .xmf files" [Start="1"] [Step="1"] [Count="1"]
[TimeStart="unknown"] [TimeStep="1.0"]/>
</Dataset>
</Xdmf>
```

This defines a collection of files by specific names. These can be via individual names or via an sprintf() filename template and associated start, step and count

values. The file may also contain timestep values for each file (in the case of a sprintf() template, the values are generated). Note: if a file has an <Information Time=""/> element, that element will superceed the value specified here.

Simple Interface Data Load

Load your geometry file (typically named with a suffix `.XDMF`) using the Simple Interface method.

Advanced Interface Data Load

Load your geometry and result files (typically named with a suffix `.XDM and .XDMF`) using the Advanced Interface method.

Data Tab	
Format	Use the **XDMF2** format.
Set xmf	Select the geometry file (typically **.XDMF**) and click this button
Format Options Tab	
Verbose mode	Provide more output to the console to track progress and perhaps understand reader problems.
Enable Data Freeing	
Set measured	Select the measured file and click this button.

(see How To Read Data)

2.4 Other External Data Sources

External Translators

Translators supplied with the EnSight application enable you to use data files from many popular engineering packages. These translators are found in the translators directory on the EnSight distribution CD. (Installed translators reside in the $CEI_HOME/ensight92/translators directory.) A README file is supplied for each translator to help you understand the operation of each Particular translator. These translators are not supported by CEI, but are supplied at no-cost and as source files, where possible, to allow user modification and porting.

Exported from Analysis Codes

Several Analysis codes can export data in EnSight file formats. Examples of these include Fluent, STAR-CD, CFX and others.

2.5 Command Files

Command files contain EnSight command language as ASCII text that can be examined and even edited. They can be saved starting at any point and ending at any point during an EnSight session. They can be replayed at any point in an EnSight session. However, *some command sequences require a certain state to exist*, such as connection to the Server, the data read, or a Part created with a Particular Part number.

There are a multitude of applications for command files in EnSight. They include such things as being able to play back an entire EnSight session, easily returning to a standard orientation, connecting to a specific host, creating Particle traces, setting up a keyframe animation, etc. Anything that you will want to be able to repeatedly do is a candidate for a command file. Further, if it is a task that you frequently do, you can turn the command file into a macro (see To Use Macros below).

Saving command file

The command file which will repeat the entire current session can be saved from the menu as follows: Main Menu> File > Save > Commands from this session... This command file can then be replayed at startup of a new EnSight session and will redo step by step each of the commands.

Documenting Bugs

Command files are one of the best ways of documenting any bugs found in the EnSight system. Hopefully that is a rare occasion, but if it occurs, a command file provided to CEI will greatly facilitate the correction of the bug.

Nested Command Files

Command files can be nested, which means that if you have a command file that does a specific operation, you can play that command file from any other command file, as long as any prerequisite requirements are completed. This is done by adding the command `play: <filename>` in the command file.

Default Command File

EnSight is always saving a command file referred to as the *default command file* (unless the you have turned off this feature with a Client command line option). This command file can be saved (and renamed) when exiting EnSight, as described later in this section. The default command file is primarily intended to be a crash recovery aid. If something unforeseen were to prematurely end your EnSight session, you can recover to the last successfully completed command by restarting EnSight and running the default command file. Saving the Default Command File for EnSight Session

Command dialog

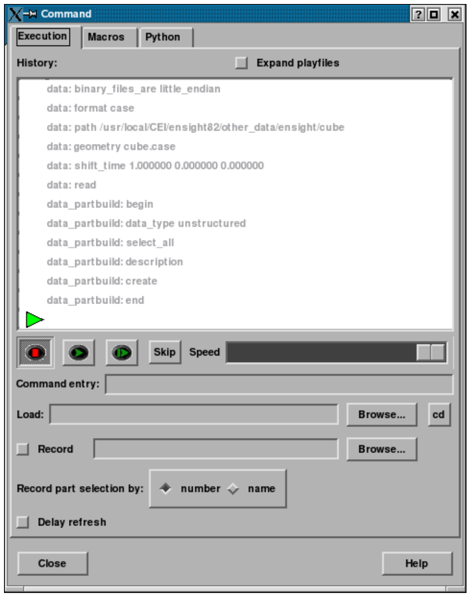


Figure 2-15
Command dialog - Execution tab

You use the Command dialog to control the execution of EnSight command language. The language can be entered by hand, or as is most often the case, played from a file. This dialog also controls the recording of command files.

`Access: Main Menu > File > Command...`

Execution Tab

History

In the History window, commands to be executed will be shown in black below the green current line indicator. As commands are executed, they will be show in gray above the current line. Many operations can be preformed on the commands in the History window by highlighting commands and clicking the right mouse button to bring up an action menu. From this menu you can:

Breakpoint	set a breakpoint which will stop command execution at the selected command.
Disable	disable the selected command(s).
Copy	copy the selected command(s) to the system clipboard.
Write/append	write the selected command(s) to a new file or append them to an existing file.
Execute	execute the selected command(s).
Goto	move the current line pointer to the selected command (Press play or step to resume execution at this command).

"VCR" buttons

Stop	Stops command file playing.
Play	Starts playing a command file. If you haven't provided a command file name (see Load, below), a File Selection dialog will open for you to Select or enter the file name. Command play continues as long as there are commands in the file, an interrupt: command has not been processed, and the Stop button has not been pressed.
Single Step	Executes only the current (next) command, indicated by the green current line pointer.

Skip — Skips over the current (next) command, indicated by the green current line pointer.

Speed — Use the Speed slider to control the speed of command file play.

Command Entry — Commands can be typed into this field for execution. Type the command and press return.

Load — The name of a command file to be executed can be typed into this field. Press return to load the file.

Or, you can use the associated Browse button to browse for a command file. A File Selection dialog will open. Select or enter the file name. The file will be loaded when you click “Open”.

The “Cd” button associated with the Load field can be used to change Ensight’s current directory to the directory of the loaded command file. This can be useful for playing command files that contain path information that assumes you’re starting from the command file’s directory.

Record — Check Record to start recording commands. If you have not typed a record filename in the text field provided, A File Selection dialog will open for you to Select or enter the file name. When Record is checked, all actions in EnSight are recorded to the specified file. As long as the record filename stays the same, the record button may be toggled on and off at any time, appending more commands to the file. When a new record file is selected, any existing commands in the file will be overwritten. You can browse for a new record file or directory at any time by clicking the browse button associated with this field.

Record Part Selection By — Use this radio button to select the method by which part selection will be recorded in the command language either by Number (default) or by Name.

Delay Refresh — When checked, this will cause the EnSight graphics window to refresh only after the playfile processing has completed or has been interrupted by the user.

Macros Tab

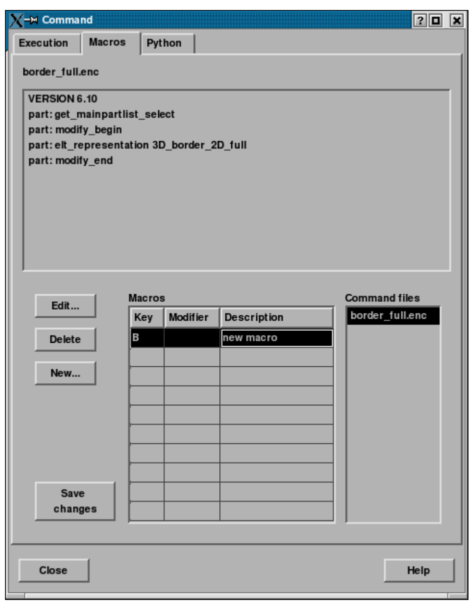


Figure 2-16
Command dialog - Macros tab

<filename> — This window displays the contents of the currently selected file in the Command files list (see below).

Macros — A three-column table that lists all of the currently defined keystroke macros.

Keystroke macros are defined in a text file, macro9.define. Macros can be defined at a site or local level, with local macros overriding site macros that might be defined for the same key. The macro9.define file (if any) that resides in the `%CEI_HOME%/ensight92/site_preferences/macros` directory defines site-level macros, while the macro9.define file (if any) under the user's EnSight Defaults directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/ Library/Application Support/EnSight92 on the Mac) will define that user's local macros. Any command files referenced by macros must be located in these directories as well.

In the Macros table, local macros are shown in black, site macros are shown in blue, and local overrides of site macros are shown in red.

The table columns are:

Key	a symbol representing the keyboard key on which a macro is based
Modifier	a symbol representing one of the modifier keys that may be pressed along with the base key, (SHIFT, CTRL, or ALT)
Description	a brief description of the macro

The "Edit", "Delete" and "New" buttons, below, operate on the macro selected in this table.

Edit Opens The Edit Macro dialog. Change any of the values in this dialog to edit the currently selected macro, then click "Close". Your changes will not be written to the macro8.define file until you either click "Save Changes" (see Below) or close the command dialog and answer "Yes" to the Save Changes query message.

Delete Deletes the selected macro, provided it is not a site macro.

New Opens The New Macro dialog. Change any of the values in this dialog to define the new macro, then click "Close". Your changes will not be written to the macro8.define file until you either click "Save Changes" (see Below) or close the command dialog and answer "Yes" to the Save Changes query message.

Save Changes Saves the changes from this Command Dialog session to the local `macro9.define` file.

Command Files This list shows the command files that are associated with the currently selected macro.

New/Edit Macros Dialog

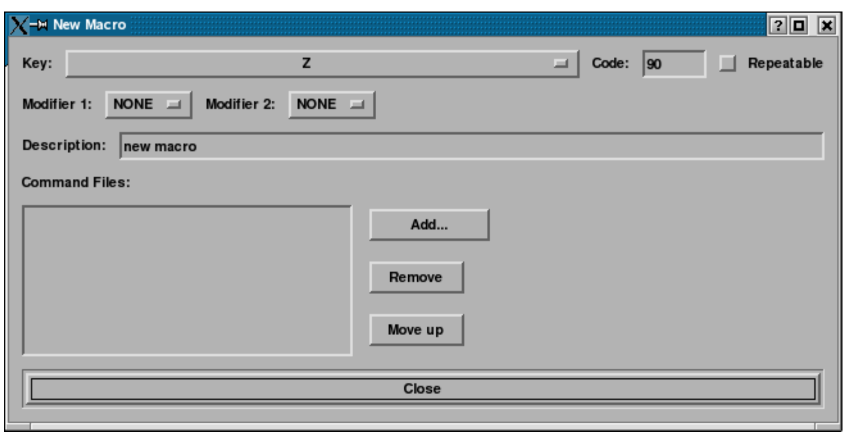


Figure 2-17

New/Edit Macro dialog

Key A list of the key symbols supported for defining macros

Repeatable If checked, this causes the macro to be repeated while the specified key is held down.

Modifier 1 An optional modifier key (SHIFT, CTRL, or ALT) to be held down along with the base key.

Modifier 2	A second optional modifier key.
Description	A brief text description that allows you to quickly identify the macro.
Command Files	Lists the command file(s) associated with a macro. Depending on how it is set up, a macro can execute command files three different ways: 1. A single file is executed once for each key press. 2. The command file repeatedly executes as long as the key is held down. 3. Multiple command files execute in a cycle for each keystroke.
Add...	To add a new command file to the list click "Add...". A File Selection dialog will open. Select or enter the desired file and click "Save".
Remove	To remove a command file from the list, select a file in the list, then click "Remove".
Move up	To change the order of execution of the command files listed, select a file, then click "Move up" to change its position.

Python Tab

The Interface Manual (see Chapter 6, EnSight Python Interpreter) contains a description of this section.

Troubleshooting Command Files

This section describes some common errors when running commands. If an error is encountered while playing back a command file you can possibly retype the command or continue without the command.

Problem	Probable Causes	Solutions
Error in command category	Incorrect spelling in the command category	Check and fix spelling
Command does not exist	Incorrect spelling in the command	Check and fix spelling
Error in parameter	Incorrect integer, float, range, or string value parameter	Fix spelling or enter a legal value
Commands do not seem to play	Command file was interrupted by an error or an interrupt command	Click continue in the Command dialog

(see How To Record and Play Command Files)

Saving the Default Command File for EnSight Session

EnSight is always saving a command file referred to as the *default command file* (unless the you have turned off this feature with a Client command line option). This default command file receives a default name starting with "ensigAAA" and is written to your /usr/tmp directory (unless you set your TMPDIR environment variable). This command file can be saved (and renamed) when exiting EnSight. If you do not save this temporary file in the manner explained below, it will be deleted automatically for you when you Quit normally from EnSight.

Quit Confirmation dialog

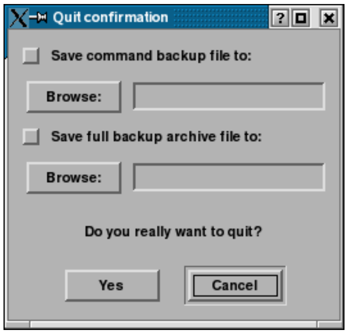


Figure 2-18
Quit Confirmation dialog

You use the Quit Confirmation dialog to save either or both the default command file and an archive file before exiting the program.

`Access: Main Menu > File > Quit...`

Save Command Backup File To: Toggle-on to save the default command file. Can also specify a new name for the command file by typing it in or browsing to it. (see Section 2.5, Command Files for more information on using command files.)

Save Full Backup Archive File To Toggle-on and specify a name (by typing it in or browsing to it) to create a Full Backup archive file.

Yes Click to save the indicated files and terminate the program.

(see How To Record and Play Command Files)

Auto recovery

While EnSight is running, an auto recovery command file, recover.enc, is written to the EnSight Defaults directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac). If EnSight crashes for some reason, this temporary file is not deleted. When EnSight is restarted (without using a play file) the user will be prompted with the option of using the auto recover command file.

Auto recovery dialog

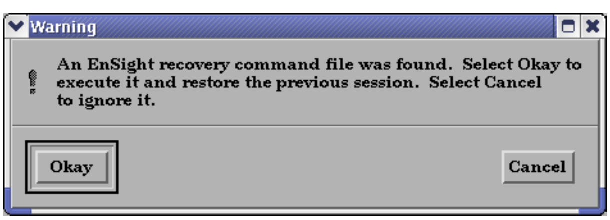


Figure 2-19
Auto recovery dialog

2.6 Archive Files

Saving and Restoring a Full backup

The current state of the EnSight Client and Server host systems may be saved to files. An EnSight session may then be restored to this saved state after restarting at a later time. A Full Backup consists of the following files. First, a small archive information file is created containing the location and name of the Client & Server files that will be described next. Second, a file is created on the Client host system containing the entire state of the Client. Third, a file is created on each Server containing the entire state of that Server. You have control over the name and location of the first file, but only the directories for the other files.

Restoring EnSight to a previously saved state will leave the system in exactly the state EnSight was in at the time of the backup. For a restore to be successful, it is important that EnSight be in a “clean” state. This means that no data can be read in before performing a restore. During a restore, any auto connections to the Server(s) will be made for you. If manual connections were originally used, you will need to once again make them during the restore. (If more than one case was present when the archive was saved, then connection to all the Servers is necessary).

An alternative to a Full Backup is to record a command file up to the state the user wishes to restore at a later date, and then simply replaying the command file. However, this requires execution of the entire command file to get to the restart point. A Full Backup returns you right to the restart point without having to recompute any previous actions.

A Full Backup restores very quickly. If you have very large datasets that take a significant time to read, consider reading them and then immediately writing a Full Backup file. Then, use the Full Backup file for subsequent session instead of reading the data.

***Important Note**: Archives are intended to facilitate rapid reload of data and context and are NOT intended for long-term data storage. Therefore, archives are likely NOT compatible between earlier EnSight versions and the current version (see Release Notes for details). If EnSight fails to open an archive, it will state that it failed and will write out a .enc file and echo its location. As command files ARE often compatible between earlier and later versions, the .enc file can likely be used to retrace the steps of the dataset.*

Save Full Backup Archive dialog

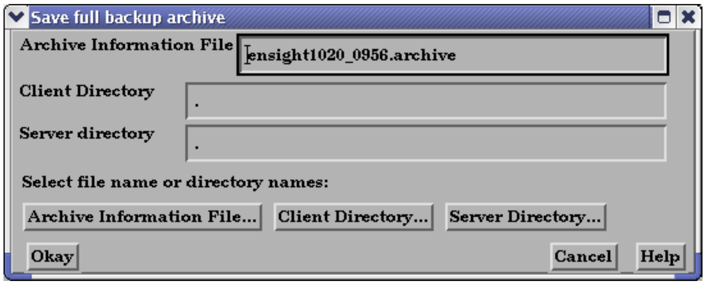


Figure 2-20
Save Full Backup Archive dialog

You use the Save Full Backup Archive dialog to control the files necessary to perform a full archive on EnSight.
`Access: Main Menu > File > Backup > Save Full Backup...`

Archive Information File — Specifies name of Full Backup control file.

Client Directory — Specifies the directory for the Client archive file.

Server Directory — Specifies the directory for the Server archive file.

Archive information File... — Click to display the file selection dialog for specifying the Archive Information File.

Client Directory... — Click to display the file selection dialog for specifying the Client Directory.

Server Directory... — Click to display the file selection dialog for specifying the Server Directory (for the selected case if there is more than one). Choose a common path if there is more than one.

Okay — Click to perform the full backup.

NOTE: This command is written to the command file, but is preceded with a # (the comment character). To make the archive command occur when you play the command file back, uncomment the #.

(see How To Save and Restore an Archive)

File Selection for Restarting from an Archive

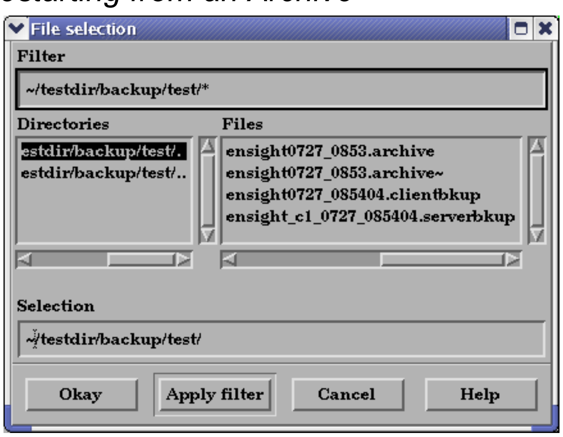


Figure 2-21
File Selection for Restarting from an Archive

You use the Restore Full Archive Backup dialog to read and restore a previously stored archive file.
`Access: Main Menu > File > Backup > Restore Full...`

Troubleshooting Full Backup

Problem	Probable Causes	Solutions
Error message indicating that all dialogs must be dismissed	When saving and restoring archives, all EnSight dialogs, except for the Client GUI, must be dismissed to free up any temporary tables that are in use. Temporary tables are not written to the archive files.	Dismiss all the dialogs except the main Client GUI.
Backup fails for any reason	Ran out of disk space on the Client or Server host system	Check the file system(s) you where you are writing (both the Server and the Client host systems) then remove any unnecessary files to free up disk space.
	Directory specified is not writable	Change permissions of destination directory or specify alternate location.

2.7 Context Files

EnSight context files can be used to duplicate the current EnSight state with the same or a different, but similar, dataset. The context file works best if the dataset it is being applied to contains the same variable names and parts, but can also be used when this is not the case.

Input and output of context files is described below as well as in How To Save or Restore a Context File and under Save and Restore of Section 6.1, File Menu Functions

Saving a Context File

To save the current context, simply entered the desired file name in the dialog under: Access: File > Save > Context...

(and if you have multiple cases to save, select Save All Cases)

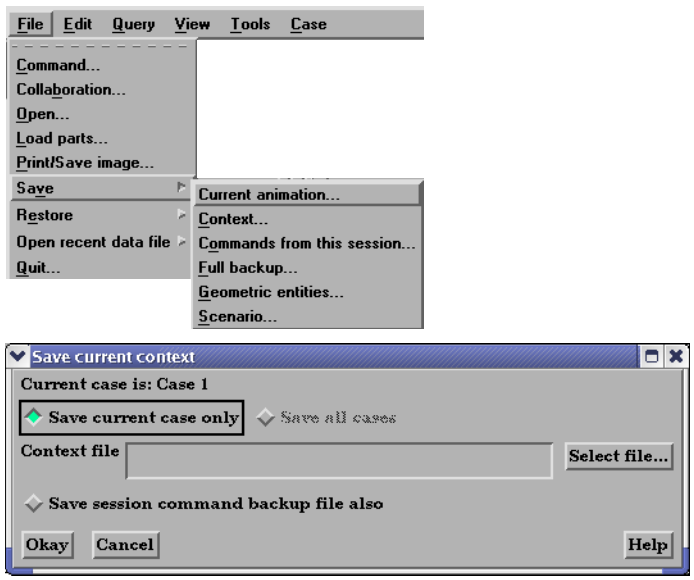


Figure 2-22
Saving a Context File

Restoring a Context

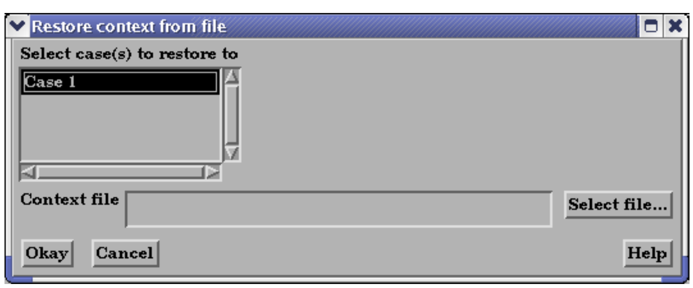


Figure 2-23
Restoring a Context File

If you are restoring a context file containing information for a single case, you can select the case or cases that you wish to apply the context to. If you are restoring a context file containing information about multiple cases, the selection list will be ignored.

When restoring a context you can 1) read the new dataset and build the new parts and then restore the context file, or 2) read the new dataset, close the part builder without building any parts and restore the context file (whereupon the context file will build the same parts as existed when it was saved) or 3) restore the context before reading any data (whereupon the previous state with the same dataset will be restored). The way you decide to do this depends upon whether the same parts exist in the new dataset.

If the same parts do not exist, you would typically read the new dataset and build the desired parts in the normal way. Then:

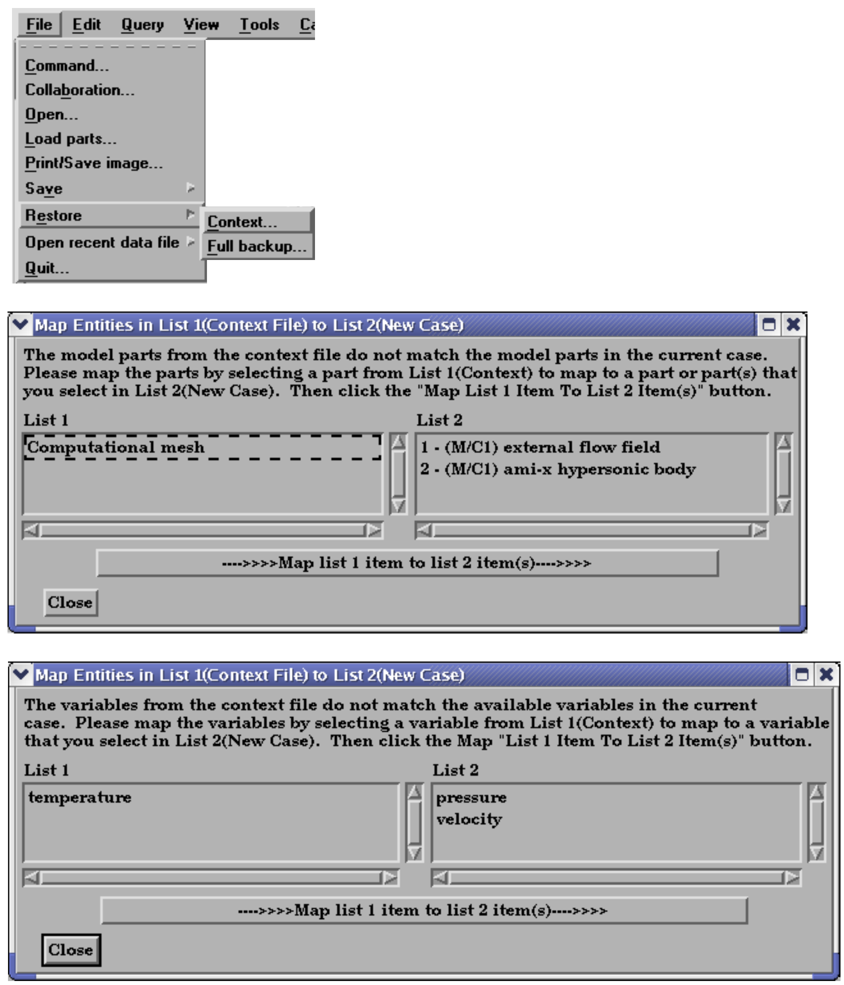


Figure 2-24
Restoring a Context

Flipbook animations are not restored using the context file because it is unknown at the time the context file is created what state existed when the flipbook was saved.

Context files use EnSight's command language and other state files (such as palette, view, and keyframe animation).

2.8 Session Files

An EnSight session file records the state of the visualization utilizing the context file capability along with a thumbnail of your graphics windows and a description of the session. When you restart EnSight, your recent sessions are displayed in the 'Welcome to EnSight' screen, complete with the thumbnail and description.

Saving a Session File

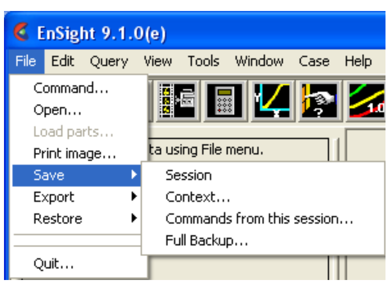
To save the current session, simply enter the desired file name in the dialog under:
Access: File > Save > Session...

You will be prompted for a description for the session. The description is shown on the Welcome screen, so this is a good place to make a note to remind yourself of the particulars of this visualization.

You can check the toggle to pack the data into your session file. Packing allows for true session file portability as it packs the original data directories as well as the context information compressed into one session file. The directories that will be compressed and saved are listed. The resulting single session file contains everything EnSight needs to reproduce the exact visualization on any EnSight installation that has the ability to read your data file. Your session file is a portable way to share visualizations with other EnSight users. Keep in mind that for a large data set, your session file can be quite large if you use the data packing option.

Click 'Save' to open a file browser and choose a location for your session file.

If you have multiple cases, a single session file will be saved that includes all the cases.



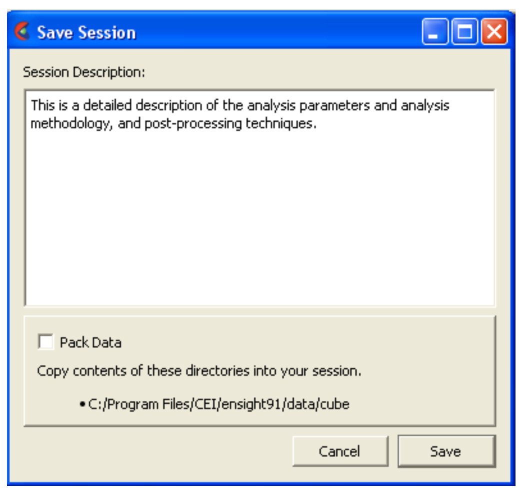


Figure 2-25
Saving a Session File

Restoring a Session

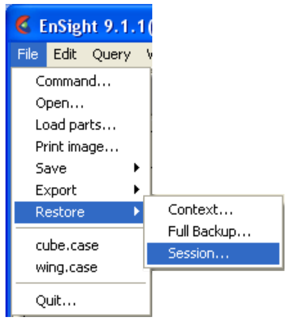


Figure 2-26
Restoring a Context File

Restoring a session is simple. First, your most recent session files are available in the 'Welcome to EnSight' screen on startup. Second, Mac and Windows users can double click any EnSight session file (ending in .ens) and the file will restore. Finally, you can browse for a session file from the EnSight menus as follows:

Access: Main Menu > File > Restore > Session...

If you are restoring a session file containing information for multiple cases, all of the cases will be restored. If you already have data loaded and restore a session, EnSight will delete all the cases, start anew and then restore the session.

2.9 Scenario Files

Scenario files are used by CEI's EnLiten product which is capable of viewing all geometry (such as parts, annotation, plots, etc.) that EnSight can display, including flipbook, keyframe, and particle trace animations.

A "scenario" defines all visible entities you wish to view with EnLiten and includes any saved views and notes that you want to make available to the EnLiten user.

When you create a scenario, the following may be saved: (a) EnLiten file containing geometric display information, saved views, and attached nodes. (b) A palette file for each visible variable legend. (c) A JPEG image file (not used by EnLiten). (d) A scenario description file (not used by EnLiten). (e) A EnSight context file (not used by EnLiten).

When saving a scenario, either the scenario file itself can be saved, or the scenario project - which includes all of the files in the previous paragraph.

EnLiten is a geometry viewer only. As such it is not capable of creating or modifying any new/existing information such as variables or parts, or of changing timesteps.

Since EnLiten is only a geometry viewer, only keyframe transformation information is stored when saving a scenario file, i.e., no transient data keyframing is possible (consider loading a flipbook instead).

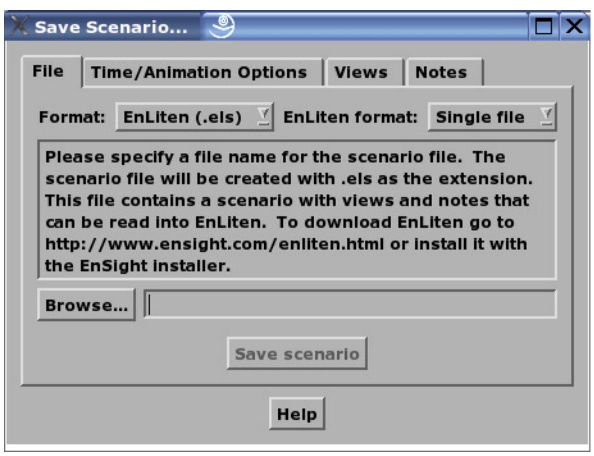


Figure 2-27
Save Scenario Dialog - File Tab

You use the Save Scenario dialog to control the options of the scenario files to be saved in EnSight for display in EnLiten.
`Access: Main Menu > File > Save > Scenario...`

File Tab

Format — EnLiten scenario files (.els) or Reveal CEI Scene Files (.csf) may be used.

EnLiten Format — *Project* - will save the scenario file plus files mentioned on the previous page.
Single file - will save only the scenario file.

File Name — Enter the file name to be saved or use the Browse... button.

Save scenario — Click to actually save the scenario.

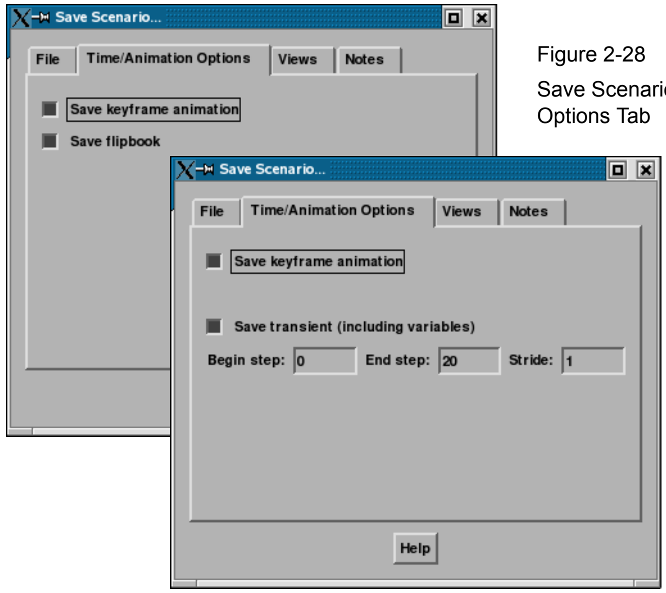


Figure 2-28
Save Scenario Dialog - Time/Animation Options Tab

Time/Animation Options Tab

Save keyframe animation — Only available if keyframe animation is defined. Toggle on to save the keyframe animation sequence to the scenario file.

Save Flipbook — Only available if flipbook is defined and saving to an EnLiten scenario file. Toggle on to save the flipbook information to the scenario file.

Save particle trace animation — Only available if animated particle traces exist. Toggle on to save the animated traces to the scenario file.

Save transient (including variables) — Only available if transient data exists and saving to a Reveal scenario file. Toggle on to save transient data to the .csf file.

Begin step - Beginning step to save.
End step - Ending step to save.
Stride - The step stride between Begin and End step.

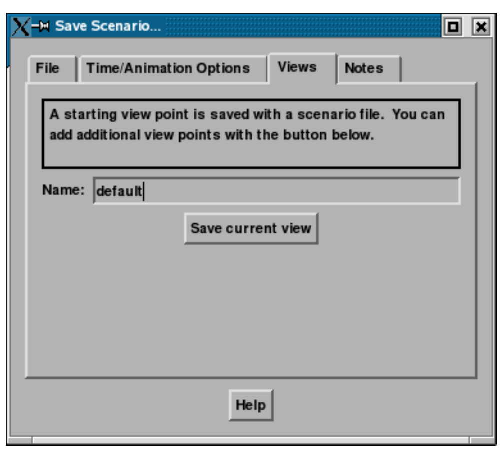


Figure 2-29
Save Scenario Dialog - Views Tab

Views Tab

After the scenario has been saved you may save additional views by setting the desired view in EnSight.

Name — The name of the view as it will appear in the scenario viewer.

Save current view — Click to actually save the view to the current scenario file.

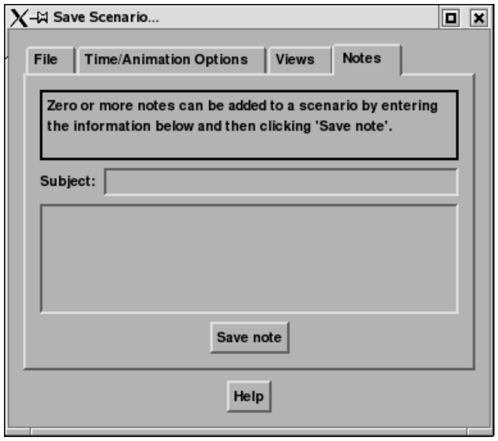


Figure 2-30
Save Scenario Dialog - Notes Tab

Notes Tab After the scenario has been saved you may write notes regarding the scenario.

Subject The subject of the note.

Note Field Any information you wish to enter for the note.

Save note Click to actually save the note to the current scenario file.

(see How To Save Scenario)

2.10 Saving Geometry and Results Within EnSight

Saving Geometric Entities

Sometimes you may wish to output geometric data or variable values from EnSight to be included in a different analysis code, or to be used in a presentation.

EnSight has three internal writers that allow saving geometric data and variable values in Brick of Values, Case (EnSight Gold) or . EnSight also allows the user to create their own writers. Each user-defined writer must be compiled into a dynamic shared library that is loaded at runtime and listed in the Save Geometric Entities dialog with the internal writer formats.

Both internal and user-defined writers have access only to the geometry of selected parts and each of their active variables. For all writers except VRML, only parts located on the server can be saved. This includes all original model parts, and the following created parts: 2D-clips, Elevated Surfaces, Developed Surfaces, and Isosurfaces.

One exception is the VRML internal writer which saves all the parts on the client in their current visible state except for Parts which have limit fringes set to transparent. The VRML 2.0 (.wrl file) will be saved on the client. The VRML output generally contains the same visual features as the Reveal product. The VRML export supports nodal variable coloring only. Parts colored using element variables will be displayed in their EnSight constant color. An element variable can be changed to a nodal variable (so that it can be written out in VRML format) using the EnSight calculator function ElemToNode. The mechanism for nodal color export is through texture mapping using an embedded texture map as the color palette.

The userd-defined writers can call the routines of an EnSight API to retrieve, to get, for example, nodal coordinates, node ids, element ids of parts selected in the Parts window to be passed by value to be used, manipulated and/or written out in any format desired. User-defined writer dialog includes a Parameter field that allows passing in a text field into the writer from the GUI for extra options.

Several example writers (including source code header files, Makefile and the corresponding shared library) are included to demonstrate this capability.

The Case (Gold) Lite reader is included to demonstrate how to exercise most of the API and output a subset of the Case (Gold) format. Complex numbers and custom Gold format are not supported with this writer. While the writer is not compiled, the source code of this writer, the required header files, and the Makefile are included.

The Flatfile user-defined writer is designed to demonstrate the output selected part nodal data (coordinates & IDs) as well as active variable values (scalar and/or vector only) data in a comma delimited format easily imported into other applications. If any of the keywords ‘ANSYS’ or ‘force’ or ‘body’ is entered into the Parameter field, then Flatfile will output an ANSYS body force file.

The STL user-defined writer is designed to write out the border geometry in the form of triangular 2D elements of the selected part(s) at the beginning timestep. The end time and the step time are ignored. The STL format does not support multiple parts in a single binary file, but does support multiple parts in a single ASCII file. Therefore, if multiple parts are selected and ascii is checked, the STL writer outputs an ascii file with the border of each of the parts. If multiple parts are selected and binary is checked, the STL writer outputs a binary file containing a single border of the multiple parts.

More user-defined writers may be distributed with EnSight in the future.

Save Geometric Entities dialog

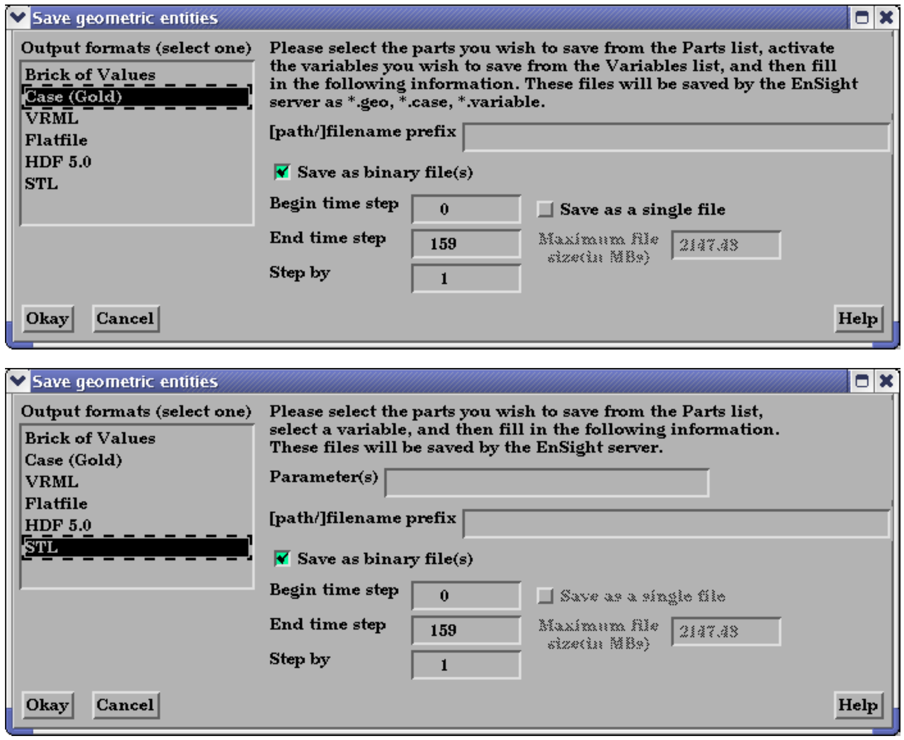


Figure 2-31

Save Geometric Entities dialog
(Showing Case (Gold) internal writer, and STL external writer)

The Save Geometric Entities dialog is used to save Selected Model, 2D-Clip, Isosurface, Elevated Surface, and Developed Surface Parts as EnSight Case (Gold) files. Thus modified model Parts and certain classes of created Parts can become model Parts of a new dataset.

`Access: Main Menu > File > Save > Save Geometric Entities...`

Output Format — Specify the desired format: Case(EnSight Gold), VRML, Flatfile, HDF5.0, STL, etc.

Parameter — Allows passing a text field from the GUI to the writer for extra options. Some writers make use of this field to modify their behavior (see Flatfile, for example) while others ignore this field. See the README file(s) in the following directory `$CEI_HOME/ensight92/src/writers`.

[path]/filename prefix — Specify path and filename prefix name for the saved files. For Case(Gold) the saved geometry file will be named filename.geo, the casefile will be filename.case, and the active variables will be filename.variablename. The VRML file will be filename.wrl. The other writers will vary.

Save As Binary File(s) — Save as Binary File(s) specifies whether to save the data in ASCII (button toggled off - default) or binary (button toggled on) format. Writers vary in their handling of this.

Begin Time Step — Begin Time Step field specifies the initial time step for which information will be available to save for all selected Parts and activated variables. Writers may vary in their usage of this information.

End Time Step	End Time Step field specifies the final time step for which information will be saved for all selected Parts and activated variables. Writers may vary in their handling of this.
Step By	Step By field specifies the time step increment for which information will be saved for all selected Parts and activated variables starting with Begin Time Step and finishing with End Time Step. The Step By value MUST be an integer. Writers may vary in their handling of this.
Save as a Single File	Toggle on to have a single file per variable - containing all values for all time steps for that variable. The default is to have a file per variable per time step. Writers may vary in their handling of this.
Maximum file Size	For Single File option, can specify the maximum file size. Continuation files are created if the file size would exceed this maximum. Writers may vary in their handling of this.
Okay	Click ok to pass the GUI values to the selected writer, and begin executing the writer routine

If Rigid Body Transformations in Model

Since EnSight does something special with the model timeset when rigid body information is read (via the rigid_body option in the casefile, or from a user-defined reader with rigid_body reading capability), you need to be aware of a few important issues. EnSight assumes that the rigid body timeset encompasses the normal geometry timeset, and it replaces the normal geometry timeset with the rigid body timeset - thus the following occurs when using this option.

1. If any created parts are in the list to be saved, EnSight will save as true changing coordinates. Namely, a geometry file containing the coordinates for each part will be saved at each time. Upon re-reading this model, you will be able to duplicate all actions, but it will be done as a true changing coordinate model. In other words, the original rigid_body file nature will not be duplicated.

2. If the original model had static geometry and rigid body file information - and you do not have any created parts in the list to be saved - saving will preserve the single static geometry and rigid_body file nature of the model. However, if the original model had changing geometry, or if variables have been activated - the number of geometry/variable files saved will be according to the rigid body timeset. This timeset often has many more steps than the original timesets - so be wise about the number of steps you write. It is often important to use the “Step by” option to control this.

3. Because of the things mentioned in 1 and 2 above - if you want to use the save geometric entities option in EnSight to “translate” a rigid body model from a different format into the EnSight format, you may want to consider the following process. First, read in the model without the rigid body transformations, activate the desired variables, and save the model. Second, read in the model with the rigid body transformations, do not activate any variables, and save the model (with a different name). Edit the Casefile of the first model to use the `model:` and `rigid_body:` lines of the second casefile instead of the first casefile.

Troubleshooting Saving Geometric Entities

Problem	Probable Causes	Solutions
A Part was not saved	User attempted to save an unsupported Part type.	Select only Model, Isosurface, 2D-Clip, and Elevated Surface Parts.
Variable(s) not saved	The variable was not activated or the variable was a constant.	Activate all scalar and vector variables you want saved.
Error saving	File prefix indicates a directory that is not writable or disk is out of space.	Re-specify a writable directory and valid prefix name. Remove unneeded files.
My custom user-defined writer doesn't show up on list of formats	Didn't load at startup	Start Ensight with -writerdbg option EnSight loads user-defined writers at startup from shared libraries found in $CEI_HOME/ensight92/machines/ $CEI_ARCH/lib_writers If your user-defined writer is not in the default directory, tell EnSight where to find it by: setenv ENSIGHT9_UDW location

(see How To Save Geometric Entities)

2.11 Saving and Restoring View States

EnSight's viewports provide a great deal of flexibility in how objects are displayed in the Graphics Window. Given the complicated transformations that can be performed, it is imperative that users be able to save and restore accumulated viewport transforms.

View saving and restoring is accessed from the Transformations dialog.

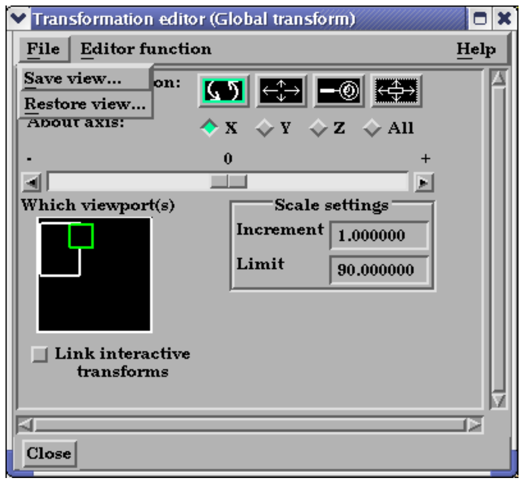


Figure 2-32
View Saving and Restoring in Transformation Dialog

`Access: Desktop > Transformation Edit... > File`

When either the Save View... or Restore View... selection is made, the user is presented with the typical File Selection dialog from which the save or restore can be accomplished. Save and Restore work on a single viewport.

(see also How To Save and Restore Viewing Parameters)

2.12 Saving and Printing Graphic Images

EnSight enables you to save an image of the Main View to a disk file or send it directly to a printer. The choice of save file formats depends on the implementation, but in all cases it is possible to obtain formats compatible with printers and plotters. Currently PostScript, AVI, BMP, EnVideo, GIF, JPEG, MPEG, PNG, PBM, Quicktime, SGI, TIFF, FLV, SWF, and XPM formats are available.

EnSight also enables you to save images of an animation to disk files. These files can then be converted and printed or recorded to video equipment (see Section 7.3, Keyframe Animation).

Print/Save Image dialog

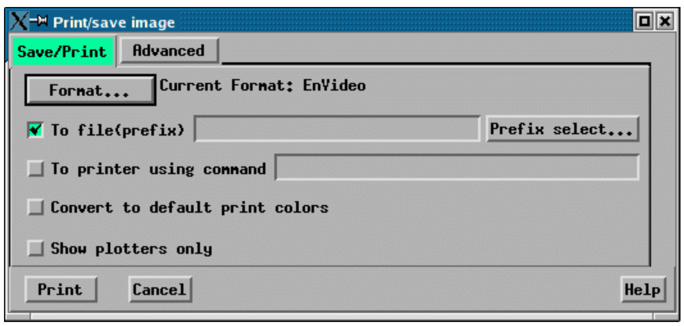


Figure 2-33
Print/Save Image dialog

You use the Print/Save Image dialog to specify the format and destination of an image to save. The destination can be a disk file or a printer. You also access the Image Format Options dialog for the various types from this dialog.

`Access: Main Menu > File > Print/Save Image...`

Format... Click to select image format.

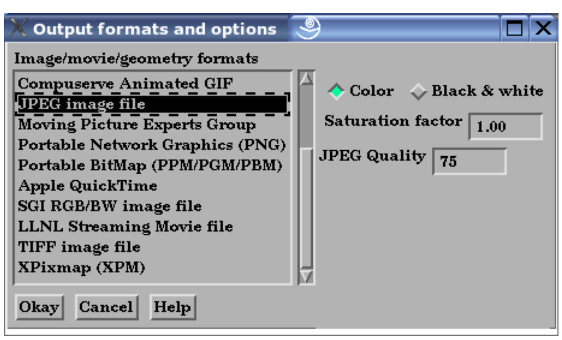


Figure 2-34
Output format and options dialogs

Color/Black & White Color versus Black and White toggle.

Saturation Factor At a value of 1.0, no change to the image. At lower values, a proportionate amount of white is added to each pixel. At a value of 0.0, the image would be all white.

Note: Each format can have other options specific to that format.

To File Toggle/Field — The image will be saved to this disk file name if toggle is on. This is a filename prefix. An appropriate suffix, according to the file format chosen, will be added.

To Printer Using Command Toggle/Field — The command to send a file to the printer if toggle is on. Make sure your printer is setup for the format you've selected.

Convert to default print colors — Clicking this toggle on will convert all black to white and all white to black but will leave all other colors as they are.

Show Plotters Only — Clicking this toggle will cause the graphics window to only display plotters.

Print — Click this button when all options are correctly specified and you are ready to print.

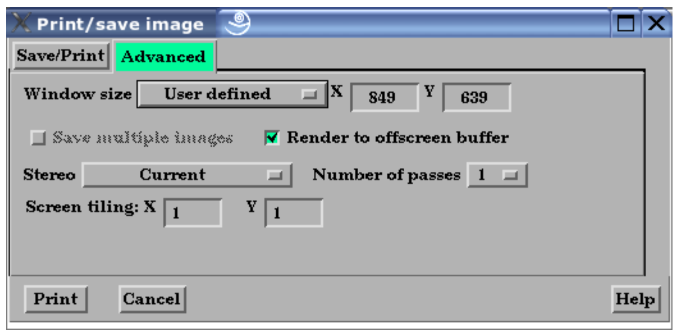


Figure 2-35
Print/Save Image Advanced dialog

Window Size — Specifies the size of the Graphics Window and the resulting image size.

- Normal — Creates a window which is the size of the current Graphics Window.
- Full — Creates a window which is the size of the full screen.
- User Defined — Creates a window which is specified in terms of its width and height in the X and Y fields.
- NTSC — Creates a window which is specified in terms NTSC standard window size (704 x 480).
- PAL — Creates a window which is specified in terms of PAL standard window size.
- Detached Display — Uses the detached display (as specified with - dconfig option) as the source for the output.
- DVD NTSC — Creates a window compatible with NTSC DVD format.
- DVD PAL — Creates a window compatible with PAL DVD format.
- HD 720p — Creates a window compatible with HiDef 1280x720 format.
- HD 1080p — Creates a window compatible with HiDef 1920x1080 format.

Save multiple images — If using a detached display (as specified with a -dconfig option), you may save an image from each display or a single image.

Render to offscreen buffer — Toggle on if the rendering shall occur in an off-screen rendering context. If off, the image will be "scraped" off the graphics window.

Stereo — Specifies the stereo vs. mono capabilities of the resulting image.

- Current — If the graphics window is mono, save a mono image. If stereo, save a stereo image.
- Mono — Save a mono image.
- Interleaved — Save a stereo interleaved image.
- Anaglyph — Save a stereo color separated image with left/right eye color as specified. (Cyan/Red, Red/Cyan, Blue/Red, or Red/Blue)

Number of passes	The number of rendering passes. The higher the number, the better the quality (but slower).
Screen tiling	The number of images in x and y that will be produced.

(see How To Print/Save an Image)

Troubleshooting Saving an Image

Problem	Probable Causes	Solutions
Image has blotches or ghosts of other windows in it	A viewport or menu was popped in front of the Main Graphics Window as the image was being saved.	Use offscreen rendering when possible.
Error while saving image file	Directory or file specified is not writable	Rename the file or change the permissions.
	Ran out of disk space	Check the file system you are using then remove any unnecessary files to free up disk space.
	Image format not selected	Select an image format before saving.
Image looks bad when printed	Original on-screen image has low resolution	Select “user defined” size or a pre-defined size such as “HD 1080p”
	Image has been dithered during processing	Do not enlarge or reduce the image until it is in your word processor.
	Non-integral ratio of printer resolution to image resolution at final size	The image is a pixel-map image. For best results, the number of printer-dots per image-dot should be an integer. For example, if the original image resolution is 72 dpi, reduced to 48% the final-size resolution is 72/.48 = 150 dpi. On a 600 dpi printer, each image pixel is exactly 4 printer-dots on a side.
Move/Draw PostScript output doesn’t look correct.	Primitives in Move/Draw PostScript output sometimes suffer from sorting problems.	Use Image Pixel type instead of Move/Draw.

(see How To Print/Save an Image)

2.13 Saving and Restoring Animation Frames

Both Flipbook and Keyframe Animation processes have save and restore capability. These are best described in the chapters devoted specifically to these features.

For Flipbook Animations, see Section 7.2, Flipbook Animation and How To Create a Flipbook Animation.

For Keyframe Animations, see Section 7.3, Keyframe Animation and How To Create a Keyframe Animation.

2.14 Saving Query Text Information

The data used for curves in EnSight's plotter and any other information from a query or otherwise which is presented in the EnSight Message Window can be saved to a file suitable for printing.

From Query/Plot Save... Formatted

One place this can occur is in the Query/Plot Quick Interaction area as described below as well as in Section 7.5, Query/Plot

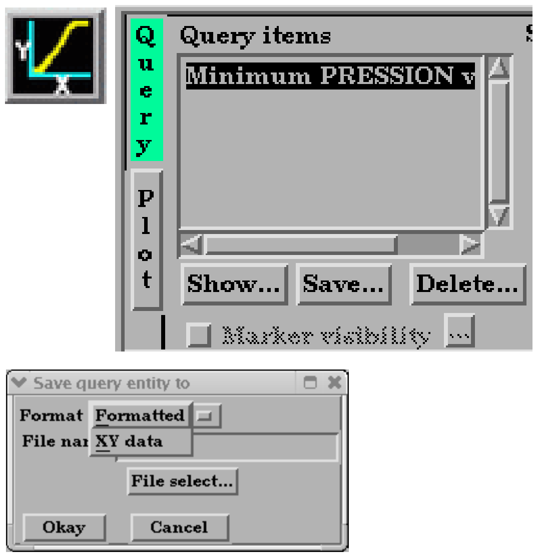


Figure 2-36
Saving or Loading XY Plot Data

`Access: Desktop > Query/Plot Feature`

Once the desired query item (curve) is selected in the list, the user can perform a Save operation by:

Save... — Click this button in the Quick Interaction area of the Query/Plot feature to save the plotter curve data.

Format — Select the Format of the data to save.

Formatted is a table suitable for printing.

XY Data is the xy file format described in Section 11.10, XY Plot Data Format, which is suitable for re-loading into EnSight.

File Name — Enter the desired filename for the xy data file, or click *File Select...* to be presented with the typical File Section dialog from which to perform the operation.

From Query/Plot Show Text

Show ... — Click this button to see the plotter curve information presented in the EnSight Message Window.

From EnSight Message Window

A file suitable for printing can be saved from any operation which places its information into the EnSight Message Window, such as Show Information queries and the Query/Plot Show Text... button described previously.

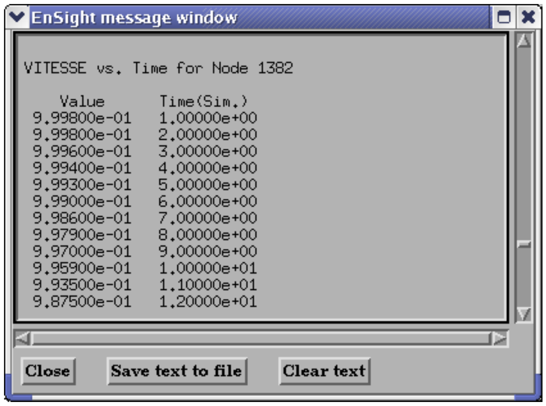


Figure 2-37
EnSight Message Window with Save Text To File Button

Save Text To File Brings up the typical File Selection dialog from which the information can be saved in the file of your choice.

2.15 Saving Your EnSight Environment

Every user has different postprocessing needs and personal preferences for how the EnSight windows should be positioned and sized. EnSight allows you to save dialog expandable section settings, and dialog size and position information to a file called `ensight9.winpos.default`. EnSight looks for this file at start up in the current Client directory and if not there in the EnSight defaults directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac) and will bring the user interface dialogs up according to your saved settings (if the file is found).

Almost all major dialog windows are saved in the:

ensight9.winpos.default_XRESxYRES

file (where XRES and YRES are the resolution of the monitor when the preferences were saved). The only exception are minor prompt dialogs. There are also some dialogs for which you cannot save the size (such as the Tool Positions dialog).

The ensight9.winpos.default file also contains the size and location for all of the windows containing graphics.

A number of other settings, such as mouse and keyboard buttons and Icon Bar settings can also be saved to a user preferences file.

(see Preferences... in Section 6.2, Edit Menu Functions and How to Save GUI Settings)

3 Parts

The *Part* is the fundamental visualization entity in EnSight. Virtually every postprocessing task you perform will involve a Part, thus it is vital to understand how Parts work.

A Part is a collection of nodes and elements that are grouped together and share the same attributes. When you start EnSight, you either read directly or interactively extract Parts from the data files. Parts which come from the original dataset are referred to as model Parts. Other Parts created within EnSight, are referred to as created (or dependent) Parts.

In this chapter you will learn how to produce created Parts (parts derived from other parts) and how to modify the attributes of all Part types.

Section 3.1, Part Overview is ***extremely important***. It defines how Parts work together to form other Parts and explains the dependencies which may exist between model Parts and created Parts. Failure to understand the concept of Parts as explained in this section will limit your ability to use EnSight. Please study this section carefully.

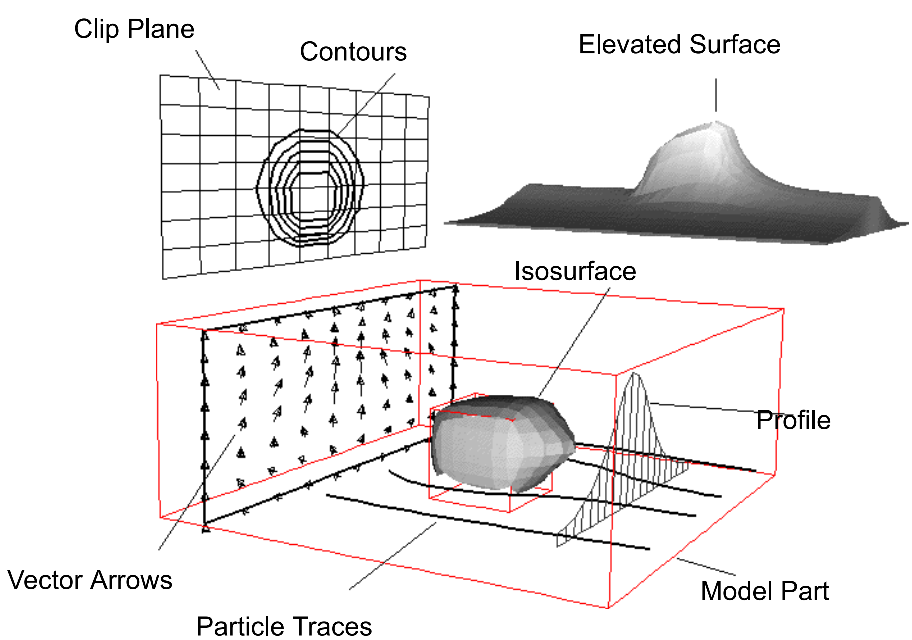


Figure 3-1
Various EnSight Part Types

Included in this Chapter are:

3.1 Part Overview

In EnSight, a Part is simply a collection of nodes and elements which are grouped together, will be manipulated together, and which share the same attributes. This section defines Parts and how they are related. It gives you an overview of the Part types and Part attributes that are available within EnSight.

Parts that are defined or extracted from your dataset are referred to as *model* Parts. Parts that are created within EnSight are referred to as *created* (or *dependent)* Parts. The types of Parts that you create depends on what features within EnSight you choose to utilize. Any created Part is derived from Parts that already exist, which is why the created Parts are sometimes called dependent Parts—they depend on the Parts from which they were created. The Parts that are used to create a dependent Part are referred to as *parent* Parts. Any time that a parent Part changes, its dependent Parts must also change. A parent Part will change when you change its attributes, or modify the current time in the case of transient data.

The Main Parts List contains all Parts that have been read in from your results data or created within EnSight. Displayed is the default parts window which lists the part number and the part name.

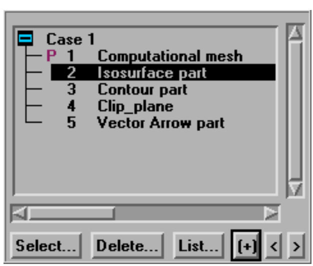


Figure 3-2
Default Main Parts List

“+” Button The “+” button can be used to lengthen the part window to display a larger number of parts.

“<“ “>” *Buttons* The “<“ and “>” buttons (in non-windows clients) can be used to expand or contract the main part list horizontally. This is useful for longer part names and when displaying part details. In Windows, the GUI has a movable vertical bar for this purpose.

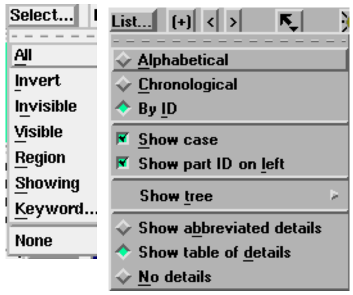


Figure 3-3
Select and List Pulldown Menus

Select Button The Select Button can be used to choose Parts using logical operators:

All	Select all parts
Invert	Unselect the selected and select the unselected
Invisible	Select the parts with visibility toggle off
Visible	Select the parts that have the visibility toggle on
Region	Select the Parts using the graphical selection tool (This does not work for vector arrow parts)
Showing	Select Parts that are toggled visible, that are within the graphics viewing region, and within the z-clip. Auxiliary clipping has no effect on this selection. (This does not work for vector arrow parts.)
Keyword	Select the parts using wildcards
None	Unselect all parts

Delete Button The Delete button can be used to delete selected parts.

List Button The List Button can be used to sort and select level of detail of information about the parts.

Sorting Click on the List button and choose:

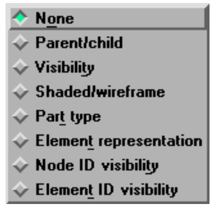
Alphabetical	To sort the list of parts alphabetically
Chronological	To sort the list of parts by their chronological creation time.
ID	To sort the list of parts by part ID (the default)

Level of Detail Click on the List button and choose:

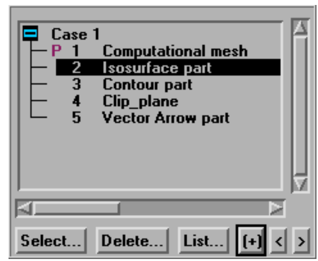
Show case Toggle off to remove the Case indicator or on (default) to see the Case designator. The Case of the data indicates the server on which the data resides. With Case toggled on, all parts in a given Case are indented underneath the Case number designator.

Show part ID on left Toggle on or off to control the display of part numbers.

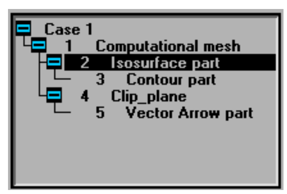
Show tree — Open to display the various hierarchical part display options.



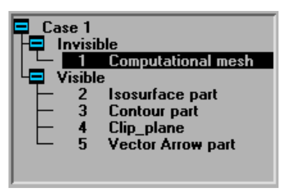
None — For the default list of parts - showing only the "P" next to the parent of the selected part. The "P" is important because modification of the Parent can affect the dependent child part.



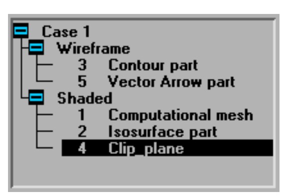
Parent/child — To show the hierarchical relationship between parent and child parts



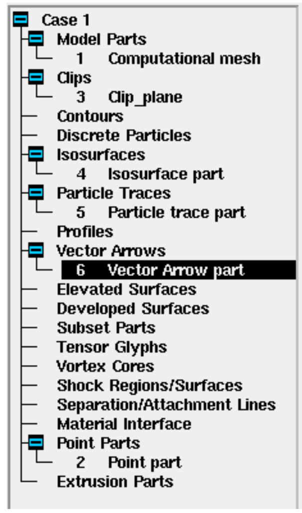
Visibility — To show parts based on whether they are visible or not.



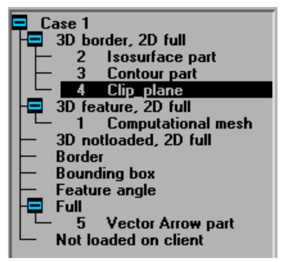
Shaded/wireframe — To show parts based on their local shaded attribute setting.



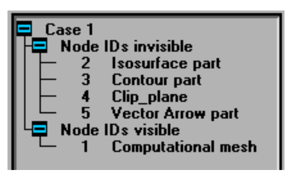
Part type To show parts based on their part type.



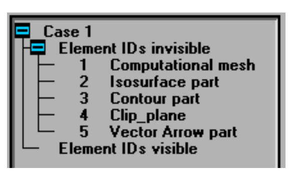
Element representation To show parts based on their element representation.



Node ID visibility To show parts based on their node id visibility.



Element ID visibility To show parts based on their element id visibility.



Show abbreviated details — Toggle on and the Part List will contain more information. Displayed are a case number, ID number, visibility attribute, shaded attribute, part symbol, representation, node and element id display status, and a Part description. The figure below of the Parts List shows a number of different Part types in this display mode

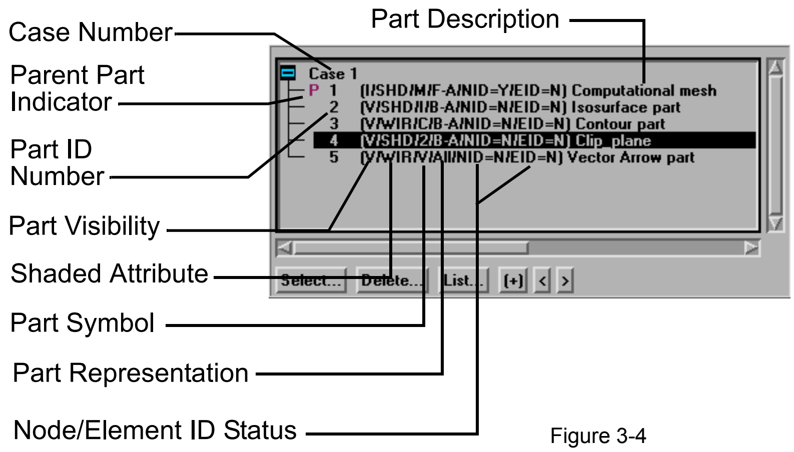


Figure 3-4
Abbreviated Parts List

Note that in the illustration above the Clip_plane part is selected and that there is a “P” in the left column next to the Computational mesh (model) Part. This indicates that the Computational mesh Part is the parent Part of the clip_plane Part. All parent Parts of a created Part will be so noted if that individual created Part is highlighted in the Main Parts List.

Every Part in the abbreviated view has a Visibility indicator, either “V” for visible or “I” for invisible.

Every Part has a shaded indicator, either “SHD” for shaded or “WIR” for wire frame.

Every Part has a representation. Shown below are the symbols used in the abbreviated view.

B-A = 3D Border 2D Full
F-A = 3D Feature 2D Full
N-A = 3D Nonvisual 2D Full
Brd = Border
Fea = Feature Angle
Box = Bounding Box
All = Full
Non = Nonvisual

Every Part has Node and Element ID display status equal to “Y” if displayed or “N” if not displayed.

Every Part has a symbol to describe it. Table 3-1 lists all of the different types of Parts and their associated symbols in the abbreviated view.

Table 3–1 Part Types, Symbols, and Descriptions

Part Type	Symbol	Description
Clip	(2)	A surface or line resulting from a clip of other Parts using the line, plane, or quadric tools
Contour	(C)	Lines of constant value on 2D elements
Developed Surface	(D)	A planar surface derived by unrolling a surface of revolution (i.e., the unrolling of a cylinder clip Part produced by the cylinder quadric tool)
Elevated Surface	(E)	Surface created by elevating elements by a variable
Extrusion Part	(U)	Translational or Rotational extrusion of 2D part into 3D elements.
Isosurface	(I)	Surface of constant value through 3D elements of other Parts
Material Part	(A)	A Part created according to the intersection of or domains of material values
Model Part	(M)	A Part that originated from the dataset
Particle Trace	(T)	Path of a massless Particle through a vector field
Point Part	(O)	Point part created using point part feature icon. Conceptually this is like a point clip.
Profile	(P)	Plot of a variable along a line (Similar to a 2D elevated surface)
Separation/ Attachment Line	(L)	Line where flow separation or attachment is occurring
Shock Surface/ Region	(K)	Surface or region where shock is occurring
Subset	(S)	Valid node and/or element label range(s) from model Part(s)
Tensor Glyph	(G)	Glyph showing direction of first, second, and third eigenvectors of a tensor field.
Vector Arrow	(V)	Arrows showing direction and magnitude of vector field
Vortex Core	(X)	Line representing center of a flow vortex

:

Show table of details — Toggle on to show a table of part details, as shown in the composite below. While this is more self explanatory than the abbreviated mode, it also requires horizontal scrolling to see the whole table of information.

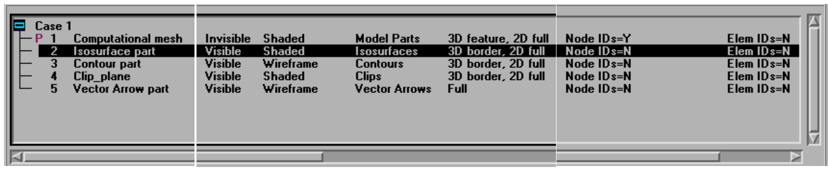


Figure 3-5
Composite of Detailed Parts List

No details — Toggle on to omit the details, showing only the part number and the part description.

Note: Several aspects of the Part display (including what is actually displayed in the table of details or abbreviated details) can be customized by using Edit > Preferences > Parts.

Reassign Parent

Parent Parts of any created Part can be changed by first selecting the created Part in the Feature Detail Editor, then selecting a new parent Part in the Main Parts List, and finally by clicking the Update Parent button in the Feature Detail Editor. You can open the Part Feature Detail Editor by right-clicking in the part list and choose Advanced.

Part Creation

Part creation occurs on either the server or the client. Since the data that is available on the client and server are different, it is useful to understand where Parts are created and where the data structures are stored. By understanding this, you will understand why some Parts can be created with certain parent Parts and others cannot. This information can be gained by examining the following table.

Table 3–2 Part Creation and Data Location

Part Type	Where Created	Data on Server	Data on Client
Clip	Server	Yes	Depending on Part attributes
Contour	Client	No	Yes
Developed Surface	Server	Yes	Depending on Part Attributes
Discrete Particle	Not Applicable	Yes	Depending on Part attributes
Elevated Surface	Server	Yes	Depending on Part attributes
Isosurface	Server	Yes	Depending on Part attributes
Model	Not Applicable	Yes	Depending on Part attributes
Material Part	Server	Yes	Depending on Part attributes
Particle Trace	Server	No	Yes
Point Part	Server	Yes	Depending on Part attributes
Profile	Client	No	Yes
Separation/ Attachment Line	Server	Yes	Depending on Part attributes
Shock Surface/ Region	Server	Yes	Depending on Part attributes
Subset	Server	Yes	Depending on Part attributes
Tensor Glyph	Client	No	Yes
Vector Arrow	Client. Server if necessary.	Maybe	Yes
Vortex Core	Server	Yes	Depending on Part attributes

(see Introduction to Part Creation)

Part Attributes

Both Model Parts (symbol: M) and Created Parts (multiple symbols) have *Creation Attributes*. For example structured Model Parts Creation Attributes include the I,J,K and step attributes used to create the part, and an Isosurface Part's Creation Attributes include the variable that was used to create it, and the value of the variable.

In addition, all Parts have a set of *Display Attributes* that are used in visualizing the Part in the Graphics Window. *Display Attributes* deal with such things as color, line width, symmetry operations, etc. *Display Attributes* do not control how the Part is created, only how it appears or how it behaves in the Graphics Window.

Both *Display* and *Creation Attributes* can be modified using the Feature Detail Editor which will show attributes grouped together under turndown sections. To access the Feature Detail Editor for a Part, simply right-click on that part in the part list, or on the geometry in the graphics window and choose ‘Edit’. The Feature Detail Editor will display all of a Part’s *Display* and *Creation Attributes* for editing.

Table 3–3 Display Attribute Sections

Section:	Includes controls for...
General Attributes (see Section 3.3, Part Editing)	Visibility in Graphics Window and individual Viewports
	Symmetry options
	Susceptibility to Auxiliary Clipping
	Reference frame
	Response to changes in time (frozen or active)
	Coloration (constant or by a palette associated with a variable)
	Shaded Surface and Hidden Line display
	Surface shading (flat, Gouraud, smooth)
	Opaqueness and Fill density
	Lighting (diffuse, shininess, and highlight intensity)
Node, Element, and Line Attributes (see Section 3.3, Part Editing)	General Visibility: Node, Line, and Element
	Label Visibility: Node and Element
	Node Representation: Node type (dot, cross, or sphere), Node Scale, Node Detail (for spheres), and Node size (constant or variable)
	Line Representation: Line Width and Line style (solid, dotted, or dot-dash)
	Element representation on client (full, border, 3D border/2D full, feature angle, or non visual), Element-size, Shrink-Factor, Element Angle and Polygon Reduction factor, Failed Elements
Displacement Attributes (see Section 3.3, Part Editing)	Displacement variable
	Displacement scaling factor

3.2 Part Selection and Identification

In the process of creating a Part you will need to be able to select the parent Part(s). This section describes Part selection and identification

Selecting Parts using the mouse in the Graphics Window

Geometries in the Graphics Window are selected using the left or right mouse buttons. Edit>Preferences, Mouse and Keyboard if you want to change this behavior. Selected parts are highlighted in the part list. They are highlighted also in the graphics window by default unless disabled by clicking on the global highlighting toggle. For more detail on this method, see How To Select Parts

To:	In the Graphics Window	Details
Select a Part	Left-click	Place the mouse pointer over the geometry in the graphics window and click the left mouse button. The part is highlighted in the graphics window and in the part list to reflect the "selected" state.
Select a Part with a to-do list	Right-click	Place the mouse pointer over the geometry in the graphics window and click the right mouse button. The part is highlighted in the graphics window and in the part list, to reflect the "selected" state. Also a pull-down menu appears to allow you to change some of the part characteristics or to create other parts using this part.
Keep current selection and toggle selection of part under cursor	Control-Left click	To add a part to the selected parts, place the mouse pointer over the geometry in the graphics window. Press the control key and click the left mouse button. This action will extend a selection by adding the new part and highlighting it in the graphics window. To de-select a part, place the mouse pointer over the geometry in the graphics window that is selected. Press the control key and click the left mouse button.
De-select all parts	Left-click	Left click on the background in order to deselect all parts.

To:	In the Graphics Window	Details
Select using ‘p’ key	Press ‘p’ key	This is considered a legacy mode of selecting parts. Place the mouse pointer over the geometry in the graphics window and press the ‘p’ key, which will select a previously unselected part, and the part will become highlighted in the graphics window and in the part list.
Quick Part To-Do List	Control Right-Click	This brings up a context-sensitive, part to-do list without changing part selection.

Selecting Parts using the mouse in the Part List Window

Items in all Parts Lists are selected using standard Motif/Win32 methods:

To:	Do This:	Details
Select an item	Select (or single-click)	Place the mouse pointer over the item and click the left mouse button. The item is highlighted to reflect the “selected” state.
Extend a contiguous selection	Select-drag	Place the mouse pointer over the first item. Click and hold the left mouse button as you drag over the remaining items to be selected. Only contiguous items may be selected in this fashion.
Extend a (possibly long) contiguous selection	Shift-click	Select the first item. Place the mouse pointer over the last item in the list to be selected. Press the shift key and click the left mouse button. This action will extend a selection to include all those items sequentially listed between the first selection and this one.
Extend a non-contiguous selection	Control-click	Place the mouse pointer over the item. Press the control key and click the left mouse button. This action will extend a selection by adding the new item, but not those in-between any previously selected items.
De-select an item	Control-click	Place the mouse pointer over the selected item. Press the control key and click the left mouse button. This action will de-select the item.
Open the Quick Interaction Area for a Part	Double-click	Place the mouse pointer over the item and click the left mouse button twice in rapid succession.

Selecting Parts using the Select Button

Underneath the Parts list is the Select Button which can be used to select Parts using logical operators.

All	Select all parts
Invert	Unselect the selected and select the unselected
Invisible	Select the parts with visibility toggle off
Visible	Select the parts that have the visibility toggle on
Region	Select the Parts using the graphical selection tool (This does not work for vector arrow parts)
Showing	Select Parts that are toggled visible, that are within the graphics viewing region, and within the z-clip. Auxiliary clipping has no effect on this selection. (This does not work for vector arrow parts.)
Keyword	Select the parts using wildcards
None	Unselect all parts

3.3 Part Editing

QIA vs. FDE

In EnSight, new Parts can be created and edited in their Quick Interaction Area (QIA) just below the top list of icons. The Quick Interaction Area contains only the most commonly used attributes, but not all of the part's attributes. To get access to a comprehensive list of attributes, you must open the Feature Detail Editor (FDE) for that part type and select that Part in the Feature Detail Editor Window and apply your modifications

Accessing FDE

There are several ways to access the advanced attributes in Feature Detail Editor for a particular part. Easiest is to right click on that part in the part list and select Edit. The FDE Window will pop up with the part selected and all it's attributes shown. Another way is to click on the Advanced button in the Quick Interaction Area for that Part type and then select the part in the FDE window that pops up. Finally, you can just right click on the part in the graphics window and select 'Edit' and the FDE Window for that part will pop up. The FDE Window will also open up if you double click a Feature Icon at the top of EnSight. This process is described in Sections 7.2 to 7.9.

Immediate Mode

Note that while each individual change made in the Quick interaction Area Editor is applied to the Part immediately, the Feature Detail Editor allows you to make a number of changes to various attributes and then apply them all at one time. This is done by toggling off View > Immediate Modification in the Feature Detail Editor. The default behavior is to immediately apply a change when you press Return.The Feature Detail Editor for Parts is opened from the Main Menu

Finally there is a legacy method of accessing the FDE via the EnSight menu: Choose your FDE using Edit>Part feature detail editors.

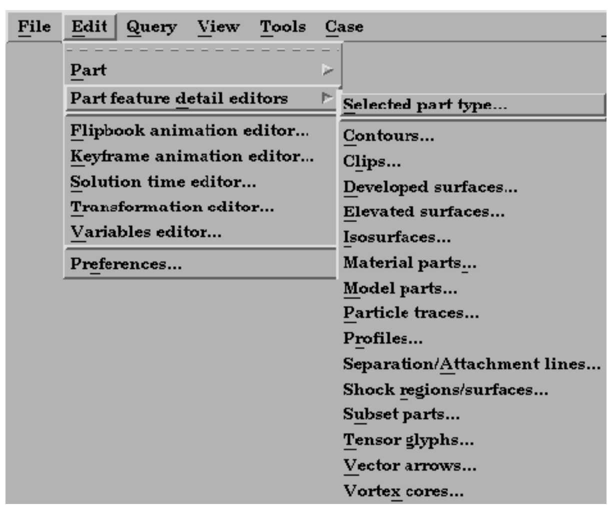


Figure 3-6
Menu selection of Feature Detail Editor

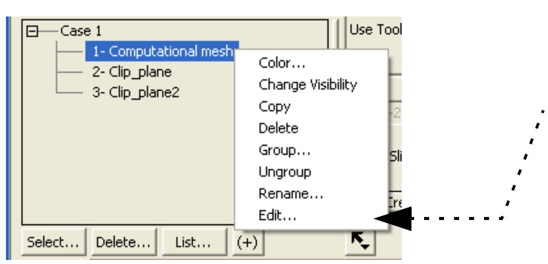


Figure 3-7
Easy Opening of the Feature Detail Editor

..

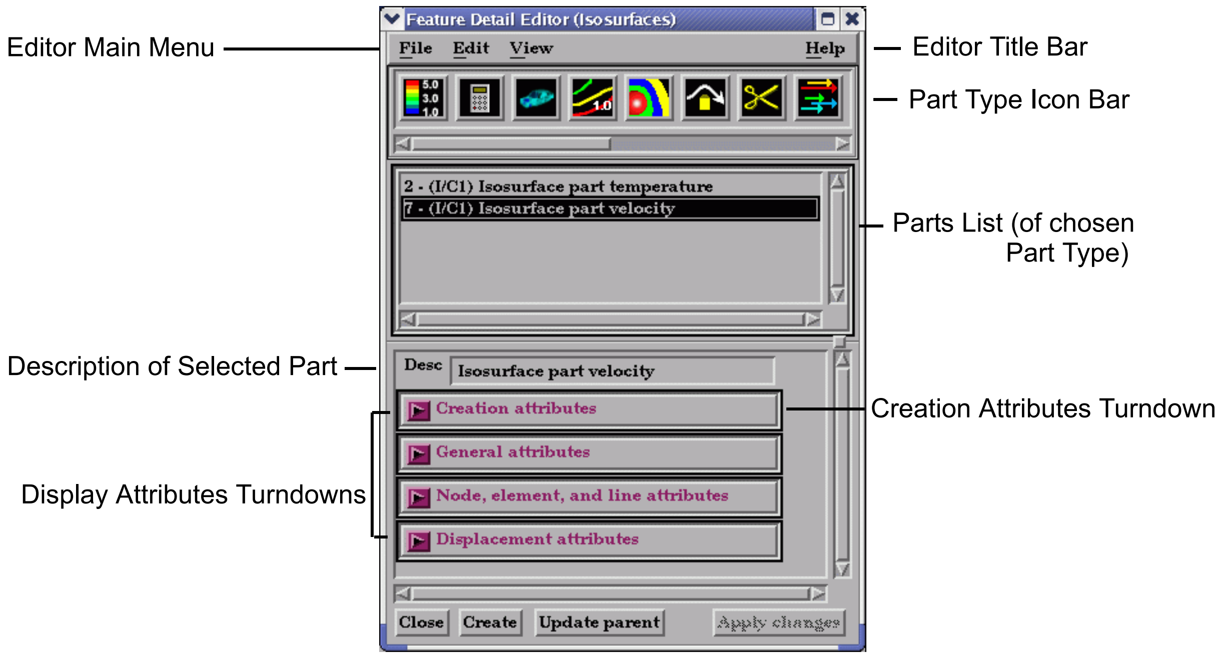


Figure 3-8
Feature Detail Editor (Isosurfaces)

Feature Detail Editor Main Menu

File Save/restore selected EnSight Constant values.

Edit Opens a pull down menu.

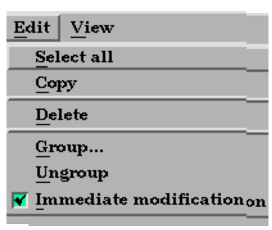


Figure 3-9
Feature Detail Editor Edit pull-down menu

Select All Selects all Parts in Feature Detail Editor Parts List. (see Section 3.4, Part Operations)

Copy Makes a copy of all selected Parts. (see Section 3.4, Part Operations), also (see How To Copy a Part)

Delete Deletes selected Parts. (see Section 3.4, Part Operations), also (see How To Delete a Part)

Group... Groups the selected parts into a new part and removes the original parts from the list.

Ungroup Extracts the original parts out of a group and removes the group part.

Immediate Modification Toggle Toggles on/off the immediate modification of Parts when individual changes are made to Attributes within the Feature Detail Editor. Default is on. By toggling off, you can make several changes within the Feature Detail Editor and then apply them all at one time by clicking the Apply Changes button.

Help

Detail Editor opens up the How To Manual addressing the usage of the Feature Detail editor.
Selected Entity opens up the How To Manual with an introduction to part creation.

Part Type Icon Bar

The Feature Detail Editor is initially opened from EnSight's Main Menu (or by double clicking a Part creation icon in the Feature Icon Bar) and the Feature Detail Editor's Parts List contains all Parts of the type named in the Editor's Title Bar. The type of Parts in the Feature Detail Editor's Parts List may be changed by clicking on the appropriate icon in the Feature Detail Editor's Part Type Icon Bar. The figure below describes the four types of choices available.

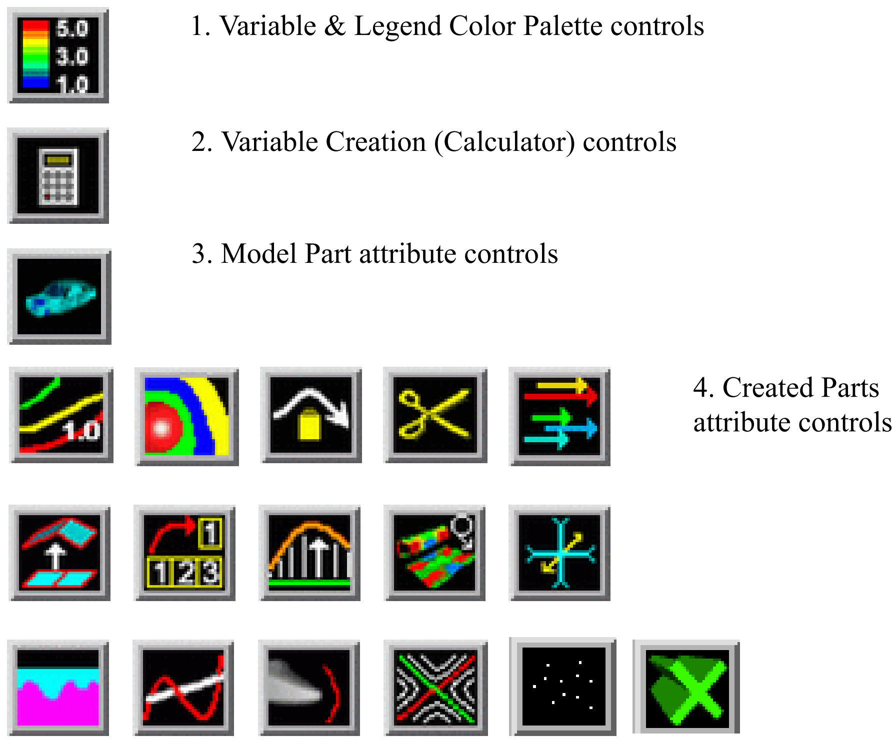
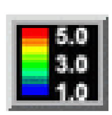


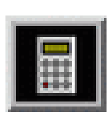
Figure 3-10
Feature Detail Editor Part Type Selection Icons

Variable & Legend Color Palette Icon



Click this icon to edit color controls. See Section 4.1 Variable Selection and Activation for further discussion.

Variable Creation (Calculator) Icon



Click this icon to use the calculator to create new variables for the parts selected in the Main Parts window.

Creation Attributes

Creation Attributes are “specific” attributes used to create (or modify) model and created Parts.

Model Parts

Creation Attributes for model parts consist of geometry scaling options (including server-side displacements) for unstructured and structured parts, and updating of I,J,K ranges for structured parts. Geometry scaling can be accomplished with a scale factor which will be applied to the model coordinates and/or a scale factor times a nodal variable. Updating the I,J,K node range attributes of the selected block structured Model Parts or the geometry scaling will cause proper updating of all dependent parts and variables.

Access: Main Menu > Edit > Part Feature Detail Editors > Model Parts...

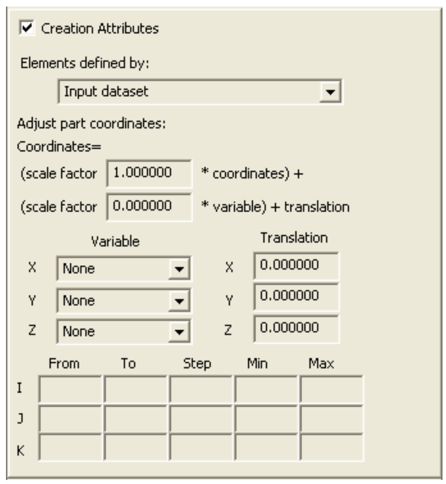


Figure 3-11
Feature Detail Editor (Model) Creation

Elements defined by:

Input dataset - This is the normal case where elements are defined in the dataset. No meshing is done.

Meshing the points to create a 3D mesh - The part’s nodes will be used as input into a volume mesher. Resulting in 3D elements for a volume, or 2D coplanar elements.

Meshing the points to create a 2D mesh - The part’s nodes will be used as input into a volume mesher. Resulting in the elements forming the convex hull.

Note: There are a few formats that will not allow you to return to the input dataset elements once you have meshed the part. Most do. For these few (ABAQUS fil, ansys, ESTET, FIDAP Neutral, Fluent Universal, and N3S), you can change between the 2D and 3D meshing options, but you need to delete the part and reload it, if you desire the part back to the input elements.

Adjust part coordinates:

The coordinates of the selected parts will be scaled and translated by the formula shown in the dialog. It is possible to apply a simple scale factor, and/or to apply a scaled nodal displacement vector variable (just choose the same vector variable for each pulldown and it will use the correct component). In fact each coordinate direction can be according to a different model scalar variable if desired. This works only with model variables, not computed variables. This is where “server-side” displacements can be used - which has the advantage of being able to properly query and compute on the displaced geometry of the model.

If you want to scale the model coordinates visually only, then you can use the transform editor and choose the scaling option and visually scale the geometry in the three orthogonal directions, and do this separate for each direction.
(see How To Display Displacements)

Using Node Ranges

IJK From These fields specify the desired minimum interval value in the respective IJK component direction of the Model Part.

IJK To These fields specify the desired maximum interval value in the respective IJK component direction of the Model Part.

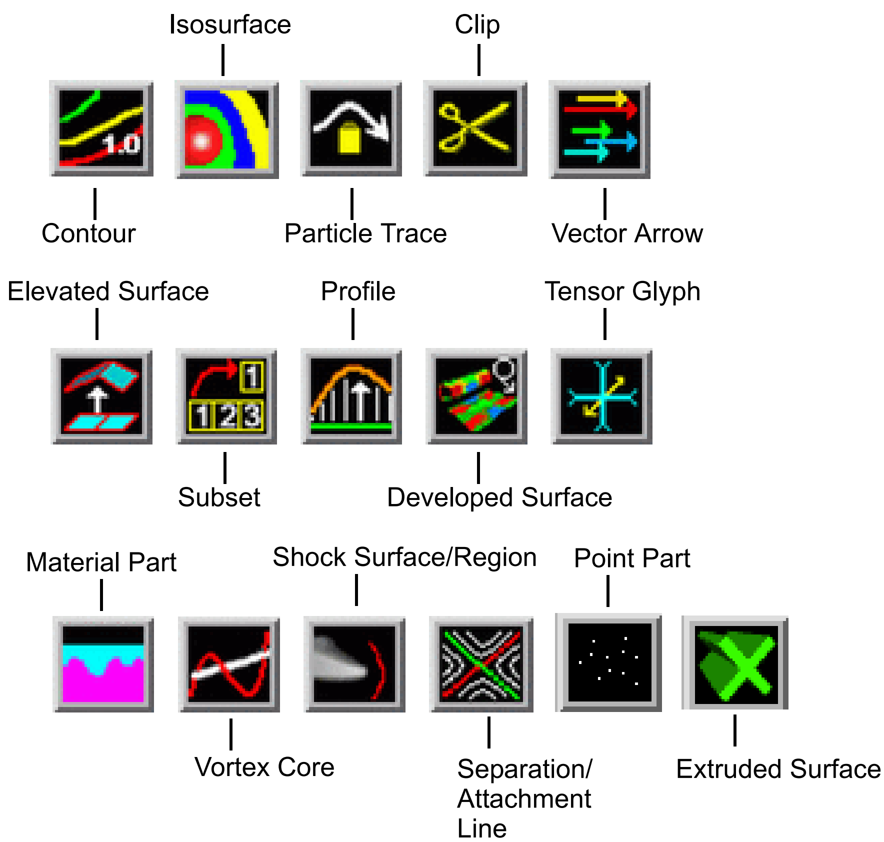
IJK Step These fields specify the desired interval stride value in the respective IJK component direction of the Model part.

IJK Min These fields verify the minimum interval limit in the respective IJK component direction of the Model part.

IJK Max These fields verify the maximum interval limit in the respective IJK component direction of the Model part.

(see How To Create IJK Clips)

Created Parts



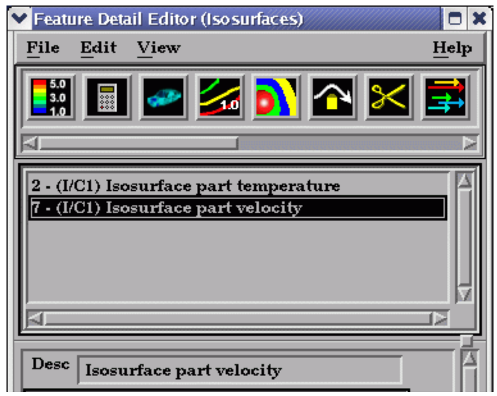
See the appropriate Section in Chapter 7 for a description of the Creation Attributes section.

- **(see Section 7.7, Contour Create/Update)**
- **(see Section 7.8, Isosurface Create/Update)**
- **(see Section 7.9, Clip Create/Update)**
- **(see Section 7.10, Vector Arrow Create/Update)**
- **(see Section 7.11, Particle Trace Create/Update)**
- **(see Section 7.12, Subset Parts Create/Update)**
- **(see Section 7.13, Profile Create/Update)**
- **(see Section 7.14, Elevated/Offset Surface Create/Update)**

- (see Section 7.15, Vortex Core Create/Update)
- (see Section 7.16, Shock Surface/Region Create/Update)
- (see Section 7.17, Separation/Attachment Lines Create/Update)
- (see Section 7.21, Developed Surface Create/Update)
- (see Section 7.20, Tensor Glyph Parts Create/Update)
- (see Section 7.21, Developed Surface Create/Update)
- (see Section 7.22, Point Parts Create/Update)
- (see Section 7.23, Extrusion Parts Create/Update)

Part Description Field

The description of model and created parts can be edited in the Desc field of the Feature Detail Dialgog. The limit for this description is 49 characters.



General Attributes

General Attributes are “general” in that: (a) all Parts have them, and (b) they can’t be neatly categorized into any other attribute type. Like all Part attributes, they are set individually for each Part.

Access: Main Menu > Edit > Part Feature Detail Editors > (Isosurfaces..., etc.) > General Attributes

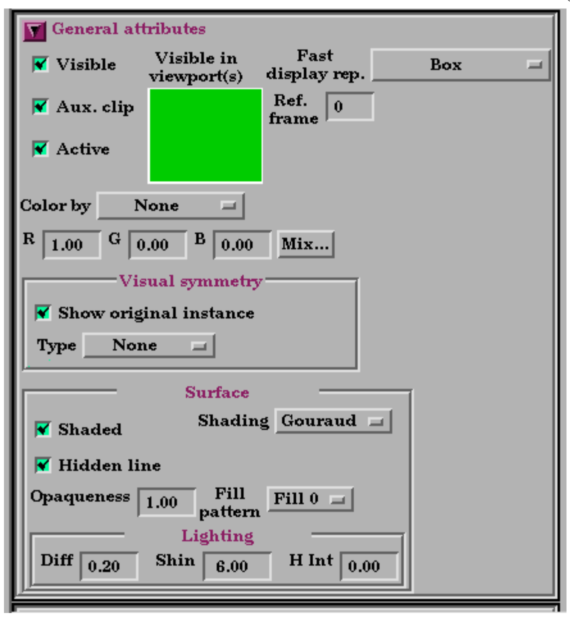


Figure 3-12
Feature Detail Editor General Attributes Area

Visible Toggle — Toggles-on/off whether Part is visible on a global basis (in the Graphics Window or in all viewports). (Performs the same function as the Visibility Toggle in the Parts Mode Icon Bar). Default is ON.

Visible In Viewports — This small window allows you to control the visibility of the selected Part(s) on a per Viewport basis. Each visible viewport is shown. A green Viewport indicates that the selected Part(s) will be visible in this Viewport, while a black Viewport indicates that the selected Part(s) will not be visible. Change the visibility (black to green, green to black) by selecting a viewport with the mouse.

Fast Display Rep.

This button opens a pop-up menu button for the selection of the fast display representation used to display a part on the client. This attribute helps the display of complex data sets. The part's fast display representation displays according to whether the Fast Display Mode (located in the View Menu or on the desktop) is on or off and on the state of the Static Fast Display button located under Edit > Preferences..., Performance. For instance, when the Fast Display Mode is Off (default) the part displays according to its specified Element Representation. When on, the parts are displayed by the fast display representation. The fast display representation will only be used while performing transformations, unless the Static Fast Display option has been selected. The part detail representations are:

Box	a bounding (Cartesian extent) box of all part elements (default).
Off	display according to specified Element Representation.
Points	point cloud representation of the part.
Reduced poly	polygon reduced representation of the part.
Sparse Model	display a percentage of the model in each display box (only available when running in immediate mode, using the -no_display_list startup option). You control this percentage in the performance preferences. *Note, that it is useful for large models, but should probably not be used for small models*.

(see How To Set Global Viewing)

Visual Symmetry

This section contains controls which allow you to produce either rotational or mirror visually symmetric images of parts. In general, symmetry enables you to reduce the size of your analysis problem while still visualizing the "whole thing." Symmetry affects only the displayed image, not the data, so you cannot query the image or use the image as a parent Part. However, you can create the same effect by creating dependent Parts with the same symmetry attributes as the parent Part.

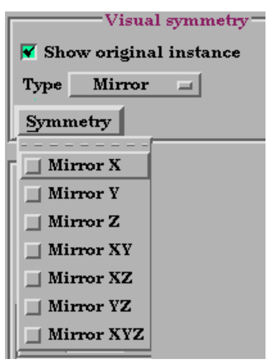


Figure 3-13
Mirror Visual Symmetry

Mirror visual symmetry includes the ability to toggle-on/off the display of a mirror image of Parts (which are selected in the Feature Detail Editor's Parts List) in each of the seven other quadrants of the Part's local frame. It also allows you to turn on or off the original (non-symmetric) part representation. It performs the same function as the Visual Symmetry Pull-down Icon in the Part Mode Icon Bar. You can mirror the Part to more than one octant. If the Part occupies more than one quadrant, each portion of the Part mirrors independently. Symmetry works as if the local frame is Rectangular, even if it is cylindrical or spherical. The images are displayed with the same attributes as the Part. For each toggle, the Part is displayed as follows. The default for all toggle buttons is OFF, except for the original representation - which is ON.

Mirror X	octant on the other side of the YZ plane.
Mirror Y	octant on the other side of the XZ plane.

Mirror Z octant on the other side of the XY plane.
Mirror XY diagonally opposite octant on the same side of the XY plane.
Mirror XZ diagonally opposite octant on the same side of the XZ plane.
Mirror YZ diagonally opposite octant on the same side of the YZ plane.
Mirror XYZ octant diagonally opposite through the origin.
Show Original Instance the original part location.

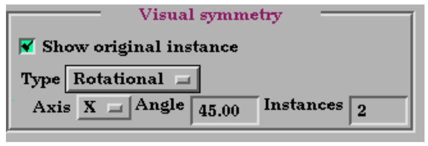


Figure 3-14
Rotational Visual Symmetry

Rotational visual symmetry allows for the display of a complete (or portion of a) “pie” from one “slice” or instance. You control this option with:

Axis X rotates about the X axis
Y rotates about the Y axis
Z rotates about the Z axis
Angle specifies the angle (in degrees) to rotate each instance from the previous
Instances specifies the number of rotational instances.
Show Original Instance show the original instance or not

(see How To Set Symmetry)

Aux Clip Toggle

Toggles-on/off whether Part(s) selected in the Part List of the Feature Detail Editor will be affected by the Auxiliary Clipping Plane feature, which enables you to make invisible that portion of each Part on the negative side of the current position of the Plane Tool. Performs the same function as the Part Mode: Auxiliary Clipping Toggle Icon. A Part with its Aux Clip attribute toggled-off will not be cut away. Default is ON. (see Auxiliary Clipping in Section 6.4, View Menu Functions).

Active Toggle

Toggles-on/off whether or not display of the Part automatically updates as the solution time changes. When visualizing transient data, you may wish to “freeze” a Part in time while other Parts continue to update. For example, you can create two identical vector-arrow Parts, toggle-off Active for one of them, change the time step of the display, and see how the vector arrows change from one time step to the other. Only the EnSight client Part is frozen, the EnSight server Part is kept current. Default is ON.

Ref. Frame

This field specifies which frame the Part is assigned to. Default is the frame of the Part’s parent Part (Frame 0 for original model Parts). Enter a different frame number in the field to change the assignment. Changing a Part’s frame causes the Part to be drawn in the new coordinate frame. Once assigned to a different frame, the Part will transform with that frame. The choice of frame does not affect variable values. The interpolated value of a variable at point 0,0,0 in Frame 0 is the same as at point 0,0,0 in Frame 1, even though the points may appear at different locations in the Main View Window.
(see Section 8.5, Frame Mode)

Color By

This button opens a pop-up menu for the selection of the variable color palette by which you wish to color the selected Part(s). Coloring a Part with a palette does not normally affect graphics performance while in line drawing mode, but Shaded Surface mode performance can become considerably slower. If you do not color by a palette (Color By > None), the Part will be displayed according to the color specified in the R, G, B fields. If you want to color Parts by palettes and want Shaded Surface mode, consider using the Static Lighting option (see Static Lighting in Section 6.4, View Menu Functions).

R G B

These fields allow you to specify a solid color for the selected Part(s) (applicable only if

Color By is None). Enter a numerical value from 0 to 1 for each component color (Red, Green, and Blue).

Mix... Opens the Color Selector dialog for the selection of a solid color for the selected Part(s) (applicable only if Color By is None). (see Section 8.1, Color Selector)

Surface

Shaded Surface Toggle: Toggles on/off surface shading for individual Parts. When global Shaded Surface has been toggled on for the Graphics Window display (from Main Menu > View > Shaded Surface or via the Global Shaded Surface Toggle in the View Mode Icon Bar), individual Parts can be forced to stay in line drawing mode using this toggle. Default is ON. (see Section 6.4, View Menu Functions)

Hidden Line Toggle: Toggles on/off hidden line representation for individual Parts. When global *Hidden Line* has been toggled on for the Graphics Window display (from Main Menu > View > Hidden Line or via the Global Hidden Line Toggle in the View Mode Icon Bar), individual Parts can be forced not to appear as Hidden Line representation using this toggle. (To have lines hidden behind surfaces, Parts must have surfaces, i.e. 2D elements) Default is ON. (see Section 6.4, View Menu Functions)

Shading: Opens a pop-up menu for selection of appearance of Part surface when Shaded Surface is on. Normally the mode is set to Gouraud, meaning that the color and shading will interpolate across the polygon in a linear scheme. You can also set the shading type to Flat, meaning that each polygon will get one color and shade, or Smooth which means that the surface normals will be averaged to the neighboring elements producing a “smooth” surface appearance.Not valid for all Part types. Options are:

Flat	Color and shading same for entire element
Gouraud	Color and shading varies linearly across element
Smooth	Normals averaged with neighboring elements to simulate smooth surfaces

Opaqueness: This field specifies the opaqueness of the selected Part(s). A value of 1.0 indicates that the Part is fully opaque, while a value of 0.0 indicates that it is fully transparent. Setting this attribute to a value other than 1.0 can seriously affect the graphics performance.

Fill Pattern: Opens a pop-up menu for selection of a fill pattern which can provide pseudo-transparency for shaded surfaces. Default is Fill 0 which uses no pattern (produces a solid surface), while Fill patterns 1 through 3 produce a EnSight defined fill pattern.

Lighting

Diff: This field specifies diffusion (minimum brightness or amount of light that a Part reflects). (Some applications refer to this as *ambient* light.) The Part will reflect no light if value is 0.0. If value is 1.0, no lighting effects will be imposed and the Part will reflect all light and be shown at full color intensity at every point. To change, enter a value from 0 to 1.

Shin: This field specifies shininess.You can think of the shininess factor in terms of how smooth the surface is. The larger the shininess factor, the smoother the object. A value of 0 corresponds to a dull finish and a value of 100 corresponds to a highly shiny finish. To change, enter a value from 0 to 100.

H Int: This field specifies highlight intensity (the amount of white light contained in the color of the Part which is reflected back to the observer). Highlighting gives the Part a more realistic appearance and reveals the shine of the surface. To change, enter a value from 0 to 1 with larger values representing more white light. Will have no effect if Shin parameter is zero.

(see How To Set Attributes)

Troubleshooting Surface Attributes and Lighting

Problem	Probable Causes	Solutions
Part not in Shaded Surface mode	Global Toggle not on, or if on, Shaded Surface is turned off for the Part in the Feature Detail Editor	Turn on Shaded Surface toggle from View menu of Main Menu or turn and make sure Shaded Surface is turned on for the Part in the Feature Detail Editor.
	Part contains only 1D elements	No Solution
Part appears not to have any lighting.	Diffuse light intensity too high	Lower the Diff value.

Node, Element, and Line Attributes

Each Part's Node, Element, and Line attributes control the representation of the Part on the client, and how nodes, elements, and lines are displayed.

Access: Main Menu > Edit > Part Feature Detail Editors > Node, Element, and Line Attributes

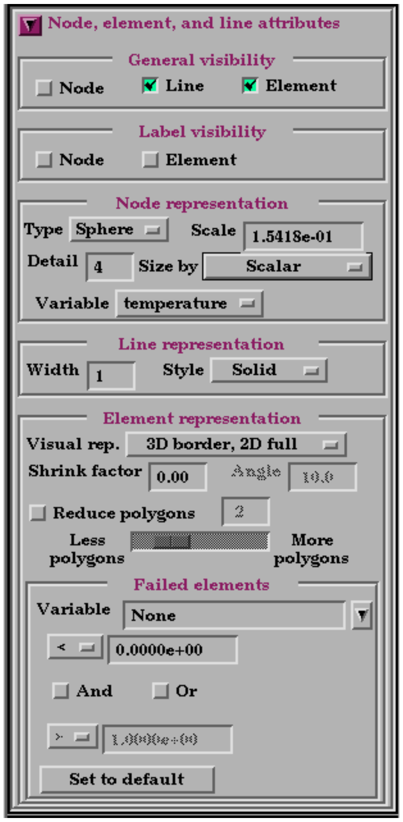


Figure 3-15
Feature Detail Editor Node, Element, and Line Attributes Area

General Visibility

Node Toggle: Toggles-on/off display of Part's nodes whenever the Part is visible. Default is OFF.

Line Toggle: Toggles-on/off display of line (1D) elements in the client-representation whenever the Part is visible. Default is ON.

Element Toggle: Toggles-on/off display of 2D elements in the client-representation whenever the Part is visible. Note that 3D elements are always represented as 2D elements on the client. Default is ON.

Label Visibility

Node Toggle: Toggles-on/off display of Part's node labels (if they exist) whenever the Part is visible. Only model Parts may have node labels. Default is OFF.

Element Toggle: Toggles-on/off display of Part's element labels (if they exist) whenever the Part is displayed in Full visual representation. Only model Parts may have element labels. Default is OFF.

Node Representation

Type: Opens a pop-up menu for the selection of symbol to use when displaying the Part's nodes. Default is Dot. Options are:
Dot to display nodes as one-pixel dots.
Cross to display nodes as three-dimensional crosses whose size you specify.
Sphere to display the nodes as spheres whose size and detail you specify.

Scale — This field is used to specify scaling factor for size of node symbol. Values between 0 and 1 reduce the size, factors greater than one enlarge the size. Not applicable when node-symbol Type is Dot. Default is 1.0.

Detail — This field is used to specify how round to draw the spheres when the node-symbol type is Sphere. Ranges from 2 to 10, with 10 being the most detailed (e.g., roundest spheres). Higher values take longer to draw, slowing performance. Default is 2.

Size By — Opens a pop-up menu for the selection of variable-type to use to size each node-symbol. For options other than Constant, the node-symbol size will vary depending on the value of the selected variable at the node. Not applicable when node-symbol Type is Dot. Default is Constant. Options are:

Constant sizes node using the Scale factor value.
Scalar sizes node using a scalar variable.
Vector Mag sizes node using magnitude of a vector variable.
Vector X-Comp sizes node using magnitude of X-component of a vector variable.
Vector Y-Comp sizes node using magnitude of Y-component of a vector variable.
Vector Z-Comp sizes node using magnitude of Z-component of a vector variable.

Variable — Selection of variable to use to size the nodes. Activated variables of the appropriate Size By type are listed. Not applicable when node-symbol Type is Dot or Size By is Constant.

Line Representation

Width — Specification of width (in pixels) of line elements and edges of 2D elements whenever they are visible. Range is from 1 to 20. Default is 1. Line widths other than 1 are not available on all hardware. This performs the same function as the Part Line Width Pulldown Icon in Part Mode.

Style — Selection of style of line when lines are visible. Default is Solid. Options are:

Solid
Dotted
Dot-Dash

Element Representation

Visual Rep. — Selection of representation of Part's elements on the client. Saves memory and time to download.

3D border, 2D full represents the Part's 3D elements in Border representation, the Part's 1 and 2D elements in Full representation. The result is the outside surfaces of the Part are displayed along with all bar elements.

3D feature, 2D full represents the Part's 3D elements in Feature representation, the Part's 1 and 2D elements in Full representation. The result is the outside sharp edges of the Part are displayed along with all bar elements.

3D nonvisual, 2D full represents the Part's 3D elements in non visual representation, the Part's 1 and 2D elements in Full representation. The result is all the 1 and 2D elements from 2D parts are displayed.

Border represents the Part's 3D elements with 2D elements corresponding to unshared element faces, the Part's 2D elements with 1D elements corresponding to the unshared edges, and the Part's 1D elements as 1D elements. The result is the outside faces and edges of the Part's elements.

Feature Angle first runs the 3D border, 2D full representation to get a list of 1 and 2D elements. The 1D elements and all non-shared 2D edges will be shown, but only the shared edges above the Angle value will be shown. The result consists of 1D elements visualizing the sharp edges of the Part.

Bounding Box represents all Part elements as a bounding box surrounding the Cartesian extent of the elements of the Part.

	Full represents all faces of the Part's 3D elements, and all the 1 and 2D elements. *Non Visual* means the Part exists on the server, but is not loaded on the client. Not Loaded Parts may be used as parent Parts, but do not exist on the client.
Shrink Factor	Specification of scaling factor by which to shrink every element toward its centroid. Enter the fraction to shrink by in range from 0 to 1. Default is 0.0 for no shrinkage.
Angle	Specification of lower limit for not displaying shared edges in Feature Angle Representation. Value is in degrees.
Load Points and Normals only	Loads only vertex information and normals for the element representation given to the client. Useful for very large models.
Reduce Polygons	Lower the polygon density used to represent the part. Useful for very large models. Toggle on, then type in a value to reduce by, or slide the slider.
Failed Elements	
Variable	Elements are removed from display on the client and from calculation on the server using the named variable and the threshold operator(s) (< , > , = , !=) and their relationship (logical 'and' or logical 'or'). Applies only to model parts.
< , > , = , !=	Elements are removed from display on the client and from calculation on the server using the above named variable and the threshold operator(s) (< , > , = , !=) and their relationship (logical 'and' or logical 'or').

(see How To Set Attributes and How To Display Labels)

Troubleshooting Node, Element and Line Attributes

Problem	Probable Causes	Solutions
After changing to Feature Angle representation, the Part is not shown.	Angle value is to large	Set Angle to smaller value.

Displacement Attributes

Displacement Attributes specify how to displace the Part nodes based on a nodal vector variable. Each node of the Part is displaced by a distance and direction corresponding to the value of a nodal vector variable at the node. The new coordinate is equal to the old coordinate plus the vector times the specified Factor, or:

$C_{new} = C_{orig} + \text{Factor} * \text{Vector},$

where C_{new} is the new coordinate location, C_{orig} is the coordinate location as defined in the data files, Factor is a scale factor, and Vector is the displacement vector.

You can greatly exaggerate the displacement vector by specifying a large Factor value. Though you can use any vector variable for displacements, it certainly makes the most sense to use a variable calculated for this purpose. Note that the variable value represents the *displacement* from the original location, not the *coordinates* of the new location.

Access: Main Menu > Edit > Part Feature Detail Editors > Displacement Attributes

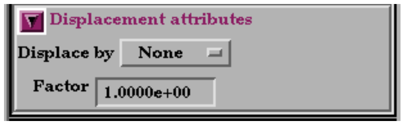


Figure 3-16
Feature Detail Editor Displacement Attributes Area

Displace by — Opens a pop-up menu for selection of vector variable to use for displacement (or None for no displacement). Variable must be a nodal vector and be activated.

Factor — This field is used to specify a scale factor for the displacement vector. New coordinates are calculated as: $C_{new} = C_{orig} + Factor*Vector$, where C_{new} is the new coordinate location, C_{orig} is the original coordinate location as defined in the data file, Factor is a scale factor, and Vector is the displacement vector. Note that a value of 1.0 will give you “true” displacements.
(see How To Display Displacements)

Troubleshooting Displacement Attributes

Problem	Probable Causes	Solutions
Displacement not visible	Displace By attribute set to None for Part that is not displacing.	Set the Displace By attribute
	Factor value too small.	Specify a larger Factor.
	Using a per-element variable.	Make sure in calculator that variable is a per-node vector.

IJK Axis Display Attributes

All Model and clip parts will have these attributes shown, but they only apply to to those model and clip parts which are structured.

Access: Main Menu > Edit > Part Feature Detail Editors > IJK Axis Display Attributes

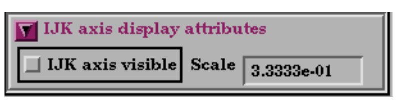


Figure 3-17
Feature Detail Editor IJK Axis Display Attributes Area

IJK Axis Visible — Toggle on to display an IJK axis triad for the part. IJK axis triad only visible when part is visible.

Scale — The scale factor for the IJK Axis triad.

(see How To Set Attributes)

Troubleshooting IJK Axis Display Attributes

Problem	Probable Causes	Solutions
IJK Axis not visible	IJK Axis Display toggle not on for part of interest.	Toggle it on.
	Scale value too small.	Specify a larger scale.

Problem	Probable Causes	Solutions
	Part is not a structured part.	No IJK axis possible for this part.
	Part is not visible	Toggle on part visibility

Dialog Action Buttons

Create — Clicking this button creates a new Part using attributes currently selected/specified in the Feature Detail Editor. This performs the same function as the Create button in the Quick Interaction Area Editor for each type of created Part. Clicking Create updates the Graphics Window and adds the new Part to the Main Parts List and to the Parts List in the Feature Detail Editor for this type of Part. Not applicable for model Parts or discrete Particles.

(see Introduction to Part Creation)

Update Parent — Clicking this button assigns the Part which is currently selected in the Main Parts List as the new parent Part of the created Part(s) which is(are) currently selected in the Feature Detail Editor's Parts List.

Apply Changes — Clicking this button applies all changes that have been made within the Feature Detail Editor all at once if Immediate Modification has been toggled off above in the Feature Detail Editor's Edit pull-down menu. If Creation attributes have been changed, the Part will be regenerated.

3.4 Part Operations

This section will describe the Part operations accessible through “Edit > Part” in the Main Menu and “Edit” in the Feature Detail Editor Menu. These include *Select All, Select, Delete*, *Assign to Single or Multiple Viewports, Group, Ungroup*, *Copy*, *Cut*, *Extract*, and *Merge*.

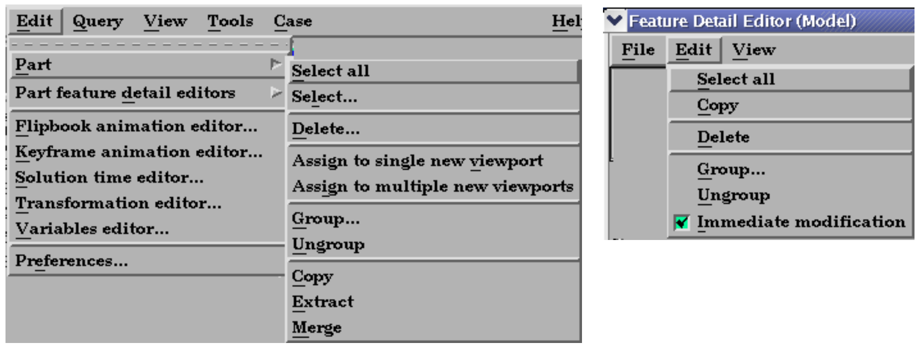


Figure 3-18
Part Operation Selection Menus

Select All

Choosing this from the Main Menu > Edit >Part pull-down, selects all Parts in the Main Parts List. Choosing this from the Edit pull-down in the Feature Detail Editor Menu selects all Parts in the Feature Detail Editor Parts List.
Access: Main Menu > Edit > Part > Select All
Feature Detail Editor Menu > Edit > Select All

(see How to Select Parts)

Select ...

Choosing this from the Main Menu > Edit >Part pull-down, opens the Select Part(s) By Keyword dialog.

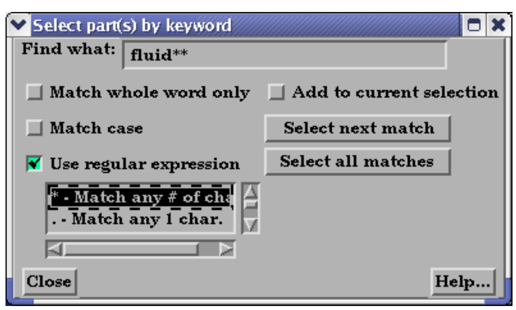


Figure 3-19
Select Part(s) By Keyword dialog

Find What

This field is used to specify the keyword or regular expression to compare (match) against Part names.

Match Whole Word Only Toggle

When on, the entire Part name must match the keyword or regular expression.
When off, a Part name will be selected if only a substring of the Part name matches.

Match Case Toggle

When on, the comparison is case sensitive. When off, case is ignored.

Use Regular Expression Toggle

When on, special characters in the keyword will be used to define a regular expression.
When off, any special characters will be treated as a regular character during comparison.

Special Character

Contains a list of special characters available to create a regular expression. Selecting an

Selection List — item from the list will insert the special character into the “Find What” text field at the cursor location.

*	Match any number of characters in the part name
.	Match any one character in the part name
~	Match part names that do not match the specified search criteria.
|	Separates multiple search keywords or regular expressions. (extra spaces are not allowed around “|”)

Examples: Find What: abc*xyz Match Whole Word Only: On
Select any Part who’s name starts with “abc” and ends with “xyz”

Find What: tom|jerry Match Whole Word Only: OFF
Select all Part s who’s names contain the string “tom” and/or the string “jerry”

Add to Current Selection Toggle — When on, any matching Part names will be added to the list of Part names currently selected. When off, only the matching Part names will be selected.

Select Next Match — Selects the next Part name which matches the keyword or regular expression.

Select All Matches — Selects all Part names which match the keyword or regular expression.

Delete… — If chosen from the Main Menu > Edit > Part pull-down, deletes all selected Parts in the Main Parts List after you have confirmed in a pop-up dialog that you wish to do so. If chosen from the Edit pull-down in the Feature Detail Editor Menu, deletes all selected Parts in the Feature Detail Editor Parts List after you have confirmed in a pop-up dialog that you wish to do so. If model Parts are deleted, they are no longer available for the current session. Parts dependent upon selected Parts will also be deleted or modified
Access: Main Menu > Edit > Part > Delete...
Feature Detail Editor Menu > Edit > Delete...

(see How to Delete a Part)

Assign to Single New Viewport — Creates a new viewport and assigns all of the selected parts to the new viewport. The new viewport will be 2D if all of the selected parts are 2D and lie on the same plane.

Assign to Multiple New Viewports — Creates a new viewport for each of the selected parts. Each new viewport will show one part only. If the part is 2D, the viewport will be 2D. Further, if the part assigned to the new viewport is a XYZ or IJK clip or an isosurface, annotation will be created in the lower left corner of the viewport indicating the value of the clip or iso.

Group/Ungroup — The group operation is used to collect any number of parts into a set which can be modified and utilized as one entity. The operation is non-destructive and reversible, and is used solely as a convenience to the user in order to organize a large number of parts.

Any attribute modification to a grouped part affects each of the parts in the group. Similarly, if a grouped part is used as a parent part, each part in the group is used as a parent in the creation process.
When group is selected, the dialog shown in the figure below will appear. A part name must be input in order to complete the grouping operation.

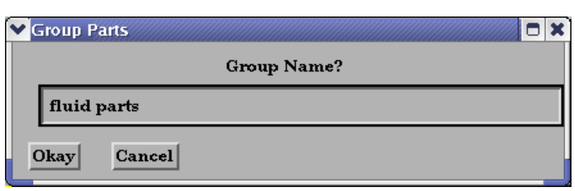


Figure 3-20
Group Parts Dialog

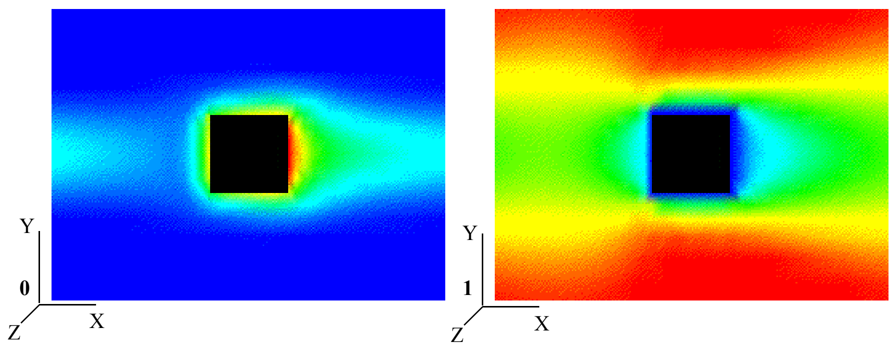
Only parts of the same type and case can be grouped together. Further, groups can not contain other part groups.

(see How To Group Parts)

Copy

If chosen from the Main Menu > Edit > Part pull-down, makes a copy of selected Part(s) in the Main Parts List. If chosen from the Edit pull-down in the Feature Detail Editor Menu, makes a copy of selected Part(s) in the Feature Detail Editor Parts List.

The Copy operation creates a dependent copy of another (original) Part. The Copy is created on the Client and its existence is not known to the EnSight Server process. A Copy shares geometric data and variable data with the original Part. (This type of Part is sometimes called a “shallow copy”.)



Original Clip Part - colored by temperature

Copy of Clip Part, assigned to new Frame, translated to right, and colored by velocity at the same time step

Figure 3-21
Part Copy Example

relationships

The relationship between a Model Part and a Copy made from a Model Part will be one of original and copy. That is, the Model Part will not be a Parent to the Copy as it is to a Created Part such as a clip.

The relationship between a Created Part and a copy made from it will also be one of original and copy since the Copy will initially regard as its Parent the same Part that the original Created Part regards as its Parent. The Parent of individual Created Parts can of course be reassigned (using the Update Parent button at the bottom of the Feature Detail Editor) but the Parent of a Created Part Copy can Not be reassigned.

A copy can be used as a Parent Part for Parts created since the create operation will operate on the original Part.

attributes

The initial attributes assigned to a Copy are the same as those of the original Part at the time of copying. All attributes for the Copy except Element Representation (3D border, 2D full, border, Feature Angle, etc.) can be changed. The Element Representation of a Copy cannot be changed independently; a change in Element Representation of the original changes the copy as well.

description

The description of the new Copy will be the same as the original Part with the suffix “-COPY” added (of course, you can change this description in the Desc field in the Feature Detail Editor).

frame assignment

A new frame is automatically created for each newly created Copy and the Copy is assigned to the new frame so that it can easily be moved with a local transformation. The location of the original Part and the Copy will initially coincide as well. Like all Parts, Copies of Parts can be reassigned to different frames in the General Attributes Section of the Feature Detail Editor (for that type of Part).

usefulness

One of the most useful purposes for copies is a separation allowing for the side-by-side display of different attributes (shown in Figure 3-11). Since all attributes except Element Representation can be different, the original and the copy can be displaying different variables, different displacements, etc.
(see How To Copy a Part)

Extract

Extracts selected Part(s) into a new, true Part, *using the Part representation in effect at the time* (full, border, or feature). If more than one Part is selected, then they are joined into a single Part. If more than one Part is selected when extract is invoked, then all will have their extracted geometry joined into a single new Part. The new Part is assigned to Frame 0.

description

The Extract option is closely tied to Element Representation. It creates a new Part using the geometry of the current representation (what you see is what you get). Extracted Parts which are in Full Representation are actual copies of the original, but extracted Parts which are in Border Representation are only the shell or boundary of the original. Extract is often used with the Save Geometric Entities feature to save extracted Parts (and not the originals) into a smaller set of data. It is also used to create hollow Parts from solid Parts to be able to look inside a solid Part after cutting it open with the Cut feature.

(see How To Extract Part Representations)

Merge

If more than one Part is selected, the Merge operation creates a new model Part on the Server host that is a combination of all selected. If only one Part is selected when Merge is invoked, then a new Part is created on the Server host that is identical but fully independent from the original Part (Note that this type of "copy" does not have the restriction on Element Representation that Part Copy does, - *all* Attributes can be reassigned - but it requires considerably more memory because it does not share the geometry with the original but now has its own copy of the geometry). The merge operation creates a new Part. The new Part is assigned the default Display Attributes and is also assigned to Frame 0.

(see How To Merge Parts)

3.5 Part List Shortcuts (Right-click)

This section will describe Part shortcuts available via the right-click in the Main Part List. These shortcuts operate on the Part(s) that are highlighted in the Main Part List and *not on the Part(s) that are right-clicked*.

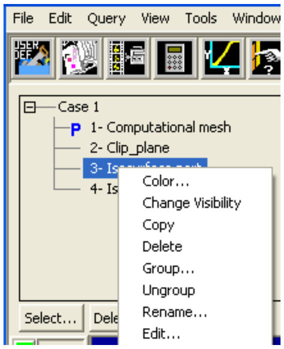


Figure 3-22
Part List Right-Click Options

Color	Brings up the "Part color, lighting, and transparency" dialogue. Choices in this dialogue will affect the Part(s) that are highlighted in the Main Part List and *not on the Part(s) that are right-clicked.*
Change Visibility	Toggles visibility of the Part(s) that are highlighted in the Main Part List and *not on the Part(s) that are right-clicked.*
Copy	Makes a "shallow copy" (only on the client) of the Part(s) that are highlighted in the Main Part List and *not on the Part(s) that are right-clicked.*
Delete	Deletes the Part(s) that are highlighted in the Main Part List *not the Part(s) that are right-clicked.*
Group/Ungroup	Groups / Ungroups the Part(s) that are highlighted in the Main Part List and *not on the Part(s) that are right-clicked.*
Rename	Renames the Part(s) that are highlighted in the Main Part List and *not on the Part(s) that are right-clicked*. Note that if more than one Part is selected, each part will be given the name typed in the Rename Dialog, but parts after the first will have the new name followed by a sequential number to make sure the Part names are unique.
Edit	Opens up the Feature Detail Editor for the Part Type of the first highlighted part in the Part List.

3.6 Part Graphics Window Shortcuts (Right-click)

This section will describe Part shortcuts available via the right-click in the Graphics window on a Part geometry. The Part geometry will change appearance indicating it is selected, and the part list will update indicating it is selected. A pulldown list of shortcuts will appear that is context-sensitive to the part type selected. These shortcuts operate on the Part(s) that is/are selected. These right-click shortcuts can be disabled by altering the mouse behavior in EnSight preferences as follows: Edit>Preferences, Mouse and Keyboard and change the single click settings to None.

The options available in the context sensitive right-click pulldown will depend on the part type selected, the sitee and user-options installed, and the version of EnSight. The figure below is for illustrative purposes only.

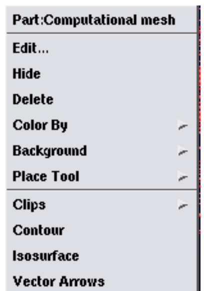


Figure 3-23
Part Graphics Window Right-Click Options

Edit	Brings up the Part Feature Detail Editor Window
Hide	Sets the Part visibility to OFF
Delete	Deletes the Part without an 'Are you Sure?' and cannot be undone
Color By	Colors the Part by Variable, Constant Color, Black, White, Grey, or Transparent
Background	Change view, modify viewports, select parts, full screen, add text, change background color, or create an image
Place Tool	Place the cursor, line or plane tool
Clips	Create clip using selected part(s) as a parent
Contour	Create a contour using selected part(s) as a parent
Isosurface	Create an isosurface using selected part(s) as a parent
Vector Arrows	Create vector arrows using selected part(s) as a parent

3.7 Part Shortcuts (Click-and-Go)

If you click on a part and a green cross hair appears, then this part has activated click and go. Click on the green cross hair and then drag it left and right or up and down to dynamically cause, for example, the the isosurface value change, or the clip location change, or the vector arrow scaling change, or the contour density change. Click and Go is available for any part that you click on and get a click and go indicator (among them Vector Arrow Parts, Contour Parts, Isosurface parts, Clip Parts). This click-and-go behavior can be disabled by altering the mouse behavior in EnSight preferences as follows: Edit>Preferences, Mouse and Keyboard and change the single click settings to None. You might want to disable this behavior if, for example, you were dealing with very large datasets with parts whose parameters cannot be changed interactively.

.

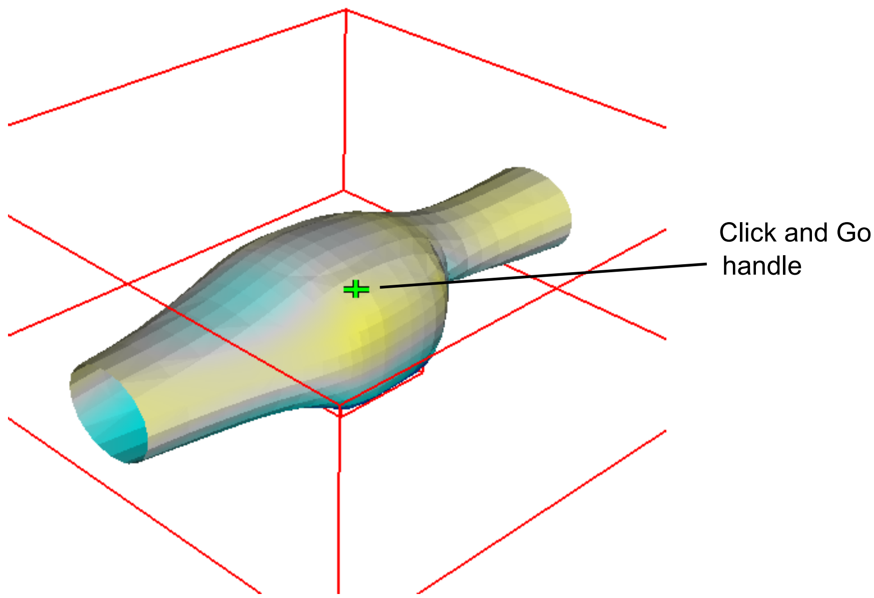


Figure 3-24
Click and Go Handle

5 GUI Overview

The purpose of this Chapter is to provide a brief overview of the EnSight 9 GUI.

"**Desktop**" - refers to the upper level of the GUI. It contains the following areas:

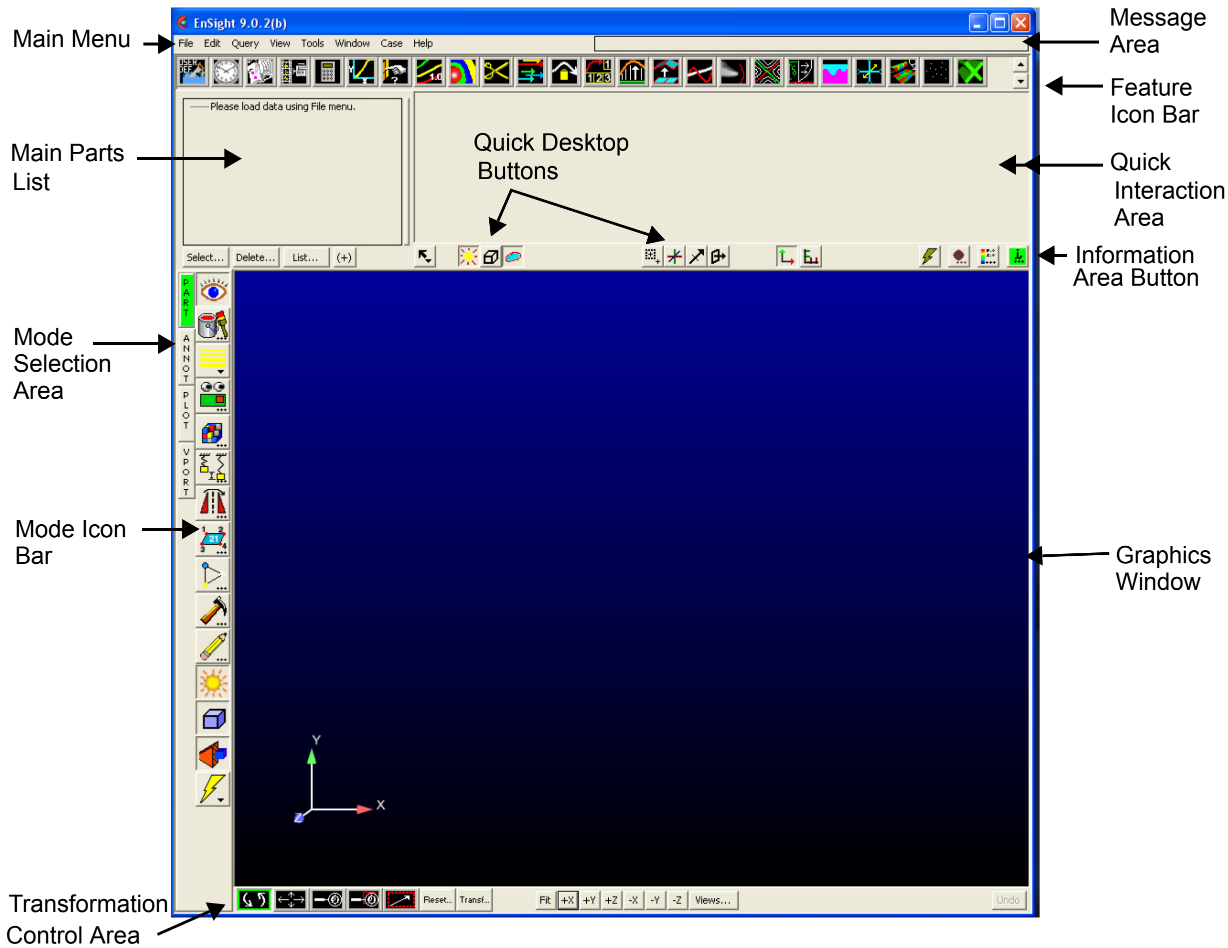


Figure 5-1
EnSight 9 Start-Up GUI

When EnSight first comes up, the Graphical User Interface should appear approximately as shown. The different sections of the GUI are used for specific purposes.

Main Menu

In addition to providing access to high-level features such as Command File Creation/Editing/Reading, Results Data Reading, File Printing, Saving/Restoring a session, and Quitting, the Main Menu provides access to often-used postprocessing features such as editing, querying variable data, part appearance adjustment, and tool visibility.

Chapter 6 contains a complete description of each section of the Main Menu. (see Chapter 6, Main Menu)

Main Parts List Default The Default Main Parts List contains the number and names of all parts that have been read in from your results data (model parts) or created within EnSight (created parts) nested within their Case description.

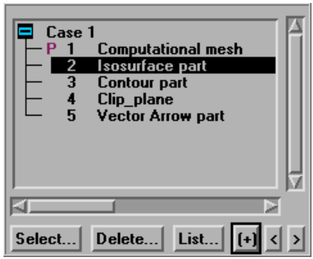


Figure 3-2
Default Main Parts List

Main Parts List Abbrev Shown below is the Main Parts List Window if you click on the List Button pulldown, toggle "Show abbreviated details". Notice that the Part List will contain more information. Displayed are a case number, ID number, visibility attribute, shaded attribute, part symbol, representation, node and element id status, and a Part name (description).

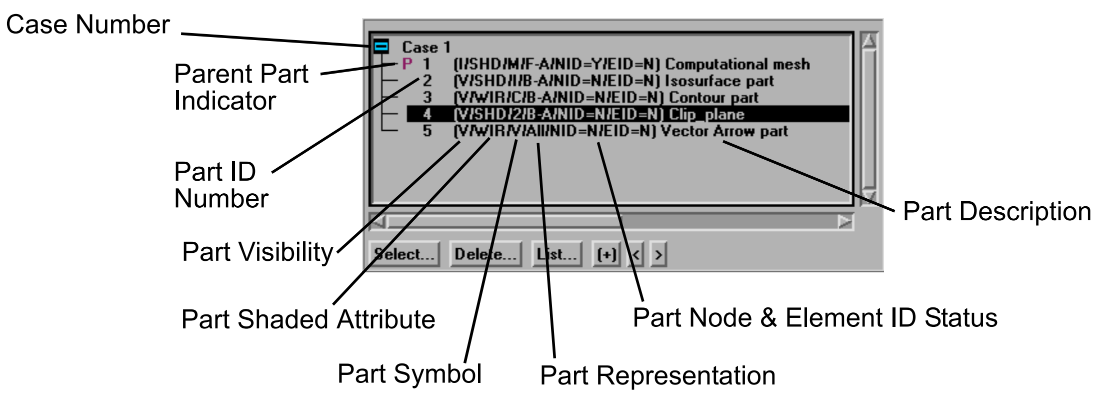


Figure 3-3
Abbreviated Parts List

You will find sub-sets of this Main Parts List in the Feature Detail Editor for each type of part. For example, the Feature Detail Editor (Isosurface) will contain a parts list of only isosurface parts.

For a complete description of the Main Parts List as well as a detailed discussion about Part selection, editing, and operations thereon:
(see Section 3.1, Part Overview)

Feature Icon Bar This Icon Bar provides rapid access to things such as new part creation, 2D plot creation, data querying, time step control, flipbook animation and keyframe animation. Clicking once on an icon opens its associated editor in the Quick Interaction Area.

Double clicking on a new part creation Icon (contours, isosurfaces, particle traces, clips, vector arrows, elevated surfaces, profiles, developed surfaces) will open the Feature Detail Editor for that type of created part.

Chapter 7 contains a detailed explanation of the features in the Quick Interaction Area which are available through each of the Icons.

(see Chapter 7, Features)

Mode Selection Area

The Mode Selection Area contains four to five buttons which allow you to choose which of the “Modes” you wish to work in. The Mode selected will not only determine which icons you see in the Mode Icon Bar but also the way in which you work within the Graphics Window, The five possible Modes are:

- **Part Mode** for the specification of attributes for specific parts
- **Annot Mode** for the addition and editing of annotation lines, text, and logos to the Graphics window as well as the editing of Variable legends
- **Plot Mode** for the creation and specification of attributes for 2D Variable plots
- **VPort Mode** for the creation and control of additional viewports within the Graphics Window
- **Frame Mode** for the creation and specification of attributes for additional frames of reference within EnSight. *By default this mode is not displayed. If needed, this mode can be turned on under Edit > Preferences... General User Interface - Frame Mode Allowed.*

Chapter 8 contains a detailed explanation of the features available and the differences between the five modes.
(see Chapter 8, Modes)

Mode Icon Bar

The vast majority of editing features available in EnSight are divided into five different groups and are accessible through the Mode Icon Bar. The set of Icons you see at any time are determined by which Mode has been selected in the Mode Selection Area.

The various Mode Icon Bars can be customized by the user:

See Preferences... in Section 6.2, Edit Menu Functions)

Chapter 8 contains a detailed explanation of the features available and the differences between the five modes.
(see Chapter 8, Modes)

Transformation Area

This area determines how you will transform Parts within the Graphics Window and also provides quick access to the Transformations Editor for precise control of transformations. Buttons are available for quick viewing down any of the major axes, for Storing/Recalling views and for undoing the last transformation.

Chapter 9 contains a detailed description of the features in the Transformation Area.
(see Chapter 9, Transformation Control)

Message Area

This area provides feedback on what EnSight is doing. If you are using Transient data, this area will indicate which time step is currently in use.

Information Area Button

This button will bring up a dialog which will display any output that EnSight generates. When no new information is in the area, the button will be the typical interface color. When new information has been placed in the area, the button will be green in color. If warning information has been placed in the area, it will be yellow in color. If error information has been placed in the area, it will be red in color.

Quick Interaction Area	This area provides quick access to the features associated with each of the Icons in the Feature Icon Bar. Chapter 7 contains a detailed explanation of the features in the Quick Interaction Area which are available through each of the Icons. (see Chapter 7, Features)
Quick Desktop Buttons	This area contains some very commonly used toggles, such as the shading and tools.
Graphics Window	This area shows the model using the current display attributes. You perform all interactive transformations in the Graphics Window.

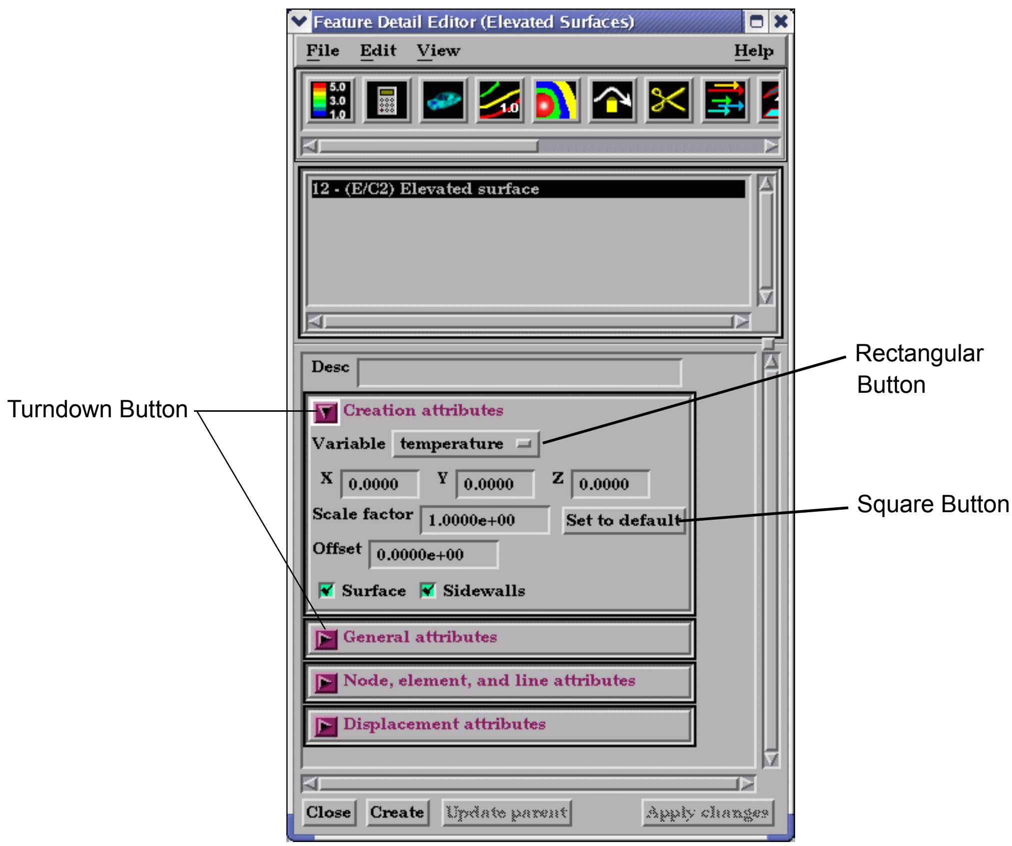
GUI Conventions

The EnSight GUI uses menus and dialogs that utilize and expand upon established OSF/Motif conventions. This section provides some general information on the operation of EnSight dialogs, menus, lists, buttons, and text fields.

Dialogs

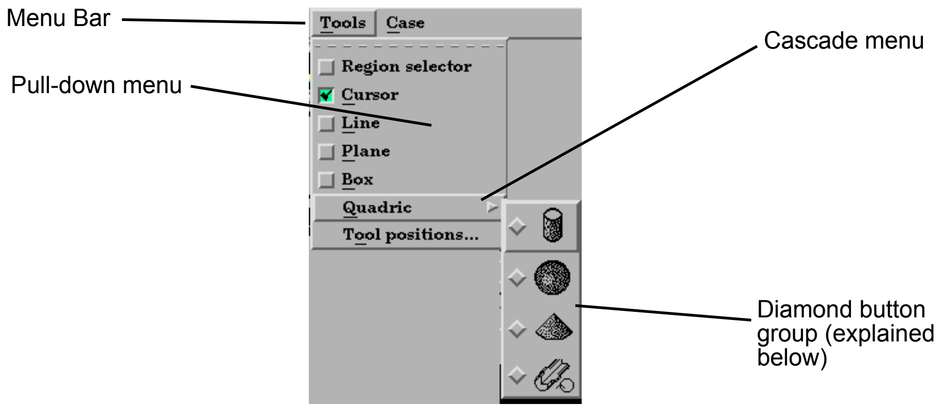
A dialog is a window that groups interface components based on function. Dialogs are typically opened by making selections from a menu. Menu selections that open dialogs always end with "...". Most EnSight dialogs can be opened and closed independently. In order to optimize scarce workstation screen real estate, you should close dialogs that are not in use.

Dialogs typically consist of buttons, menus, lists, and areas to type in. Many EnSight dialogs also have expandable sections that let you hide parts of the interface that you use infrequently. Each expandable section consists of an indicator button, a section title, and the contents of the section. The indicator button and the section title are always visible. If the section is open, the contents are visible as well.



Menus

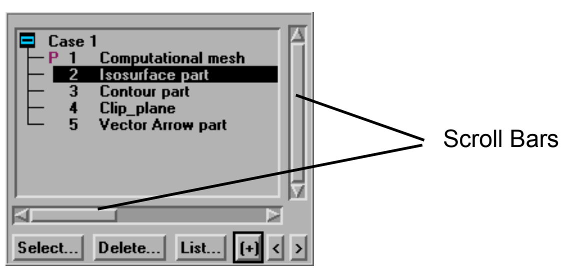
The EnSight documentation uses the following terms to describe various types of menus.



Menu Bar	A horizontal strip across the top of some dialogs listing menu titles.
Pull-down menu	A pull-down menu is one accessed directly from a menu bar.
Cascade menu or submenu	A submenu is accessed from another menu selection. Submenu selections are indicated by a right-pointing arrow.
Pop-up menu	A pop-up menu is accessed by pressing the associated rectangular button. The current selection from the menu always appears as the button title. (An example is the rectangular button labeled "PRESSURE" beside the word Variable shown above in the Feature Detail Editor.)

Lists

EnSight provides access to the list of Model and Created Parts as well as Original and Created Variables through the Main Parts List and the Main Variables List as well as the sub-lists available in the various Feature Detail Editors. These lists are presented as scrollable sections. Various mechanisms are used to select items from a list for further action:



Select (or single-click)	Place the mouse cursor over the item and click the left mouse button. The item is highlighted to reflect the "selected" state.
Select-drag	Place the mouse cursor over the first item. Click and hold the left mouse button as you drag over the remaining items to be selected. Only contiguous items may be selected in this fashion.
Shift-click	Place the mouse cursor over the item. Depress the shift key and click the left mouse button. This action will extend a selection to include all those items sequentially listed between the previous selection and this one.

Control-click Place the mouse cursor over the item. Depress the control key and click the left mouse button. This action will extend a selection by adding the new item, but not those in-between. Use this mechanism to build a non-contiguous selection.

Double-click Place the mouse cursor over the item and click the left mouse button twice in rapid succession.

Right-click Place the mouse cursor over the item and click the right mouse button. There are a number of shortcuts available using the right-click in the graphics window, and in the part list window.

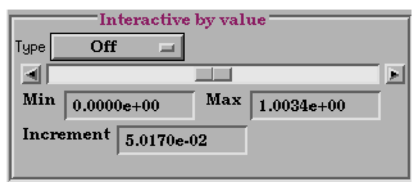
Middle-click Place the mouse cursor over the item and click the middle mouse button.

Buttons

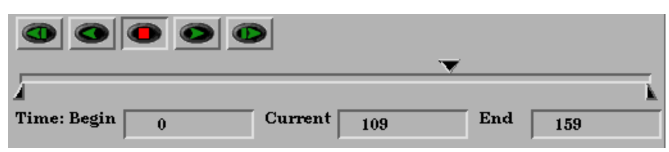
EnSight uses the following kinds of buttons:

Rectangular Place the mouse cursor in the button area and click the left mouse button. Rectangular buttons typically access the function described in the label. If the label is followed by "..." then the button opens another dialog. (Example shown above.)

Arrow Place the mouse cursor in the button area and click the left mouse button. Arrow buttons typically have an associated text field. Clicking the button increments or decrements the text field value. (Example shown below.).



Triple Slider Click on the slider above the bar and drag it to change the current step or value. Click on the left slider below the bar and slide to set the begin step or begin value. Click on the right slider below the bar and slide to set the end step or end value.
The buttons above the triple slider are traditional VCR controls.



Diamond Place the mouse cursor in the button area and click the left mouse button. Diamond buttons (also called radio buttons) are toggles that select an item from a mutually exclusive list. Exactly one diamond button of a group can be on at any given time. (Example shown above.)

Square Place the mouse cursor in the button area and click the left mouse button. Square buttons are toggles that access the function indicated by the label (Example shown above.).

Turndown — Place the mouse cursor in the button area and click the left mouse button. Turndown buttons are toggles for opening and closing a section. A down pointing arrow button indicates an open section. A right pointing arrow button indicates a closed section. (Example shown above.)

Text Fields

EnSight utilizes two types of text fields:

Information Text Fields — These text fields are used to report information and cannot be edited by the user. Information text fields are surrounded with a single pixel border.

Editable Text Fields — Place the mouse cursor in the text field and click to insert a blinking insertion cursor. Several techniques are available to accelerate text editing. Select a single word by double-clicking or the entire string by triple-clicking. Selected text is replaced by subsequent typing. The left and right arrow keys (on most systems) will move the insertion cursor.
EnSight does not recognize the change in the text field until you press Return.

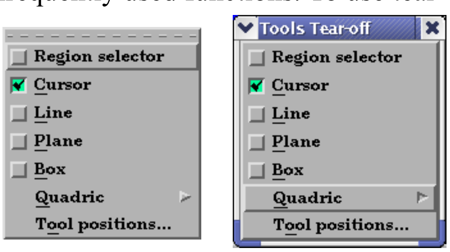
Directory conventions

Shown below are shortcut specifications for directory paths in text fields.

`~/`	Expands to your home directory
`~username/`	Expands to the home directory of `username`
`./`	Expands to the current working directory
`../`	Expands to the parent directory of the current working directory

Tear-Off Menus

If your window system allows it, the EnSight user interface supports “tear-off” menus. Judicious use of tear-off menus can provide custom, rapid access to frequently used functions. To use tear-off menu.



Select (or single-click) — Place the mouse cursor over a pulldown menu button, then click and release the left mouse button. This operation will open the pulldown menu.

Tear off — Move the mouse cursor to the dotted lines on the menu, and again click and release the left mouse button. This will “tear off” the pulldown into a separate window which can be placed anywhere on the screen.

Closing a tear-off — A tear-off menu can be closed by selecting Close from the tear-off window's frame menu which is accessed clicking on the button in the upper left of the dialog frame.

Dialog Control — The window manager will normally allow you to control some basic functions (Restore, Move, Raise, Lower, Close) by clicking-holding the right mouse button on a dialog or window border.

4 Variables

Included in this chapter:

General Description

Variables are numerical values provided by your analysis software or created within EnSight. Variables can be dependent on server part-geometry (for example, the area of a part), and a part's geometry can be dependent on its parent part's variable values (for example, an isosurface).

Variable Types

There are four types of variables: *tensor, vector*, *scalar*, and *constant*. Scalars and vectors can be real or complex. Symmetric tensors are defined by six values, while asymmetric tensors are defined by nine values. Vectors, such as displacement and velocity, have three values (the components of the vector) if real, or six values if complex. Scalars, such as temperature or pressure, have a single value if real, or two values if complex. Constants have a single value for the model, such as analysis time or volume. All four types can change over time for transient models.

Activation

Before using a variable, it must be loaded by EnSight, a process called activation. EnSight normally activates variables as they are needed. Section 4.1 describes how to select, activate, and deactivate variables to make efficient use of your system memory.
(see Section 8.1, Color Selector)

Creation

In addition to using the variables given by your analysis software, EnSight can create additional variables based on any existing variables and geometric properties of server parts. EnSight provides approximately 100 functions to make this process simpler.(see Section 8.1, Color Selector)

Color Palettes

Very often you will wish to color a part according to the values of a variable. EnSight associates colors to values using a *color palette*. You have control over the number of value-levels of the palette and the type of scale, as well as control over colors and method of color gradation. You also use function palettes to specify a set of levels for a variable, such as when creating contours.
(see Section 8.1, Color Selector)

Queries

You can make numerical queries about variables and geometric characteristics of Server-based parts. These queries can be at points, nodes, elements, parts, along lines, and along 1D parts. If you have transient data, you can query at one time step or over a range of time steps, looking at actual variable values or a Fast Fourier Transform (FFT) of the values. (see Section 8.1, Color Selector)

Plotting

Once you have queried a variable, you can plot the result.
(see Section 8.1, Color Selector)

From More than One Case

Variables can come from more than one case. If more than one case has a variable with the same name, this will be treated as one variable. If a variable is applicable to one case but not another, it will not be applied to the non-applicable case(s).

Parts

When variables are activated or created, all parts except Particle Trace parts are updated to reflect the new variable state. Particle Trace parts will always show variables which are activated after the part's creation as zero values.

The input to all of the predefined functions includes some type of server based parts. Please be aware that parts which reside only on the client (*contours, particle traces, profiles, vector arrows, and tensor glyphs*) will not be used.

Location

Variables can be defined at the vertices, at the element centers, or undefined.

User Defined Math Functions

Users can write external variable calculator functions called User Defined Math Functions (UDMF) that can be dynamically loaded by EnSight. These functions appear in EnSight's calculator in the general function list and can be used just as any other calculator function to derive new variables.

Several examples of UDMFs can be found in the directory `$CEI_HOME/ensight92/src/math_functions/`. Please see these examples if you wish to create your own UDMFs.

When the EnSight server starts it will look in the following subdirectories for UDMF dynamic shared libraries:

```
./libudmf-devel.so (.sl) (.dll)
$ENSIGHT9_UDMF/libudmf-*.so (.sl) (.dll)
$CEI_HOME/ensight92/machines/$ENSIGHT9_ARCH/lib_udmf/libudmf-*.so (.sl)
(.dll)
```

Depending on the server platform, the dynamic shared library must have the correct suffix for that platform (e.g. `.so, .sl, .dll`).

Currently, when a UDMF is used in the EnSight calculator, it is invoked for each node in the specified part(s) if all the variables operated on for the specified part(s) are node centered. If all of the variables are element centered, then the UDMF is invoked for each element in the part(s). If the variables are a mix of node and element centered values, then the node centered values are automatically converted to element centered values and then the UDMF is invoked for each element using element centered variables.

Arguments and the return type for the UDMF can be either scalar or vector EnSight variables or constants. At this time, only variable quantities and constants can be passed into UDMFs. There is no mechanism for passing in either part geometry, neighboring variables, or other information.

4.1 Variable Selection and Activation

All available variables, both those read in and those created within EnSight, are shown in the Feature Detail Editor (Variables), whether they have been activated or not. In addition, a variable list is included in each function requiring a variable. In this case, only the appropriate variable types are shown.

Feature Detail Editor Variables

Double clicking on the Color Icon in the Part Mode Icon Bar (or from the Menu, Edit > Variables Editor) opens the Feature Detail Editor (Variables).

Feature Detail Editor Variables List

Click here to Edit Palette

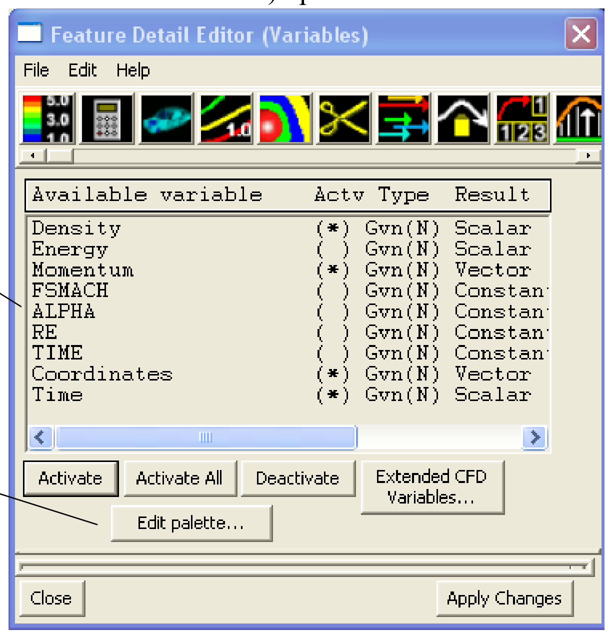


Figure 4-1
Feature Detail Editor (Variables)

Feature Detail Editor Variables List

This list shows all variables currently available, both those read from data and those you have created within EnSight. Each row provides information about a variable.

Available Variable

The description or name of the variable.

( ) or (*)

Activation status. An asterisk indicates that the variable has been activated.

Type

Type of the variable:

Gvn Scalar:	real scalars read from the dataset (Given).
Cmp Scalar:	real scalars created within EnSight (Computed).
Gvn Complex Scalar:	complex scalars read from the dataset (Given).
Cmp Complex Scalar:	complex scalars created within EnSight (Computed).
Gvn Vector:	real vectors read from the dataset (Given).
Cmp Vector:	complex vectors created within EnSight (Computed).
Gvn Complex Vector:	complex vectors read from the dataset (Given).
Cmp Complex Vector:	complex vectors created within EnSight (Computed).
Gvn Tensor:	real tensors read from the dataset (Given).
Cmp Tensor:	real tensors created within EnSight (Computed).
Gvn #:	constants read from the dataset (Given).
Cmp #:	constants created within EnSight (Computed).

Result

Current value of a constant variable (is blank for other types of variables). Changing the current solution time will update the value in this column to the value for the new time.

Caes(s) — Shows which case(s) the variable belongs to, or "All" if available for all cases.

Activate — Clicking this button activates the variable(s) selected in the Feature Detail Editor Variables List. Activation of a variable loads its values into the memory of the EnSight Server host system. The EnSight Server then passes the necessary data to the Client. One way you can control EnSight's memory usage is to only activate the variables you want to use. Once activated, a variable becomes available in the Main Variables List and, as is described in Section 4.2, EnSight creates a default color palette for the variable.

Activate All — Clicking this button activates all variables listed in the Feature Detail Editor Variables List, regardless of which are selected.

Deactivate — Clicking this button deactivates the variable(s) selected in the Feature Detail Editor Variables List. Deactivating a variable frees up some memory on both the Client and the Server. You can activate and deactivate variables as often as you like. For example, you could activate one variable to color a part, deactivate that variable, then activate a different variable to re-color the part. Of course, if you have enough memory and a small enough model, you can simply activate all the variables and leave them activated.

Extended CFD Variables... — These were intended as a suppliment to the OVERFLOW and PLOT3D readers. Opens the Extended CFD Variable Settings dialog. If your data defines variables or constants for density (SCALAR or CONSTANT), total energy per unit volume (SCALAR), and momentum (or velocity) (VECTOR), it is possible to show new variables defined by these basic variables in the Main Variables List of the GUI by utilizing the capabilities of this dialog. (See Preferences... in Section 6.2, Edit Menu Functions).
WARNING: Modifications to this dialog will not affect extended CFD variables that have already been activated - only future activated variables are affected. To modify an existing variable you will need to modify the variable's working expression in the calculator and recalculate it.

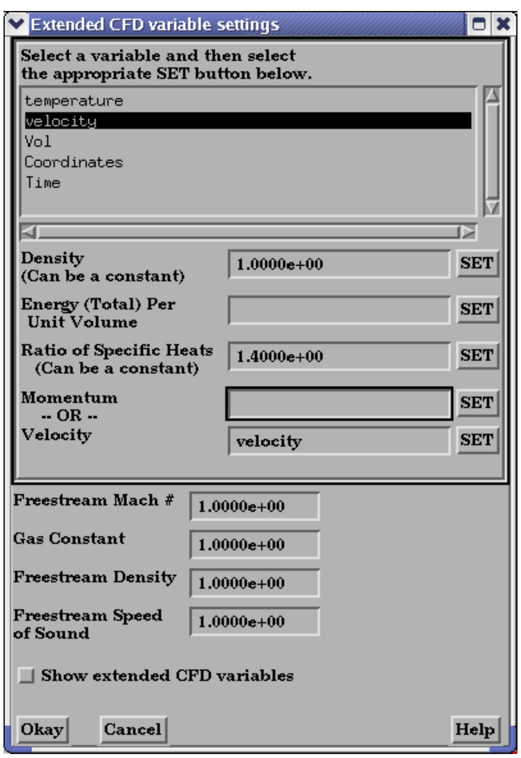


Figure 4-2
Extended CFD Variable Settings Dialog

WARNING If you deactivate a created variable or any of the variables used to define it, both the values and the definition of the created variable are deleted. If you deactivate a variable used to create a part's geometry, the part will be deleted. If you deactivate a variable who's color palette has been used to color a part, the part's appearance will change.

(see How To Activate Variables)

Menus

File Menu — Clicking this button opens a pull-down menu with the following options:

Save Selected Palette(s)... — Opens the file selection dialog for the specification of a filename in which to save the selected color palette(s).

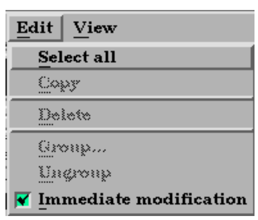
Save All Palettes… — Opens the file selection dialog for the specification of a filename in which to save all color palette(s).

Restore Palette(s)... — Opens the file selection dialog for the specification of a filename from which to restore previously saved color palettes.

Load constants)... from file... — Opens the file selection dialog for the specification of a filename which is an EnSight constant variables formated file (see section 11.16 Constant Variable File Format)

Save Selected Constant(s)... — Opens the file selection dialog for the specification of a filename in which to save the selected constant values.

Save All Constant(s)... — Opens the file selection dialog for the specification of a filename in which to save all constant values.



Edit Menu — Clicking this button opens a pulldown menu with the following choice:

Select All — Clicking this selects all variables in the Feature Detail Editor Available Variables List.

Immediate Modification Toggle — Default is On. While on, any modification made in the Editor is immediately implemented by EnSight. For large problems, this may be impractical. In such instances, click this toggle off, make all desired modifications, and then implement them all at once by clicking the Apply Changes button at the bottom of the Editor dialog.

4.2 Variable Summary & Palette

You can visualize information about a model by representing variable values with colors, often called fringes. Fringes are an extremely effective way to visualize variable variations and levels. A variable color palette associates (or maps) variable values to colors. Palettes are also used in the creation of contours. The number of contour levels is based on the number of palette color levels, and the contour values are based on the palette level values.

EnSight uses a variable's color palette to convert numbers to colors, while you, the viewer, use them in the opposite manner—to associate a visible color with a number. If you wish, EnSight can display a color-value legend in the Main View window.

Default Palettes

At least one color palette—the Coordinate color palette—always exists, even if your model has no variables. In addition, EnSight creates a color palette for each real scalar and vector variable that you activate, giving the color palette the same name as the variable. If the variable is a vector variable, the default color palette uses the vector's magnitude. Tensor variables have no palette.

Default color palettes have five color levels. Ranging from low to high, the colors are blue, cyan, green, yellow, and red (the spectral order). The numerical values mapped to these five levels are determined by first finding the value-range for the variable at the current time step when the variable is activated. The value for the lowest level is set to the minimum value. The value for the highest level is set to the maximum value. The three middle levels are spaced evenly between the lowest and highest values. For datasets with only one time step, the scheme just described works well because the variable's value range is not changing over time. However, if you have transient data, the range could vary widely at different times and since the default was based on one time step, it may not be appropriate for other time steps. EnSight can show you a histogram of the variable values over time to assist you in setting a palette for transient cases.

Value Levels

A color palette can have up to 21 levels at which the variable value is specified. Each color palette level's value must be between the value at the adjoining levels. You select whether the scale is linear (the default), quadratic (2^x), or logarithmic ($\log_{10}$).

Sometimes you may wish to only visualize areas whose palette-variable values are in a limited range. You can choose to visualize other areas with a different, uniform color, or to make those areas invisible.

Management

The Feature Detail Editor (Variables) enables you to manage your color palettes. You can copy, save to a file, and restore from a file existing palettes. You can also edit the palette. To see the Palette Editor dialog, click on the Edit Palette... button.

Clicking the Edit Palette... button opens the Palette Editor dialog.

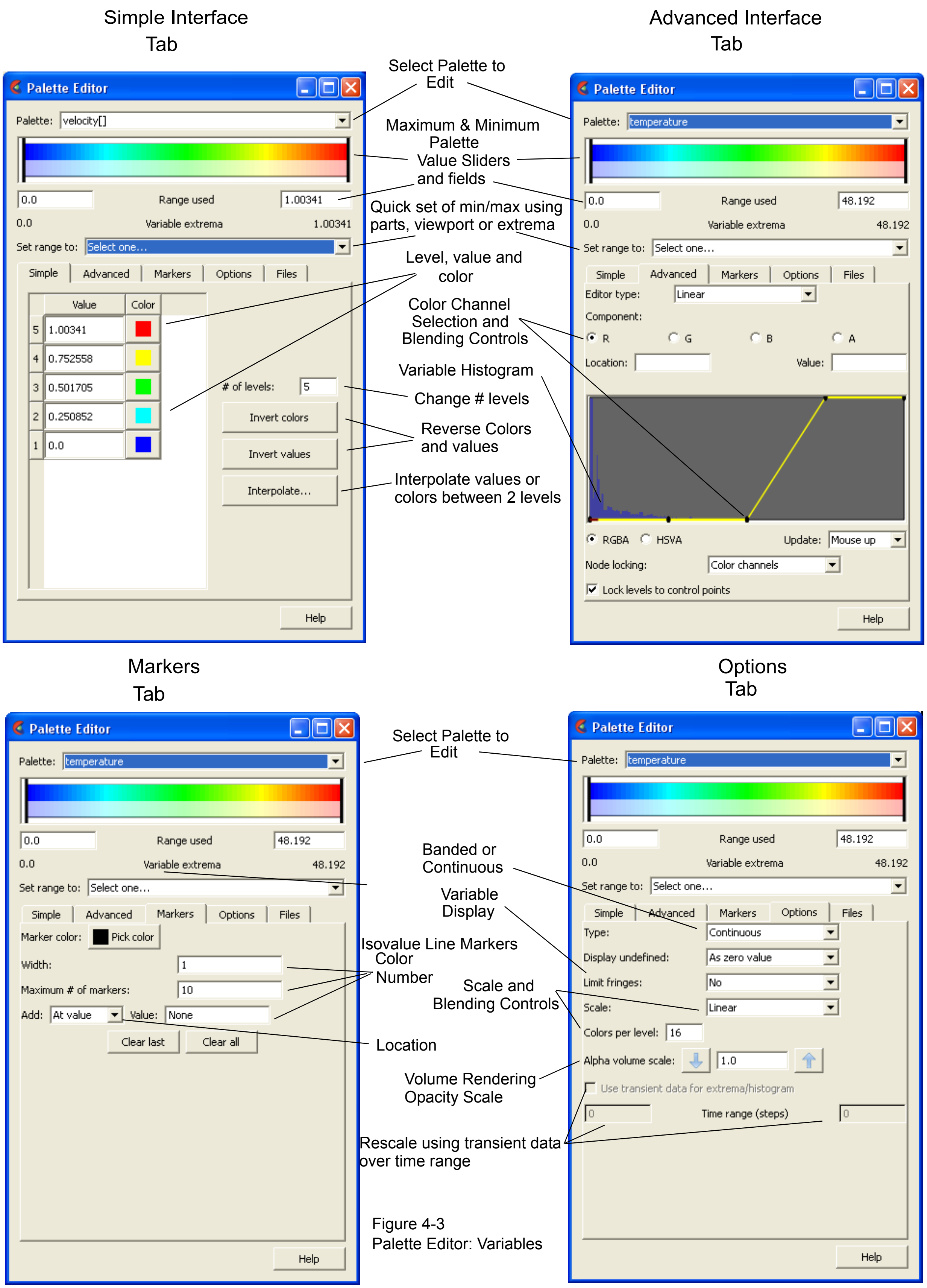


Figure 4-3
Palette Editor: Variables

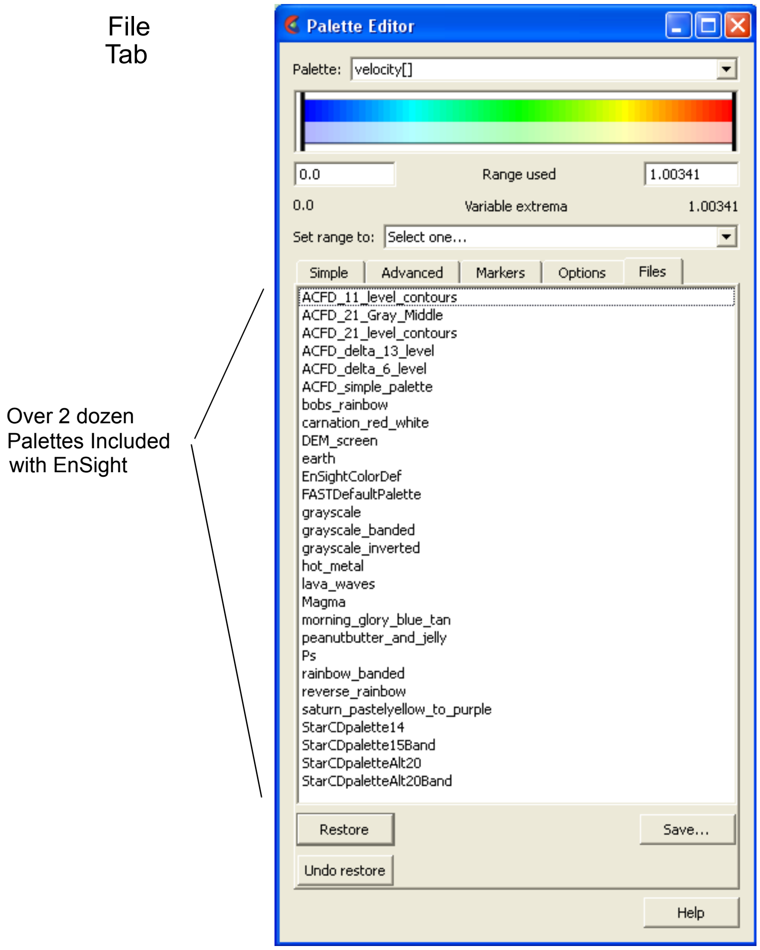


Figure 4-4
Palette Editor: Variables File Tab

Palette Editor Items Available on Every Tab

Palette	Select the variable palette to be edited
Color Palette	This horizontal color legend shows the color range for the palette selected. The left and right black vertical lines indicate the current min/max values in use in reference to the variable extrema and can be clicked/dragged to adjust the minimum and maximum values in use.
Range Used	Specifies the minimum and maximum values to be used for the bottom and top levels in the palette respectively
Set range to:	The minimum and maximum values assocated with the palette can be set to one of the following
extrema	Sets the minimum and maximum palette range values to a static value which is the variable's minimum and maximum value at the current timestep. This is value can be set using the variable's minimum and maximum value over all time under the Options Tab by toggling on "Use Transient Data for Extrema/Histogram". The min and max can be reset each timestep under the Options Tab by toggling on "Reset Palette Range on Time Change".
selected parts	Does a one-time reset of the minimum and maximum palette range values using the *viewable* elements on the screen of the selected parts (blanking and part element

representation affect this range). If you change time you will need to select this opton again to reset the min and max to the selected parts variable min and max.

current viewport Sets the minimum and maximum palette range values using the *currently visible* elements in the *current* viewport (blanking, part element representation, and transformation will all affect this range). If you change time you will need to select this opton again to reset the min and max to the current viewport variable min and max.

Palette Editor Simple Tab

Value/Color matrix You can type new values into each numeric field to adjust the value associated with a color. If you click on the color swatch for any level a color selector dialog will appear allowing you to set the color at the level indicated.

# of levels This field specifies the number of value levels for the selected palette. Each level will be defined as a value and color in the field at the left of the dialog.

Invert colors Reverses the colors for the palette

Invert values Reverses the values for the palette

Interpolate... Brings up a dialog allowing you to interpolate values or colors between two levels.

Palette Editor Advanced Tab

Variable Palette The background of the middle graphic shows the relative number of nodes at which the

Histogram Value of the selected variable is within the range represented by a particular value band. The small horizontal line at the far left of the graphic can be used to interactively adjust the vertical scale of the histrogram.

Control Points By default EnSight will create the same number of control points as there are levels in the palette. Control points can be added or deleted by right clicking on a control point marker. When adding a control point the point will be added half way between the selected point and the one immediately to the right. To decouple the number of control points from the number of palette levels see "Lock levels to control points" under the Options tab. To adjust the component value at the control point click and drag the control point.

Editor Type The control points can control color by straight line interpolation (Linear) or by creating a spline.

Component Selects which color channel will be edited via the control points - Red, Green, Blue, or Alpha, or Hue, Saturation, Value, or Alpha, depending on the state of the RGB/HSVA toggle below.

Location Indicates the location (the x value) of the selected control point in the range 0 (left side) to 1.

Value Indicates the value of the component of the selected control point in the range 0 (off) to 1 (full value)

RGB/HSVA Toggle between Red/Green/Blue or Hue/Saturation/Value to represent color.

Update Specifies when the update to the scene will occur. Delayed will cause the update to occur when you select the Apply Changes button at the bottom of the dialog. Mouse up indicates that as you modify the control points or min/max range markers in the palette editor that the update will occur when you release the mouse button. And finally, Immediate will update the scene while you modify the control points.

Node locking By default the control points for the color and alpha channels are locked together, i.e., if a node is added/deleted it is added to all channels and if the location of a control point is moved it affects the location of all channels. "Color channels" indicates that only the color is locked together and the alpha control points are independent. "None" indicates that all color as well as alpha are independent. Warning - you can not make the node locking more restrictive than the current setting, i.e., if you set the node locking to None you will not be able to set it to either of the two other choices.

Lock levels to control points — If turned on (the default), the number of control points for the palette will be the same as the number of levels and they will be uniformly spaced. Turning this off allows you to decouple the number and location of control points from the number of levels.

Palette Editor Markers Tab

It is possible to modify a texture entry in the color palette to show a particular color. These inserted colors show up as regions (contours) of constant color in the graphics window. The width of the resulting contour is a result of the marker width as well as the number of colors per level (see Options tab).

A marker can be added by directly clicking on the horizontal legend in the dialog.

Marker Color — Sets the color for all defined markers

Width — Sets the number of texels covered by the marker

Maximum # of markers — Sets the total number of marker objects that will be stored for the palette

Add:

- *At Value* — Specify the variable value in the Value field which will be inserted as a marker into the palette.
- *Uniformly* — Specify the number of uniformly spaced markers in the 'How many' field that will be added into the palette
- *At levels* — Will add a marker at each level of the palette

Clear last — Remove the last marker object created

Clear all — Remove all markers from the palette

Palette Editor Options Tab

Type — This pulldown allows for the selection of the desired type of color gradation. The options are:

- *Continuous* — Displays graduated color variation across or along each element interpolating the color across each element based on the value of the variable at the nodes. If the variable tied to the palette is defined at the element centers the result will be a constant color across the entire element (see alse "Use continous palette for per element vars" option under Edit->Preferences->Color Palettes).
- *Banded* — Displays discrete color bands across the elements the number of which is controlled by the number of levels and the number of Colors per level

Display undefined — If the variable is not defined, the element can not be colored acording to the color palette. In this case the element will be colored by the color assocated with the value of 0. or the element can be removed (invisible).

Limit fringes — This pulldown allows you to select how you wish to display elements with variable values above or below the minimum and maximum of the color palette. Options are as follows:

- *No* — color scalar values that exceed the minimum or maximum of the palette by the same color as associated with the minium or maximum of the palette. This is the default.
- *By part color* — Color scalar values that exceed the minium or maximum of the palette by the part's color.
- *By invisible* — Color scalar values that exceed the minimum or maximum of the palette using full transparency

Scale — This pulldown allows you to select the desired type of interpolation in the palette. The options are:

- *Linear* — Interpolation in the palette is linear
- *Quadratic* — Interpolation in the palette is quadratic

Logarithmic Iinterpoloation in the palette is log base 10

Colors per level Specifies the number of textels that will be used in the texture between each palette level. Typically used with Banded type palettes to set the number of colors that are viewed or with markers to create wider/thinner markers.

Alpha volume scale For volume rendered parts this factor will scale the control point alpha value by the factor indicated.

Use Transient data for extrema histogram Toggle this on and enter the begin and end time in the Time range (steps) field, to use the variable data over the indicated time range to recalculate the histogram and the palette extrema.

Reset palette range on time change Toggle this on and the palette min and max will be reset each timestep using the currrent timestep min and max values of the variable.

Palette Editor Files Tab

EnSight includes over two-dozen predefined palettes to help you display your variable variation in a way that communicates your message more quickly and effectively. Loading a Palette is done using the Restore button.You can create your own palettes and save them using the Save button.

Restore Select a palette from the list and click the Restore button to set the colors and levels of the current palette.

Save... Will save the current color and level information to a named palette for future use.

Predefined palettes can enable you to more efficiently communicate your message from your data.

The human eye is drawn to vibrant colors. There are several predefined rainbow palettes (*`EnSightDefaultPalette`*, *`ACFD_11_level_contours`*, *`ACFD_21_level_contours`*, *`bobs_rainbow`* and *`reverse_rainbow`*). There are several pastel and earthtone palettes (*`DEM_screen, earth, saturn_pastelyellow_to_purple`* and *`morning_glory_blue_tan`*).

In contrast, dull colors such as gray are uninteresting. Several of the predefined palettes are designed with a dull or uninteresting color in the middle (perhaps representing a value near zero) with the vibrant colors at the positive and negative extremes, thus highlighting the important values of your variable (*`ACFD_21_Gray_Middle, ACFD_delta_13_level,`* and *`ACFD_delta_6_level`*).

The human eye sees contrast. Alternating dark and light hues, or alternating colors with black provides a strong contrast showing the gradation of your data (`lava_waves, rainbow_banded, StarCDpalette15Band, StarCDpaletteAlt20`, and `StarCDpaletteAlt20Band`).

Finally, colorblindness is quite common. Some of the palettes below are useful for various kinds of coloblindness (`EnSightColorDef, grayscale, grayscale_banded, grayscale_inverted`).

(See How To Create Color Legends, How To Edit Color Palettes)

4.3 Variable Creation

You can create additional variables based on existing data. Typical mathematical operations, as well as many special built-in functions, enable you to produce simple or complex equations for new variables. Some built-in functions enable you to use values based on the geometric characteristics of server parts. In general, created variables are available for any process, just like given variables. If you have transient data, a time change will recompute the created variable values.

Often an analysis program produces a set of basic results from which other results can be derived. For example, if a computational fluid dynamics analysis gives you density, momentum and total energy, you can derive pressure, velocity, temperature, mach number, etc. EnSight provides many of these common functions for you, or you can enter the equation(s) and build your own.

As another example, suppose you would like to normalize a given scalar or vector variable according to its maximum value, or according to the value at a particular node. Variable creation enables you to easily accomplish such a task. The more familiar you become with this feature, the more uses you will discover.

EnSight allows variables to be defined at vertices (nodes) or element centers. If a new variable is created from a combination of nodal and element based variables, such a new variable will always be element based.

Note: Measured Variables are not supported by this functionality

Building Expressions

The Feature Detail Editor (Variables) dialog Variable Creation turn-down section provides function selection lists, calculator buttons, and feedback guidance to aid you in building the working expression (or equation) for a new variable. You can use three types of values in an expression: constants, scalars, and vectors.

Constants

A *constant* in a variable expression can be a…	for example…
• number	3.56
• constant variable from the Active Variables list	Analysis_Time
• scalar variable at a particular node/element (component and node/element number in brackets)	temperature[25]
• vector variable component at a particular node /element (component and node/element number in brackets)	velocity[Z][25]
• coordinate component at a particular node/element (component and node/element number in brackets)	coordinate[X][25]
• any of the previous three at a particular time step (time step in braces right after the variable name) *(Note: This only works for model variables, not created ones)*	temperature{15}[25] velocity{15}[Z][25] coordinate{15}[X][25]
• Math function	COS(1.5708)
• General function that produces a constant	AREA(plist)

Scalars

A *scalar* in a variable expression can be a…	for example…
• Scalar variable from the Active Variables list	pressure
• vector variable component (component in brackets)	velocity[Z]
• coordinate component (component in brackets)	coordinate[Y]
• any of the previous three at a particular time step (time step in braces right after the variable name) *(Note: This only works for model variables, not created ones)*	pressure{29} velocity{29}[Z] coordinate{29}[Y]
• General function that produces a scalar	Divergence(plist,velocity)

Vectors

A *vector* in a variable expression can be a…	for example…
• vector variable from the Active Variables list	velocity
• coordinate name from the Active Variables list	coordinate
• any of the previous two at a particular time step (time step in braces right after the variable name) *(Note: This only works for model variables, not created ones)*	velocity{9} coordinate{9}
• General function that produces a vector	Vorticity(plist,velocity)

Examples of Expressions and How To Build Them

The following are some example variable expressions, and how they can be built. These examples assume Analysis_Time, pressure, density, and velocity are all given variables.

Working Expression	*Discussion and How To Build It*
-13.5/3.5	A true constant since it does not change over time. To build it, type on the keyboard or click on the Variable Creation dialog calculator buttons `-13.5/3.5`
Analysis_Time/60.0	A simple example of modifying a given constant variable. If Analysis_Time is in seconds, this expression would give you the value in minutes. To build it, select *Analysis_Time* from the Active variable list and then type or click `/60.0`.
velocity*density	This expression is a vector * scalar, which is momentum, which is a vector. To build it, select *velocity* from the Active Variables list, type or click *, then select *density* from the Active Variable list.
SQRT(pressure[73] * 2.5)+ velocity[X][73]	This says, take the pressure at node (or element if pressure is an element center based variable) number 73, multiply it by 2.5, take the square root of the product, and then add to that the x-component of velocity at node (or element) number 73. To build it, select *SQRT* from the Math function list, select *pressure* from the Active Variables list, type `[73]*2.5)+`, select *velocity* from the Active Variable list, then type `[X][73]`
velocity^2	You have to be careful here. A vector * vector in EnSight is performed component-wise (x-component * x-component, y-component*y-component, and z-component*z-component). The magnitude of this expression is SQRT(x-component^4 + y-component^4 + z-component^4) which is NOT the square of the magnitude.. If you are looking for a scalar result, use SQRT(DOT(velocity,velocity)), or RMS(velocity) or SQRT(velocity[x]*velocity[x] + velocity[y]*velocity[y]+velocity[z]*velocity[z])
pressure{19}	This is a scalar, the value of pressure at time step 19. It does not change with time. To build it, select pressure from the Active Variables list, then type `{19}`. *(Note: variable must be a model variable, not a computed variable)*
MAX(plist,pressure)	MAX is one of the built-in General functions. This expression calculates the maximum pressure value for all the nodes of the selected parts. To build it, type or click (, select *MAX* from the General function list and follow the interactive instructions that appear in the Feedback area of this dialog (in this case, to select the parts, click Okay, and select *pressure* from the Active Variable list).
(pressure /pressure_max)^2	This scalar is essentially the normalized pressure, squared. To build it, first build the preceding MAX(plist,pressure) expression and name it “pressure_max”. Then to build this expression, select *pressure* from the Active Variables list, type or click `/`, select *pressure_max* from the Active Variables list, then type or click `)^2`.

Notice in the last example how a complex equation can be broken down into several smaller expressions. This is necessary as EnSight can compute only one variable at a time. Calculator limitations include the following:

1. The variable name cannot be used in the expression.
   The following is invalid:
   temperature = temperature + 100

   Instead use new variable:
   temperature2 = temperature + 100

2. The result of a function cannot be used in an expression.
   The following is invalid:
   norm_press_sqr = (pressure / MAX(plist,pressure))^2
   Instead use two steps:
   p_max = MAX(plist,pressure)
   then:
   norm_press_sqr = (pressure / p_max)^2

3. Neither created parts, changing geometry model parts, computed variables, nor coordinates can be used with a time calculation (using {}). If one of these is selected when you use {}, the calculation will fails with an error message.

4. Because calculations occur only on server based parts, client based parts are ignored when included in the part list of the pre-defined functions, and variable values may be undefined.

Clicking the Calculator Icon opens the Feature Detail Editor (Calculator) dialog.

Figure 4-5
Feature Detail Editor (Calculator) dialog

Variable Name — This field is used to specify the name for the variable being created. Built-in general functions will provide a default, but they can be modified here. Variable names must not start with a numeric digit and must not contain any of the following reserved characters:

(	[	{	+	@	!	*	$
)	]	}	–	space	#	^	/

Working Expression — The expression or equation for the new variable is presented in this area. Interaction with the expression takes place here, either directly by typing in values and variable names, etc., or indirectly by selecting built-in functions and clicking calculator buttons.

Clear — Clicking this button clears the Variable name field, Working Expression area, Feedback area, and deselects any built-in function.

Evaluate — Clicking this button produces the new variable defined in the working expression area. Until you click this button, nothing is really created. The selection commands specify to which parts the new variable should be applied.

Predefined functions Tab Scroll this list of built-in functions provided for your convenience. Click on a function to insert it into your Variable Name and Working Expression. For some functions, dynamic instructions and fields will appear to the right of the list.

Area

Area (any part(s))

Computes a constant variable whose value is the area of the selected parts. If a part is composed of 3D elements, the area is of the border representation of the part. The area of 1D elements is zero.

Boundary Layer

A Gradient Of Velocity Magnitude

BL_aGradOfVelMag(boundary part(s), velocity).

Computes a vector variable which is the gradient of the magnitude of the specified velocity variable on the selected boundary part(s) defined as:

$$GRAD_{BP}\,|V| = \nabla_{BP}\,|V| = \partial|V|/\partial x\, i + \partial|V|/\partial y\, j + \partial|V|/\partial z\, k$$

where:

$_{BP}$ = on boundary part
$V = V(x,y,z)$ = velocity vector
$|V|$ = magnitude of velocity vector = *SQRT(DOT(V,V))*
x, y, z = coordinate directions
i, j, k = unit vectors in coordinate directions

Note1: For each boundary part, this function finds it corresponding field part (pfield), computes the gradient of the velocity magnitude on the field part (Grad(pfield,velocity), and then maps these computed values onto the boundary part.

Note2: Node or element ids are used if they exist. Otherwise the coordinate values between the field part and boundary part are mapped and resolved via a floating-point hashing scheme.

Note3: This velocity-magnitude gradient variable can be used as an argument for the following boundary-layer functions that require this variable.

Boundary part	2D part
Velocity	vector variable

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer

Edge Skin-Friction Coefficient

BL_CfEdge(boundary part(s), velocity, density, viscosity, ymax, flow comp(0,1,or2), grad)

Computes a scalar variable which is the edge skin-friction coefficient $C_{f(e)}$ (that is, using the density ρ_e and velocity U_e values at the edge of the boundary layer – not the free-stream density ρ_∞ and velocity U_∞ values) defined as:

Component: 0 = Total tangential-flow (parallel) to wall:

$$C_{f(e)} = 2\,\tau_w / (\rho_e\, U_e^2)$$

Component: 1 = Stream-wise (flow) component tangent (parallel) to wall:

$$C_{fs(e)} = 2\,\tau_{ws} / (\rho_e\, U_e^2)$$

Component: 2 = Cross-flow component tangent (parallel) to wall:

$$C_{fc(e)} = 2\,\tau_{wc} / (\rho_e\, U_e^2)$$

where:

τ_w = fluid shear stress magnitude at the boundary = $\mu\,(\partial u/\partial n)_{n=0} = \sqrt{(\tau_{ws}^2 + \tau_{wc}^2)}$
τ_{ws} = $\mu\,(\partial u_s/\partial n)_{n=0}$ = stream-wise component of τ_w
τ_{wc} = $\mu\,(\partial u_c/\partial n)_{n=0}$ = cross-flow component of τ_w
μ = dynamic viscosity of the fluid at the wall
$(\partial u/\partial n)_{n=0}$ = magnitude of the velocity-magnitude gradient in the normal direction at the wall
$(\partial u_s/\partial n)_{n=0}$ = stream-wise component of the velocity-magnitude gradient in the normal direction at the wall
$(\partial u_c/\partial n)_{n=0}$ = cross-flow component of the velocity-magnitude gradient in the normal direction at the wall
ρ_e = density at the edge of the boundary layer
U_e = velocity at the edge of the boundary layer

boundary part	2D part
velocity	vector variable
density	scalar variable (compressible flow), constant number (incompressible flow)
viscosity	scalar variable, constant variable, or constant number
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)
flow comp	constant number 0 = tangent flow parallel to surface 1 = stream-wise component tangent (parallel) to wall 2 = cross-flow component tangent (parallel) to wall
grad	-1 = flags the computing of the velocity-magnitude gradient via 3-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradOfVelMag*

Provides a measure of the skin-friction coefficient in the tangent (parallel to surface) direction, and in its tangent's respective stream-wise and cross-flow directions, respective to the decomposed velocity parallel to the surface at the edge of the boundary layer.

This is a non-dimensionalized measure of the fluid shear stress at the surface based on the local density and velocity at the edge of the boundary layer. The following figure illustrates the derivations of the computed 'edge' related velocity values U_e, u_s, u_c &c.

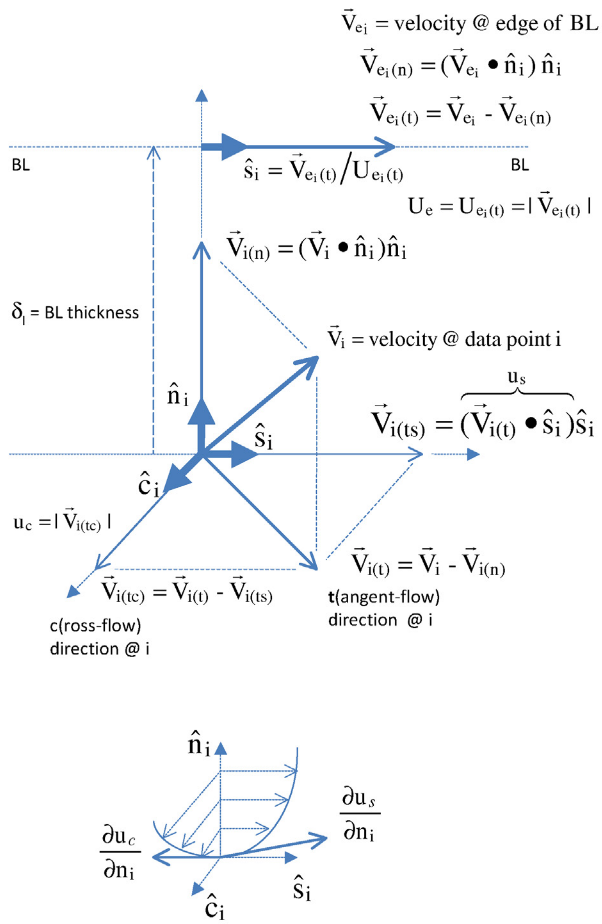


Figure 4-6
Figure Illustrating Derivation of Edge Velocity Related Values and Components

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer **Wall Skin Friction Coefficient**

BL_CfWall(boundary part(s), velocity, viscosity, free density, free velocity, grad).

Computes a scalar variable which is the skin-friction coefficient $C_{f(\infty)}$, defined as:

$$C_{f(\infty)} = \frac{\tau_w}{0.5\rho_\infty (U_\infty)^2}$$

where:

$$\tau_w = \mu_w \left(\frac{\partial u}{\partial n}\right)_{n=0} = \text{fluid shear stress at the wall}$$

μ_w = dynamic viscosity of the fluid at the wall
May be spatially and/or temporarily varying quantity (usually a constant).

n = distance profiled normal to the wall

ρ_∞ = freestream density

U_∞ = freestream velocity magnitude

$\left(\frac{\partial u}{\partial n}\right)_{n=0}$ = tangent (parallel to surface) component of the velocity-magnitude gradient in the normal direction under the "where:" list.

This is a non-dimensionalized measure of the fluid shear stress at the surface. An important aspect of the Skin Friction Coefficient is:

$C_{f(\infty)} = 0$, indicates boundary layer separation.

boundary part	2D part
velocity	vector variable
viscosity	scalar variable, constant variable, or constant number
free density	constant number
free velocity	constant number
grad	-1 = flags the computing of the velocity-magnitude gradient via 3-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradOfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer

Wall Skin-Friction Coefficient Components

BL_CfWallCmp(boundary part(s), velocity, viscosity, free-stream density, free-stream velocity-mag., ymax, flow comp(1or2), grad).

Computes a scalar variable which is a component of the skin-friction coefficient C_f tangent (or parallel) to the wall, either in the stream-wise $C_{fs(\bullet)}$ or in the cross-flow $C_{fc(\bullet)}$ direction defined as:

Component 1 = Steam-wise (flow) component tangent (parallel) to wall:

$C_{fs(\infty)} = 2\ \tau_{ws} / (\rho_\infty\ U^2_\infty)$

Component 2 = Cross-flow component tangent (parallel) to wall:

$C_{fc(\infty)} = 2\ \tau_{wc} / (\rho_\infty\ U^2_\infty)$

where:

τ_{ws} = $\mu\ (\partial u_s/\partial n)_{n=0}$ = stream-wise component of τ_w

τ_{wc} = $\mu\ (\partial u_c/\partial n)_{n=0}$ = cross-flow component of τ_w

τ_w = fluid shear stress magnitude at the wall = $\mu\ (\partial u/\partial n)_{n=0} = \sqrt{(\tau_{ws}{}^2 + \tau_{wc}{}^2)}$

μ = dynamic viscosity of the fluid at the wall

$(\partial u_s/\partial n)_{n=0}$ = stream-wise component of the velocity-magnitude gradient in the normal direction at the wall

$(\partial u_c/\partial n)_{n=0}$ = cross-flow component of the velocity-magnitude gradient in the normal direction at the wall

ρ_∞ = density at the edge of the boundary layer

U_∞ = velocity at the edge of the boundary layer

boundary part	2D part
velocity	vector variable
density	scalar variable (compressible flow), constant number (incompressible flow)
viscosity	scalar variable, constant variable, or constant number
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)

flow comp	constant number 1 = stream-wise component tangent (parallel) to wall 2 = cross-flow component tangent (parallel) to wall
grad	-1 = flags the computing of the velocity-magnitude gradient via 3-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradOfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer

Wall Fluid Shear-Stress

BL_CfWallTau(boundary part(s), velocity, viscosity, ymax, flow comp(0,1,or 2), grad).

Computes a scalar variable which is the fluid's shear-stress at the wall τ_w or in its stream-wise τ_{ws} , or cross-flow τ_{cs} component direction defined as:

Component 0 = Total fluid shear-stress magnitude at the wall:

$$\tau_w == \mu (\partial u/\partial n)_{n=0} = \sqrt{(\tau_{ws}^2 + \tau_{wc}^2)}$$

Component 1 = Steam-wise component of the fluid shear-stress at the wall:

$$\tau_{ws} = \mu (\partial u_s/\partial n)_{n=0}$$

Component 2 = Cross-flow component of the fluid shear-stress at the wall:

$$\tau_{wc} = \mu (\partial u_c/\partial n)_{n=0}$$

where:

μ = dynamic viscosity of the fluid at the wall

$(\partial u/\partial n)_{n=0}$ = magnitude of the velocity-magnitude gradient in the normal direction at the wall

$(\partial u_s/\partial n)_{n=0}$ = stream-wise component of the velocity-magnitude gradient in the normal direction at the wall

$(\partial u_c/\partial n)_{n=0}$ = cross-flow component of the velocity-magnitude gradient in the normal direction at the wall

boundary part	2D part
velocity	vector variable
viscosity	scalar variable, constant variable, or constant number
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)
flow comp	constant number 0 = RMS of the stream-wise and cross-flow components 1 = stream-wise component at the wall 2 = cross-flow component at the wall
grad	-1 = flags the computing of the velocity-magnitude gradient via 3-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradOfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer

Displacement Thickness

BL_DispThick(boundary part(s), velocity, density, ymax, flow comp(0,1,or 2), grad).

Computes a scalar variable which is the boundary-layer displacement thickness δ^*, δ^*_s, or δ^*_c defined as:

Component: 0 = Total tangential-flow parallel to the wall

$$\delta^*_{tot} = \int_0^{\delta} \left(1 - \frac{\rho u}{\rho_e U_e}\right) dn$$

Component: 1 = Stream-wise flow component tangent (parallel) to the wall

$$\delta*_s = \int_0^{\delta}\left(1 - \frac{\rho u_s}{\rho_e U_e}\right)dn$$

Component: 2 = Cross-flow component tangent (parallel) to the wall

$$\delta*_c = \int_0^{\delta}\left(\frac{\rho u_c}{\rho_e U_e}\right)dn$$

where:

n	=	distance profiled normal to the wall
δ	=	boundary-layer thickness (distance to edge of boundary layer)
ρ	=	density at given profile location
ρ_e	=	density at the edge of the boundary layer
u	=	magnitude of the velocity component parallel to the wall at a given profile location in the boundary layer
u_s	=	stream-wise component of the velocity magnitude parallel to the wall at a given profile location in the boundary layer
u_c	=	cross-flow component of the velocity magnitude parallel to the wall at a given profile location in the boundary layer
U_e	=	u at the edge of the boundary layer
y_{max}	=	distance from wall to freestream
comp	=	flow direction option
grad	=	flag for gradient of velocity magnitude

Provides a measure for the effect of the boundary layer on the “outside” flow. The boundary layer causes a displacement of the streamlines around the body.

boundary part	2D part
velocity	vector variable
density	scalar variable (compressible flow), constant number (incompressible flow)
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)
flow comp	constant number: 0 = total tangential flow direction parallel to wall 1 = stream-wise flow component direction parallel to wall 2 = cross-flow component direction parallel to wall
grad	-1 = flags the computing of the velocity-magnitude gradient via 4-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradOfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer **Distance to Value from Wall**

BL_DistToValue(boundary part(s), scalar, scalar value).
Computes a scalar variable which is the distance d from the wall to the specified value defined as

$$d = n|_{f(\alpha) = c}$$

where:

n = distance profile d normal to boundary surface

$f(\alpha)$ = scalar field (variable)

α = scalar field values

c = scalar value at which to assign d

boundary part	0D, 1D, or 2D part
scalar	scalar variable
scalar value	constant number

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer **Momentum Thickness**

BL_MomeThick(boundary part(s), velocity, density, ymax, flow comp$_i$(0,1,or2), flow comp$_j$(0,1,or2), grad).

Computes a scalar variable which is the boundary-layer momentum thickness θ_{tot}, θ_{ss}, θ_{sc}, θ_{cs}, or θ_{cc} defined as:

Components: (0,0) = Total tangential-flow parallel to the wall

$$\theta_{tot} = \frac{1}{\rho_e U^2_e} \int_0^{\delta} (U_e - u)\rho u dn$$

Components: (1,1) = stream-wise, stream-wise component

$$\theta_{ss} = \frac{1}{\rho_e U^2_e} \int_0^{\delta} (U_e - u_s)\rho u_s dn$$

Components: (1,2) = Stream-wise, cross-flow component

$$\theta_{sc} = \frac{1}{\rho_e U^2_e} \int_0^{\delta} (U_e - u_s)\rho u_c dn$$

Components: (2,1) = cross-flow, stream-wise component

$$\theta_{cs} = \frac{-1}{\rho_e U^2_e} \int_0^{\delta} \rho u_c u_s dn$$

Components: (2,2) = cross-flow, cross-flow component

$$\theta_{cc} = \frac{-1}{\rho_e U^2_e} \int_0^{\delta} \rho u_c^2 dn$$

where:

n = distance profiled normal to the wall

δ = boundary-layer thickness (or distance to edge of boundary layer)

ρ = density at given profile location

ρ_e = density at the edge of the boundary layer

u = magnitude of the velocity component parallel to the wall at a given profile location in the boundary layer

u_s = stream-wise component of the velocity magnitude parallel to the wall at a given profile location in the boundary layer

u_c = cross-flow component of the velocity magnitude parallel to the wall at a given profile location in the boundary layer

U_e = u at the edge of the boundary layer

y_{max} = distance from wall to freestream

$comp_i$ = first flow direction option

$comp_j$ = secon flow direction option

grad = flag for gradient of velocity magnitude

Relates to the momentum loss in the boundary layer.

boundary part	2D part
velocity	vector variable
density	scalar variable (compressible flow), constant number (incompressible flow)
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)
$comp_i$	constant number 0 = total tangential flow diection parallel to wall 1 = stream-wise flow component direction parallel to wall 2 = cross-flow component direcion parallel to wall
$comp_j$	constant number 0 = total tangential flow diection parallel to wall 1 = stream-wise flow component direction parallel to wall 2 = cross-flow component direcion parallel to wall
grad	-1 = flags the computing of the velocity-magnitude gradient via 4-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer **Scalar**

BL_Scalar(boundary part(s), velocity, scalar, ymax, grad).

Computes a scalar variable which is the scalar value of the corresponding scalar field at the edge of the boundary layer. The function extracts the scalar value while computing the boundary-layer thickness (see Boundary Layer **Thickness**).

where:

y_{max} = distance from wall to freestream

grad = flag for gradient of velocity magnitude

boundary part	2D part
velocity	vector variable
scalar	scalar variable

ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)
grad	-1 = flags the computing of the velocity-magnitude gradient via 4-point interpolation. vector variable = Grad(velocity magnitude)

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer Recovery Thickness

BL_RecoveryThick(boundary part(s), velocity, total pressure, ymax, grad).

Computes a scalar variable which is the boundary-layer recovery thickness δ_{rec} defined as:

$$\delta_{rec} = \int_0^{\delta} \left(1 - \frac{p_t}{p_{te}}\right) dn$$

where::

n	=	distance profiled normal to the wall
δ	=	boundary-layer thickness (distance to edge of boundary layer)
p_t	=	total pressure at given profile location
p_{te}	=	p_t at the edge of the boundary layer
y_{max}	=	distance from wall to freestream
grad	=	flag for gradient of velocity magnitude option

This quantity does not appear in any physical conservation equations, but is sometimes used in the evaluation of inlet flows.

boundary part	2D part
velocity	vector variable
total pressure	scalar variable
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)
grad	-1 = flags the computing of the velocity-magnitude gradient via 4-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer Shape Parameter

BL_Shape is not explicitly listed in the general function list, but can be computed as a scalar variable via the calculator by dividing a displacement thickness by a momentum thickness, i.e.

$$H = \delta^*/\theta$$

where:

δ^*	=	boundary-layer displacement thickness
θ	=	boundary-layer momentum thickness

Used to characterize boundary-layer flows, especially to indicate potential for separation.

This parameter increases as a separation point is approached, and varies rapidly near a separation point.

Note: Separation has not been observed for H < 1.8, and definitely has been observed for H = 2.6; therefore, separation is considered in some analytical methods to occur in turbulent boundary layers for H = 2.0.

In a Blasius Laminar layer (i.e. flat plate boundary layer growth with zero pressure gradient), H = 2.605. Turbulent boundary layer, H ~= 1.4 to 1.5, with extreme variations ~= 1.2 to 2.5.

Boundary Layer **Thickness**

BL_Thick(boundary part(s), velocity, ymax, grad).

Computes a scalar variable which is the boundary-layer thickness δ defined as:

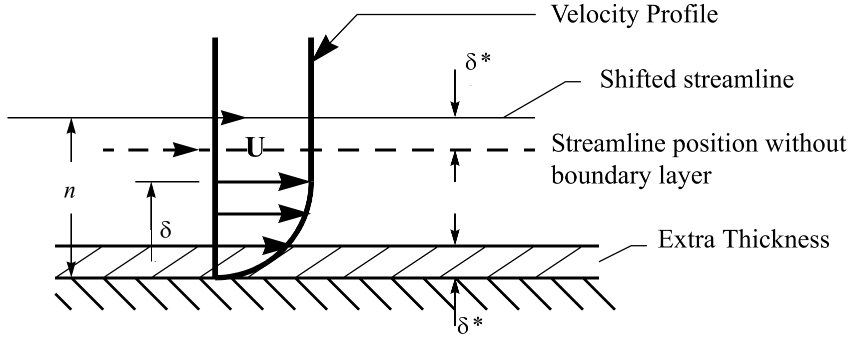
$$\delta = n|_{u/U = 0.995}$$

The distance normal from the surface to where $u/U = 0.995$,

where:

u = magnitude of the velocity component parallel to the wall at a given location in the boundary layer

U = magnitude of the velocity just outside the boundary layer



boundary part	2D part
velocity	vector variable
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note below)
grad	-1 = flags the computing of the velocity-magnitude gradient via 3-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Algorithm Note: The ymax argument allows the edge of the boundary layer to be approximated by two different algorithms, i.e. the Baldwin-Lomax-Spalart and convergence algorithms. Both schemes profile velocity data normal to the boundary surface, or wall. Specifying ymax > 0 leverages results from both the Baldwin-Lomax and vorticity functions over the entire profile to produce a fading function that approximates the edge of the boundary layer. Whereas, specifying ymax = 0 uses velocity and velocity gradient differences to converge to the edge of the boundary layer.

Please see the following references for more detailed explanations.

P.M. Gerhart, R.J. Gross, & J.I. Hochstein, Fundamentals of Fluid Mechanics, 2nd Ed.,(Addison-Wesley: New York, 1992)

P. Spalart, A Reasonable Method to Compute Boundary-Layer Parameters from Navier-Stokes Results, (Unpublished: Boeing, 1992)

H. Schlichting & K. Gersten, Boundary Layer Theory, 8th Ed., (Springer-Verlag: Berlin, 2003)

Boundary Layer

Velocity At Edge

BL_VelocityAtEdge(boundary part(s), velocity, ymax,comp(0,1,2),grad).

Extracts a vector variable which is a velocity vector V_e, V_p, or V_n defined as:

$V_e = V_e(x,y,z)$ = velocity vector at the edge of the boundary layer δ

$V_n = \mathrm{Dot}(V_e,N)$ = the decomposed velocity vector normal to the wall at the edge of the boundary layer δ

$V_p = V_e - V_n$ = the decomposed velocity vector parallel to the wall at the edge of the boundary layer δComputes a scalar variable which is the boundary-layer thickness δ defined as:

boundary part	2D part
velocity	vector variable
density	scalar variable (compressible flow), constant number (incompressible flow)
ymax	constant number > 0 = Baldwin-Lomax-Spalart algorithm 0 = convergence algorithm (See Algorithm Note under Boundary Layer Thickness)
comp	constant number 0 = velocity vector at edge of boundary layer 1 = decomposed velocity vector parallel to wall tangent to surface 2 = decomposed velocity vector normal to wall
grad	-1 = flags the computing of the velocity-magnitude gradient via 4-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Boundary Layer

y_1^+ off Wall

BL_Y1Plus(boundary part(s), velocity, density, viscosity, free density, free velocity, grad).

Computes a scalar variable which is the coefficient off the wall by one element height y_1^+ defined as

$$y_1^+ = \frac{y_1 \rho_w}{\mu_w}\sqrt{\frac{\tau_w}{\rho_w}}$$

where:

n = distance profiled normal to the wall

$\tau_w = \mu_w\left(\frac{d\mu}{dn}\right)_{n=0}$ = fluid shear stress at wall

μ_w = dynamic viscosity of fluid at wall
May be spatially and/or temporally varying quantity (usually a constant)

ρ_∞ = freestream density

U_∞ = freestream velocity magnitude

ρ_w = density at the wall

y_1 = first element height profiled normal to wall

Normally y^+ is used to estimate or confirm the required 1st grid spacing for proper capturing of viscous-layer properties. The values are dependent on various factors including, what variables at the wall are sought, the turbulent models used, and whether the law of the wall is used or not. Consult a boundary-layer text for correct interpolation of the values for your application.

boundary part	2D part
velocity	vector variable
density	scalar variable
viscosity	scalar variable, constant variable, or constant number
free density	constant number (not needed, just enter 1 for this argument)
free velocity	constant number (not needed, just enter 1 for this argument)
grad	-1 = flags the computing of the velocity-magnitude gradient via 4-point interpolation. vector variable = Grad(velocity magnitude), i.e. see *BL_aGradfVelMag*

Note: See Section 7.18, Boundary Layer Variables Create/Update

Not currently implemented for Nsided boundary part.

Case Map

CaseMap (2D or 3D part(s), case to map from, scalar/vector/tensor, 0/1 0=search only 1=if search fails find closest)

Finds the first defined variable value (scalar, vector, or tensor) using 3D parts in the 'case to map from' and maps them onto the specified part(s) If no 3D parts in the 'case to map from' then will attempt to use 2D parts. .

case to map from	constant number
scalar/vector/tensor	scalar, vector, or tensor variable
0 or 1 flag	If mapping search is successful, always assigns the exact value found. If search mapping is not successful, then the following occurs: If flag is set to 0, an undefined value will be assigned. If flag is set to 1, the defined variable value at the closest node will be assigned (so no undefined values). This option will search 3D, then 2D, then 1D and even 0D elements to find the first defined variable value.

Note: This function uses EnSight's search capability to do the mapping. It is critical that the nodes of the parts being mapped onto, lie within the geometry of all of the parts of the case being mapped from. Thus mapping onto a volume or plane in one case, that is enclosed by 3D elements in the other case, will work nicely. Mapping from a 2D surface to a 2D surface will only work reliably if the surfaces are the same (or very very close, and the flag=1 option is chosen). If the variable in the case to map from is located at the nodes, then the casemapped variable will be at the nodes, and if the variable is located at the elements, then the casemapped variable will be at the elements.

Case Map Image

CaseMapImage (2D or 3D part(s), part to map from, scalar, viewport number, Undefined value limit)

This Function does a projection of a 2D part variable from a different case onto a 3D geometry taking into account the view orientation from the specified viewport number, similar to a texture mapping.The function in effect maps 2D results to a 3d geometry taking into account view orientation and surface visibility.

part to map from	Part number of the 2D part. This 2D part is usually data from an infrared camera.
scalar	scalar variable
viewport number	The viewport number showing part(s) the variable is being computed on, from the same camera view as part to map from
Undefined value limit	Values on the 2D part that are under this value are considered Undefined

Note: If the variable in the part to map from is located at the nodes, then the casemapped variable will be at the nodes. If the variable is located at the elements the casemapped variable will be at the elements. This function takes only a scalar variable.

Coefficient

Coeff (any 1D or 2D part(s), scalar, component)

Computes a constant variable whose value is a coefficient Cx , Cy, or Cz such that

$$C_x = \int_S f n_x dS \qquad C_y = \int_S f n_y dS \qquad C_z = \int_S f n_z dS$$

where:

f = any scalar variable
S = 1D or 2D domain
n_x = x component of normal
n_y = y component of normal
n_z = z component of normal

component	[X], [Y], or [Z]

Specify [X], [Y], or [Z] to get the corresponding coefficient.

Note: Normal for a 1D part will be parallel to the plane of the plane tool.

Complex

Cmplx(any part(s), scalar/vector(real portion), scalar/vector(complex portion), [optional frequency(Degrees)])

Creates a complex scalar or vector from two scalar or vector variables. The frequency is optional and is used only for reference.

Z = A + Bi

real portion	scalar or vector variable
complex portion	scalar or vector variable (but must be same as real portion)
[frequency]	constant number (optional)

Complex Argument

CmplxArg (any part(s), complex scalar or vector)

Computes the Argument of a complex scalar or vector. The resulting scalar is given in degrees and will be in the range -180 and 180 degrees.

Arg = atan(Vi/Vr)

Complex Conjugate

CmplxConj (any part(s), complex scalar or vector)

Computes the Conjugate of a complex scalar of vector. Returns a complex scalar or vector where:

Nr = Vr
Ni = -Vi

Complex Imaginary

CmplxImag (any part(s), complex scalar or vector)

Extracts imaginary portion of a complex scalar or vector into a real scalar or vector.

N = Vi

Complex Modulus

CmplxModu (any part(s), complex scalar or vector)

Returns a real scalar/vector which is the modulus of the given scalar/vector

N = SQRT(Vr*Vr + Vi*Vi)

Complex Real

CmplxReal(any part(s), complex scalar or vector)

Extracts the real portion of a complex scalar or vector into a real scalar or vector.

N = Vr

Complex Transient Response

CmplxTransResp(any part(s), complex scalar or vector, constant PHI(0.0-360.0 Degrees))

Returns a real scalar or vector which is the real transient response:

Re(Vt) = Re(Vc)Cos(phi) - Im(Vc)Sin(phi)

which is a function of the transient phase angle "phi" defined by:
phi = 2 Pi f t
where
t = the harmonic response time parameter
f = frequency of the complex variable "Vc"

and the complex field "Vc", defined as:
Vc = Vc(x,y,z) = Re(Vc) + i Im(Vc)
where
Vc = the complex variable field
Re(Vc) = the Real portion of Vc
Im(Vc) = the imaginary portion of Vc
i = Sqrt(-1)

Note, the transient complex function, was a composition of Vc and Euler's relation, namely:

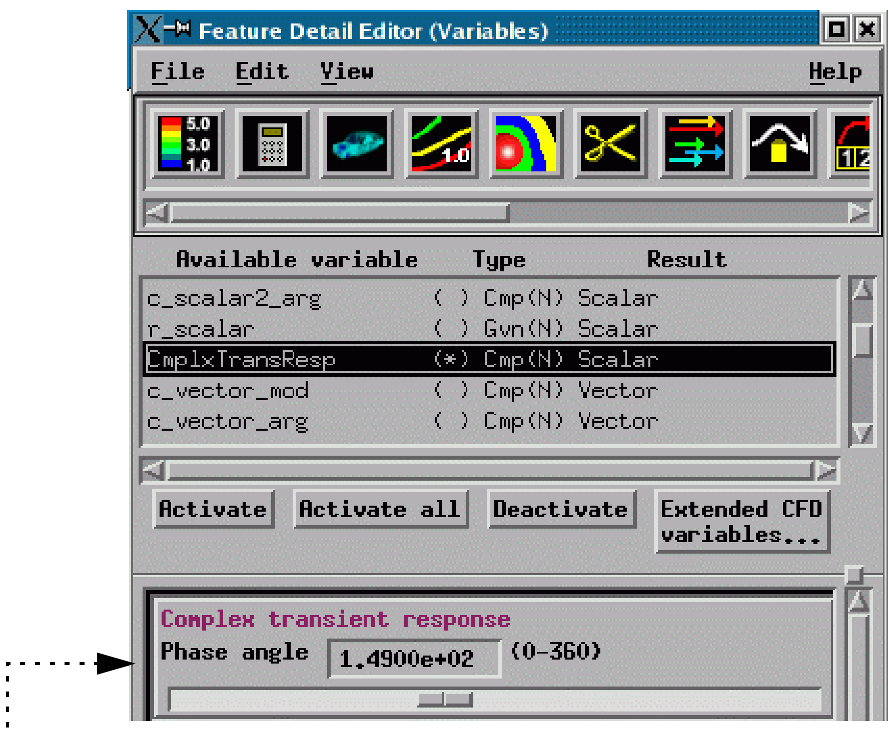
Vt = Vt(x,y,z,t) = Re(Vt) + i Im(Vt) = Vc * e^(i phi)

where:
e^(i phi) = Cos(phi) + i Sin(phi)

The real portion Re(Vt), is as designated above:

Note: this function is only good for harmonic variations, thus fields with a defined frequency!

phi angle	constant number between 0 and 360 degrees.



Note: A special area becomes available in the Feature Detail Editor (Variables) and Feature Detail Editor (Calculator) when you highlight a variable of this type - allowing you to modify the phase angle (phi) easily with a slider.

Curl

Curl (any part(s), vector)

Computes a vector variable which is the curl of the input vector

$$Curl_f = \bar{\nabla} \times \bar{f} = \left(\frac{\partial f_3}{\partial y} - \frac{\partial f_2}{\partial z}\right)\hat{i} + \left(\frac{\partial f_1}{\partial z} - \frac{\partial f_3}{\partial x}\right)\hat{j} + \left(\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}\right)\hat{k}$$

Density

Density(any part(s), pressure, temperature, gas constant).

Computes a scalar variable which is the density ρ, defined as:

$$\rho = \frac{p}{TR}$$

where: p = pressure
T = temperature
R = gas constant

pressure	scalar variable
temperature	scalar variable
gas constant	scalar variable, constant variable, or constant number

Log of Normalized **Density**

DensityLogNorm (any part(s), density, freestream density)

Computes a scalar variable which is the natural log of Normalized Density defined as:

$$ln\rho_n = ln(\rho / \rho_i)$$

where: ρ = density

ρ_i = freestream density

density	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number

Normalized **Density**

DensityNorm (any part(s), density, freestream density)

Computes a scalar variable which is the Normalized Density ρ_n defined as:

$$\rho_n = \rho / \rho_i$$

where: ρ = density

ρ_i = freestream density

density	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number

Normalized Stagnation **Density**

DensityNormStag (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, freestream velocity magnitude)

Computes a scalar variable which is the Normalized Stagnation Density ρ_{on} defined as:

$$\rho_{on} = \rho_o / \rho_{oi}$$

where: ρ_o = stagnation density

where: ρ_{oi} = freestream stagnation density

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
freestream velocity magnitude	constant variable or constant number

Stagnation **Density**

DensityStag (any part(s), density, total energy, velocity, ratio of specific heats)

Computes a scalar variable which is the Stagnation Density ρ_o defined as:

$$\rho_o = \rho\left(1 + \left(\frac{\gamma - 1}{2}\right)M^2\right)^{(1/(\gamma - 1))}$$

where: ρ = density

γ = ratio of specific heats

M = mach number

total energy must be a scalar

velocity must be a vector

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Distance Between Nodes

Dist2Nodes(any part(s), nodeID1,nodeID2).

Computes a constant, positive variable that is the distance between any two nodes. Searches down the part list until it finds nodeID1, then searches until it finds nodeID2 and returns Undef if nodeID1 or nodeID2 cannot be found. Nodes are designated by their node id's, so the part must have node ids. (Note that most created parts do not have node ids.)

Note also for transient results, that the geometry type is important for using this function. There are three geometry types: static, changing coordinate, and changing connectivity. You can find out your geometry type by doing a Query>Dataset and look in the General Geometric section of the pop up window. If you have a static geometry with visual displacement turned on then dis2nodes will not use the displacement in its calculations. You will need to turn on server-side (computational) displacement (see How To use Server Side Displacements). If you have changing coordinate geometry, then dist2node should work fine, and if you have changing connectivity then dist2node should not be used as it may give nonsensical results because connectivity is redone each timestep and node ids may move around.

And finally, to find the distance between two nodes on different parts, or between two nodes if one or both don't have ids, or the ids are not unique for the model (namely, more than one part has the same node id) use the line tool. See the Advanced Usage section of How To Use the Line Tool.

nodeID1	constant number
nodeID2	constant number

Distance to parts Node to nodes

Dist2Part(origin part + field part(s), origin part, origin part normal).

Computes a scalar variable on the origin part and field parts that is the minimum distance at each node of the origin and field parts to any node in the origin part. This distance is unsigned by default. The origin part is the origin of a Euclidean distance field. So, by definition the scalar variable will always be zero at the origin part because the distance to the origin part will always be zero.

If a nodal vector variable defined at the origin part is supplied then the normal vector is used to return a signed distance function (with positive being the direction of the normal). The signed distance is determined using the dot product of the vector from the given field node and it's closest node on the origin with the origin node's normal vector.

Notes: The origin part must be included in the field part list (although, as discussed earlier, the scalar variable will be zero for all nodes on the origin part). This algorithm has an execution time on the order of the number of nodes in the field parts times the number of nodes in the origin part. While the implementation is both SOS aware and threaded, the run time is dominated by the number of nodes in the computation.

This function is computed between the nodes of the origin and field parts. As a result, the accuracy of it's approximation to the distance field is limited to the density of nodes (effectively the size of the elements) in the origin part. If a more accurate approximation is required, use the Dist2PartElem() function. It is slower, but is less dependent on the nodal distribution in the origin part because it uses the nodes plus the element faces to calculate the minimum distance.

Usage. A typical usage would be to use an arbitrary 2D part to create a clip in a 3D field. Use the 2D part as your origin part, and select the orgin part as well as your 3D field parts. No need to have normal vectors. Create your scalar variable, called say distTo2Dpart, then create an isosurface=0 in your field using the distTo2Dpart as your variable. See also the EnSight Tips and Tutorials on our website.

origin part	part number to compute the distance to
origin part normal	a constant for unsigned computation or a nodal vector variable defined on the origin part for a signed computation

Distance to parts Node to elements

Dist2PartElem(origin part + field part(s), origin part, origin part normal).
Computes a scalar variable that is the minimum distance at each node of the origin part and field parts and the closest point on any element in origin part. This distance is unsigned by default. If a nodal vector variable is supplied on the origin part, the direction of the normal is used to return a signed distance function with distances in the direction of the normal being positive. Once the closest point in the origin part has been found for a node in an field part, the dot product of the origin node normal and a vector between the two nodes is used to select the sign of the result.
Notes: The origin part must be included in the field part list (although the output will be zero for all nodes of the origin part because it is the origin of the Euclidean distance). This algorithm has an execution time on the order of the number of nodes in the field parts times the number of elements in the origin part. While the implementation is both SOS aware and threaded, the run time is dominated by the number of nodes in the computation. This function is a more accurate estimation of the distance field than Dist2Part() because it allows for distances between nodes and element surfaces on the origin part. This improved accuracy results in increased computational complexity and as a result this function can be several times slower than Dist2Part(). See also the EnSight Tips and Tutorials on our website.

origin part	part number to compute the distance to
origin part normal	a constant for unsigned computation or a nodal vector variable defined on the origin part for a signed computation

Divergence

Div (2D or 3D part(s), vector)
Computes a scalar variable whose value is the divergence defined as:

$$Div = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$$

where u,v,w = velocity components in x,y,z directions.

Element Size

EleSize (any part(s)).
Calculates the Volume/Area/Length for 3D/2D/1D elements respectively, at each element creating a scalar, element-based variable.

Element to Node

ElemToNode (any part(s), element-based scalar or vector).
Averages an element based variable to produce a node-based variable.

Energy:

Total **Energy**

EnergyT (any part(s), density, pressure, velocity, ratio of specific heats).
Computes a scalar variable of total energy per unit volume

$$e = \rho\left(e_i + \frac{V^2}{2}\right) \quad \text{Total Energy}$$

$$e_i = e_0 - \frac{V^2}{2} \quad \text{Internal Energy}$$

$$e_0 = \frac{e}{\rho} \quad \text{Stagnation Energy}$$

where: $\rho = density$
$V = Velocity$

Or based on gamma, pressure and velocity:

$$e = \frac{p}{(\gamma - 1)} + \rho\frac{V^2}{2}$$

density	scalar variable, constant variable, or constant number
pressure	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Kinetic **Energy**

KinEn (any part(s), velocity, density)
Computes a scalar variable whose value is the kinetic energy E_k defined as:

$$E_k = \frac{1}{2}\rho V^2$$

where ρ = density
V = Velocity variable

velocity	vector variable
density	scalar variable, constant variable, or constant number

Enthalpy

Enthalpy (any part(s), density, total energy, velocity, ratio of specific heats)
Computes a scalar variable which is Enthalpy h defined as:

$$h = \gamma\left(\frac{E}{\rho} - \frac{V^2}{2}\right)$$

where: E = total energy per unit volume
ρ = density
V = velocity magnitude
γ = ratio of specific heats

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Normalized **Enthalpy**

EnthalpyNorm (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound)
Computes a scalar variable which is Normalized Enthalpy h_n defined as:

$$h_n = h / h_i$$

where: h = enthalpy
h_i = freestream enthalpy

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number

Stagnation **Enthalpy**

EnthalpyStag (any part(s), density, total energy, velocity, ratio of specific heats)
Computes a scalar variable which is Stagnation Enthalpy h_o defined as:

$$h_o = h + \frac{V^2}{2}$$

where: h = enthalpy
V = velocity magnitude

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Normalized Stagnation **Enthalpy**

EnthalpyNormStag (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, freestream velocity magnitude)
Computes a scalar variable which is Normalized Stagnation Enthalpy h_{on} defined as:

$$h_{on} = h_o / h_{oi}$$

where: h_o = stagnation enthalpy
h_{oi} = freestream stagnation enthalpy

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
freestream velocity magnitude	constant variable or constant number

Entropy

Entropy (any part(s), density, total energy, velocity, ratio of specific heats, gas constant, freestream density, freestream speed of sound)
Computes a scalar variable which is Entropy s defined as:

$$s = ln\left(\frac{\frac{p}{p_i}}{\left(\frac{\rho}{\rho_i}\right)^{\gamma}}\right)\left(\frac{R}{\gamma - 1}\right)$$

where: R = gas constant
ρ = density
ρ_i = freestream density
p = pressure
p_i = freestream pressure = $(\rho_i c_i^2)/\gamma$
c_i = velocity magnitude
γ = ratio of specific heats

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable

ratio of specific heats	scalar variable, constant variable, or constant number
gas constant	constant variable or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number

Flow

Flow (any 1D or 2D part(s), velocity).

Computes a constant variable whose value is the volume flow rate Q_c defined as:

$$Q_c = \int_S (V \cdot \hat{n}) dS$$

where V = Velocity vector

$\hat{n}$ = Unit vector normal to surface S

S = 1D or 2D domain

velocity	vector variable

Note: Normal for a 1D part will be parallel to the plane of the plane tool.

Note also: To calculate mass flow rate, multiply Velocity vector by the Density scalar and then substitute this vector value in for the velocity vector in the above equation.

Flow Rate

FlowRate (any 1D or 2D part(s), velocity).

Computes a scalar V_n which is the component of velocity normal to the surface, defined as:

$$V_n = V \cdot \hat{n}$$

where V = Velocity

$\hat{n}$ = Unit vector normal to surface S

S = 1D or 2D domain

velocity	vector variable

Note: This function is equivalent to calculating the dot product of the velocity vector and the surface normal (using the Normal function).

Fluid Shear

FluidShear(2D part(s), velocity magnitude gradient, viscosity)

Computes a scalar variable tau whose value is defined as:

$$\text{tau} = \mu \frac{\partial V}{\partial n}$$

where tau = shear stress

μ = dynamic viscosity

$\frac{\partial V}{\partial n}$ = Velocity gradient in direction of surface normal

Hints: To compute fluid shear stress:

1. Use Gradient function on velocity to obtain "Velocity Grad" variable in the 3D part(s) of interest.
2. Create a clip part or extract the outer surface of the part using part extract (create a 2D part from the 3D part(s) used in 1.) a surface on which you wish to see the fluid shear stress.
3. Create the surface normal vector using the Normal calculator function on the 2D part, then make sure the normal faces into the flow.
4. Calculate the DOT product of the Normal and the Velocity magnitude gradient variable.
5. Compute Fluid Shear variable (on the 2D clip surface of 2 using normal of velocity gradient magnitude in direction of surface normal from 4).

velocity gradient	vector variable
viscosity	scalar variable, constant variable, or constant number

Fluid Shear Stress Max

FluidShearMax (2D or 3D part(s), velocity, density, turbulent kinetic energy, turbulent dissipation, laminar viscosity)

Computes a scalar variable Σ defined as:

$$\Sigma = F/A = (u_t + u_l)E$$

where F = *force*
A = *unit area*
u_t = *turbulent (eddy) viscosity*
u_l = *laminar viscosity (treated as a constant)*
E = *local strain*

The turbulent viscosity u_t is defined as:

$$u_t = \frac{\rho 0.09\kappa^2}{\varepsilon}$$

where ρ = *density*
κ = *turbulent kinetic energy*
ε = *turbulent dissipation*

A measure of local strain *E* (i.e. local elongation in 3 directions) is given by

$$E = \sqrt{(2tr(D \cdot D))}$$

where

$$2tr(D \cdot D) = 2((d_{11})^2 + (d_{22})^2 + (d_{33})^2) + ((d_{12})^2 + (d_{13})^2 + (d_{23})^2)$$

given the *Euclidean norm* defined by

$$tr(D \cdot D) = (d_{11})^2 + (d_{22})^2 + (d_{33})^2 + \frac{1}{2}((d_{12})^2 + (d_{13})^2 + (d_{23})^2) \;;$$

and the rate of deformation tensor *dij* defined by

$$D = [d_{ij}] = \frac{1}{2}\begin{bmatrix} 2d_{11} & d_{12} & d_{13} \\ d_{21} & 2d_{22} & d_{23} \\ d_{31} & d_{32} & 2d_{33} \end{bmatrix}$$

with $d_{11} = {}^1u/{}^1x$
$d_{22} = {}^1v/{}^1y$
$d_{33} = {}^1w/{}^1z$
$d_{12} = {}^1u/{}^1y + {}^1v/{}^1x = d_{21}$
$d_{13} = {}^1u/{}^1z + {}^1w/{}^1x = d_{31}$
$d_{23} = {}^1v/{}^1z + {}^1w/{}^1y = d_{32}$

given the strain tensor e_{ij} defined by $e_{ij} = \frac{1}{2}d_{ij}$

velocity	vector variable
density	scalar variable, constant variable, or constant number
turbulent kinetic energy	scalar variable
turbulent dissipation	scalar variable
laminar viscosity	constant variable or constant number

Force

Force(2D part(s), pressure)

Computes a vector variable whose value is the force F defined as:

$$F = pA$$

where p = pressure
A = unit area

Note: The force acts in the surface normal direction.

pressure	scalar variable

Force 1D *Force1D*(1D planar part(s), pressure, surface normal)

Computes a vector variable whose value is the force F defined as:

$$F = pL$$

where p = pressure
L = unit length times 1

Note: The force acts in the part's normal direction (in plane).

pressure	scalar variable
surface normal	vector variable

Gradient *Grad* (2D or 3D part(s), scalar or vector(Magnitude will be used))

Computes a vector variable whose value is the gradient $GRAD_f$ defined as:

$$GRAD_f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k}$$

where f = any scalar variable (or the magnitude of the specified vector)
x, y, z = coordinate directions
i, j, k = unit vectors in coordinate directions

Gradient Approximation *GradApprox* (2D or 3D part(s), scalar or vector(Magnitude will be used))

Same as Gradient, except all elements are first subdivided into triangles (for 2D) or tetrahedrons (for 3D) and a closed-form solution is done on the subdivided element's nodal values (only applicable for per node variables). This is basically a quicker, linear approximation of the regular gradient.

Gradient Tensor *GradTensor* (2D or 3D part(s), vector)

Computes a tensor variable whose value is the gradient $GRAD_F$ defined as:

$$GRAD_F = \frac{\partial F}{\partial x}\hat{i} + \frac{\partial F}{\partial y}\hat{j} + \frac{\partial F}{\partial z}\hat{k}$$

where F = any vector variable
x, y, z = coordinate directions
i, j, k = unit vectors in coordinate directions

Gradient Tensor Approximation *GradTensorApprox* (2D or 3D part(s), vector)

Same as Gradient Tensor, except all elements are first subdivided into triangles (for 2D) or tetrahedrons (for 3D) and a closed-form solution is done on the subdivided element's nodal values (only applicable for per node variables). This is basically a quicker, linear approximation of the regular gradient tensor.

Helicity:

Helicity Density *HelicityDensity*(any part(s), velocity)

Computes a scalar variable H_d whose value is:

$$H_d = V \bullet \Omega$$

where: V = Velocity
Ω = Vorticity

velocity	vector variable

Relative Helicity *HelicityRelative*(any part(s), velocity)

Computes a scalar variable H_r whose value is:

$$H_r = \cos\phi = \frac{V \bullet \Omega}{|V||\Omega|}$$

where: ϕ = the angle between the velocity vector and the vorticity vector.

velocity	vector variable

Filtered Relative Helicity *HelicityRelFilter*(any part(s), velocity, freestream velocity magnitude).

Computes a scalar variable H_{rf} whose value is:

$$H_{rf} = H_r \quad \text{, if } |H_d| \geq filter$$

or $$H_{rf} = 0 \quad \text{, if } |H_d| < filter$$

where H_r = relative helicity (as described above)

H_d = helicity density (as described above)

$filter = 0.1(V_\infty)^2$

velocity	vector variable
freestream velocity magnitude	constant variable or constant number

Iblanking Values *IblankingValues* (Any iblanked structured part(s))

Computes a scalar variable whose value is the iblanking flag of selected parts.

Integrals:

Line Integral *IntegralLine* (1D part(s), scalar or (vector, component))

Computes a constant variable whose value is the integral of the input variable over the length of the specified 1D part(s).

Surface Integral *IntegralSurface* (2D part(s), scalar or (vector, component))

Computes a constant variable whose value is the integral of the input variable over the surface of the specified 2D part(s).

Volume Integral *IntegralVolume* (3D part(s), scalar or (vector, component))

Computes a constant variable whose value is the integral of the input variable over the volume of the specified 3D part(s).

Kinetic Energy (See under Energy)

Length *Length* (any 1D part(s))

Computes a constant variable whose value is the length of selected parts. While any part can be specified, it will only return a nonzero length if the part has 1D elements.

Line Integral See Line Integral under **Integrals**.

LineVectors *LineVectors* (any 1D part(s))

Computes a nodal, vector variable which is the vector beginning at each node to the next node in the connectivity of the 1D part. This vector indicates the direction of the line segments.

$$Vec_i = \left[(Px_{i+1} - Px_i)\ (Py_{i+1} - Py_i)\ (Pz_{i+1} - Pz_i)\right]$$

where:

Vec_i = Vector with origin at point i, with i from 1 to n-1.

(Pxi,Pyi,Pzi) = Coordinates of Point i of 1D part

n = Number of points in the 1D part

Lambda2

Lambda2 (any part(s), Grad_Vel_x, Grad_Vel_y, Grad_Vel_z)
Computes a scalar variable which is the second eigenvalue, or λ_2, of the second invariant (or Q-criterion) of the velocity gradient tensor. Vortex shells may then be visualized as an iso-surface of $\lambda_2 = 0$.

where

Explicitly calculate the three components of Velocity

Vel_x = Velocity[X] = x-component of the velocity vector

Vel_y = Velocity[Y] = y-component of the velocity vector

Vel_z = Velocity[Z] = z-component of the velocity vector

and then

Grad_Vel_x = Grad(any part(s), Vel_x) = gradient of x component Velocity

Grad_Vel_y = Grad(any part(s), Vel_y) = gradient of y component Velocity

Grad_Vel_z = Grad(any part(s), Vel_z) = gradient of z component Velocity

where

Velocity = velocity vector variable

Note: Common mistake is to try to calculate the Gradient from the component of the velocity without using the intermediate Vel_x, Vel_y, and Vel_z variables. For example **this is wrong and will use only the velocity magnitude:**

Grad_Vel_x = Grad(any part(s), Velocity[X])

Note:

This is a User-Defined Math Function (UDMF) which may be modified and recompiled by the user. Please see the EnSight Interface Manual for more details (Help>Interface Manual, Chapter 4 User-Defined Math Functions).

Algorithm

The three gradient vectors of the components of the velocity vector constitute the velocity gradient tensor. Using the 9 components of this (anti-symmetric) velocity gradient tensor, ∇v, construct both the symmetric, S, and the anti-symmetric, Ω, parts of the velocity gradient tensor,

$$\nabla v = S + \Omega$$

where

$$S = \frac{1}{2}[\nabla v + (\nabla v)^T]$$

$$\Omega = \frac{1}{2}[\nabla v - (\nabla v)^T]$$

then combine to compute the symmetric tensor

$$Q = S^2 + \Omega^2$$

Next compute and sort the eigenvalues of Q (using Jacobi eigen analysis), and assign the 2nd eigenvalue, or λ_2, as the scalar value at the node.

$$\lambda_1 < \lambda_2 < \lambda_3$$

The vortex is to be visualized as an iso-surface with

$$\lambda_2 = 0$$

See also the Q_criteria calculator function.

References

Haller, G., "An objective definition of a vortex," Journal of Fluid Mechanics, 2005, vol. 525, pp. 1-26.

Jeong, J. and Hussain, F., "On the identification of a vortex," Journal of Fluid Mechanics, 1995, vol. 285, pp. 69-94.

Mach Number *Mach* (any part(s), density, total energy, velocity, ratio of specific heats)

Computes a scalar variable whose value is the Mach number M defined as:

$$M = \frac{u}{\sqrt{\frac{\gamma p}{\rho}}} = \frac{u}{c}$$

where m = momentum

ρ = density

u = speed, computed from velocity input.

γ = ratio of specific heats (1.4 for air)

p = pressure (see *Pressure* below)

c = speed of sound

See Total Energy in this section for a description.

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Make Scalar at Elements *MakeScalElem* (any part(s), constant number or constant variable)

Assigns the specified constant value to each element, making a scalar variable.

Make Scalar at Nodes *MakeScalNode* (any part(s), constant number or constant variable)

Assigns the specified constant value to each node, making a scalar variable.

Make Vector *MakeVect* (any part(s), scalar or zero, scalar or zero, scalar or zero)

Computes a vector variable formed from scalar variables. First scalar becomes the X component of the vector, second scalar becomes the Y component, and the third scalar becomes the Z component. A zero can be specified for some of the scalars, creating a 2D or 1D vector field.

Massed Particle Scalar *MassedParticle* (massed particle trace part(s))

This scalar creates a massed-particle per element scalar variable for each of the parent parts of the massed-particle traces. This per element variable is the mass of the particle times the sum of the number of times each element is exited by a mass-particle trace. See Particle-Mass Scalar on Boundaries in Chapter 7

Mass-Flux Average

MassFluxAvg (any 1D or 2D part(s), scalar, velocity, density)

Computes a constant variable whose value is the mass flux average b_{avg} defined as:

$$b\text{avg} = \frac{\oint_A \rho b(V \bullet N) dA}{\oint_A \rho (V \bullet N) dA} = \frac{MassFluxOfScalar}{MassFlux} = \frac{Flow(plist, b\rho V)}{Flow(plist, \rho V)}$$

where b = any scalar variable, i.e. pressure, mach, a vector component, etc.
ρ = density (constant or scalar) variable
V = velocity (vector) variable
dA = area of some 2D domain
N = unit vector normal to dA

scalar	any scalar variable, i.e. pressure, mach, a vector component, etc
velocity	a vector variable
density	scalar variable, constant variable, or constant number

MatSpecies

MatSpecies (any model part(s), any material(s), any specie(s), scalar per element).

Computes a scalar per element variable whose value σ is the sum of all specified material and species combinations multiplied by the specified element variable on specified 'model' parts with defined material species.

$$\sigma = e_s \sum ms_{ij}$$

where

e_s = scalar per element variable value or value

ms_{ij} = $m_i * s_j$

= The product of the material fraction m_i and its corresponding specie value s_j

= 0, if specie s_j does not exist for material m_i

= m_i, if no species are specified.

This function only operates on model part(s) with pre-defined species. The specified material(s) can either be a list of materials or a single material value. The specified species can either be a list, a single specie, or no specie (i.e. a null species list which then computes an element value based on only material fraction contributions). The scalar per element value can either be an active variable, or a scalar value (i.e. the value 1 would give pure material fraction and/or specie value extraction).

Both material and specie names are selected from the context sensitive Active Variables list which changes to a Materials list and then a Species List for their respective prompts.

For more information on Species see Species under 7.19 Material Parts Create/ Update, and both MATERIAL Section under EnSight Gold Case File Format, and Example Material Dataset (with Species) in User Manual section 11.1.

MatToScalar

MatToScalar (any model part(s), a material).

Computes a scalar per element variable whose value s is the specified material's value m of the element on the specified part(s).

$s = m$

where

s = scalar per element variable value of each element

m = the corresponding material fraction value of each element

This function only operates on model part(s) with pre-defined materials that are given by sparse mixed material definitions. Only one material may be converted into one per element scalar variable at a time. The material cannot be the null material.

For more information on Materials,(see Chapter 7.19, Material Parts Create/Update) , and both MATERIAL Sections under EnSight Gold Case File Format, and Example Material Dataset in User Manual section 11.1.

Max *Max* (any part(s), scalar or (vector, component))

Computes a constant variable whose value is the maximum value of the scalar (or vector component) in the parts selected. The component is not requested if a scalar is selected.

[component]	if vector variable, magnitude is the default, or specify [x], [y], or [z]

Min *Min* (any part(s), scalar or (vector, component))

Computes a constant variable whose value is the minimum value of the scalar (or vector component) in the parts selected.

[component]	if vector variable, magnitude is the default, or specify [x], [y], or [z]

Moment *Moment* (any part(s), vector, component).

Computes a constant variable (the moment about the cursor tool location) whose value is the *x*, *y*, or *z* component of Moment M.

$$M_x = \Sigma(F_y d_z - F_z d_y)$$

$$M_y = \Sigma(F_z d_x - F_x d_z)$$

$$M_z = \Sigma(F_x d_y - F_y d_x)$$

where F_i = force vector component in direction *i* of vector *F*(x,y,z) = (Fx,Fy,Fz)

d_i = signed moment arm (the perpendicular distance from the line of action of the vector component F_i to the moment axis (which is the current cursor tool position)).

vector	any vector variable
component	[X], [Y], or [Z]

MomentVector *MomentVector* (any part(s), force vector).

Computes a *nodal* vector variable (the moment is computed about each point of the selected parts) whose value is the *x*, *y*, or *z* component of Moment M.

$$M_x = \Sigma(F_y d_z - F_z d_y)$$

$$M_y = \Sigma(F_z d_x - F_x d_z)$$

$$M_z = \Sigma(F_x d_y - F_y d_x)$$

where F_i = force vector component in direction *i* of vector *F*(x,y,z) = (Fx,Fy,Fz)

d_i = signed moment arm (the perpendicular distance from the line of action of the vector component F_i to the moment axis (model point position)).

force vector	any vector variable (per node or per element)

Momentum *Momentum*(any part(s), velocity, density).

Computes a vector variable *m*, which is:

$$m = \rho V$$

where ρ = density
V = velocity

velocity	a vector variable
density	scalar variable, constant variable, or constant number

Node Count *NodeCount* (any part(s))

Produces a constant variable containing the node count of the part(s) specified.

Node to Element *NodeToElem* (any part(s), node-based scalar or vector).

Averages a node based variable to produce an element based variable.

Normal *Normal* (2D part(s) or 1D planar part(s))

Computes a vector variable which is the normal to the surface at each element for 2D parts, or for 1D planar parts - lies normal to the 1D elements in the plane of the part.

Normal Constraints *NormC* (2D or 3D part(s), pressure, velocity, viscosity)

Computes a constant variable whose value is the Normal Constraints *NC* defined as:

$$NC = \int_S \left(-p + \mu\frac{\partial V}{\partial n}\hat{n}\right) dS$$

where p = pressure
V = velocity
μ = dynamic viscosity
n = direction of normal
S = border of a 2D or 3D domain

pressure	scalar variable
velocity	vector variable
viscosity	scalar variable, constant variable, or constant number

Normalize Vector *NormVect* (any part(s), vector)

Computes a vector variable whose value is a unit vector U of the given vector V.

$$U = \frac{V(V_x, V_y, V_z)}{\|V\|}$$

where: V = vector variable field

$$\|V\| = \sqrt{V_x^{\,2} + V_y^{\,2} + V_z^{\,2}}$$

Offset Field *OffsetField* (2D or 3D part(s))

Computes a scalar field of offset values. The values will be in model distance units perpendicular to the boundary of the part. Note that an isosurface created in this field would mimic the part boundary, but at the offset distance into the field.

Offset Variable *OffsetVar*(2D or 3D part(s), scalar or vector, constant offset value)

Computes a scalar (or vector) variable defined as the offset value into the field of that variable that exists in the normal direction from the boundary of the part. This assigns near surface values of a variable to the surface of 3D parts or to a 2D part from the neighboring field.

constant offset value	constant number (constant variable is not valid)

Pressure *Pres* (any part(s), density, total energy, velocity, ratio of specific heats)

Computes a scalar variable whose value is the pressure *p* defined as:

$$p = (\gamma - 1)\rho\left(\frac{E}{\rho} - \frac{1}{2}V^2\right)$$

where: m = momentum
E = internal energy
ρ = density
V = velocity = m/ρ
γ = ratio of specific heats (1.4 for air)

density	scalar variable, constant variable, or constant number

total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Pressure Coefficient

PresCoef (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, freestream velocity magnitude)

Computes a scalar variable which is Pressure Coefficient C_P defined as:

$$C_P = \frac{p - p_i}{\frac{\rho_i V_i^2}{2}}$$

where: p = pressure

p_i = freestream pressure

ρ_i = freestream density

V_i = freestream velocity magnitude

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
freestream velocity magnitude	constant variable or constant number

Dynamic **Pressure**

PresDynam (any part(s), density, velocity)

Computes a scalar variable which is Dynamic Pressure q defined as:

$$q = \frac{\rho V^2}{2}$$

where: ρ = density

V = velocity magnitude

See also: Kinetic Energy

density	scalar variable, constant variable, or constant number
velocity	vector variable

Normalized **Pressure**

PresNorm (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound)

Computes a scalar variable which is Normalized Pressure p_n defined as:

$$p_n = p / p_i$$

where: p_i = freestream pressure = $1/\gamma$

γ = ratio of specific heats

p = pressure

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number

Log of Normalized **Pressure**

PresLogNorm (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound)

Computes a scalar variable which is the natural log of Normalized Pressure defined as: $lnp_n = ln(p/p_i)$

where: p_i = freestream pressure = $1/\gamma$

γ =ratio of specific heats

p = pressure

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number

Stagnation **Pressure**

PresStag (any part(s), density, total energy, velocity, ratio of specific heats)

Computes a scalar variable which is the Stagnation Pressure p_o defined as:

$$p_o = p\left(1 + \left(\frac{\gamma - 1}{2}\right)M^2\right)^{(\gamma/(\gamma - 1))}$$

where: p = pressure

γ = ratio of specific heats

M = mach number

Note: In literature, stagnation pressure is used interchangeably with total pressure. The stagnation pressure (or total pressure) use two different equations depending upon the flow regime: compressible or incompressible. EnSight has chosen to define Stagnation Pressure using the compressible flow equation (above), and Total Pressure using the incompressible flow equation (see Total Pressure below).

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Normalized Stagnation **Pressure**

PresNormStag (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, freestream velocity magnitude)

Computes a scalar variable which is Normalized Stagnation Pressure p_{on}

defined as: $p_{on} = p_o/p_{oi}$

where: p_o = stagnation pressure

p_{oi} = freestream stagnation pressure

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
freestream velocity magnitude	constant variable or constant number

Stagnation **Pressure** Coefficient

PresStagCoef (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, freestream velocity magnitude)

Computes a scalar variable which is Stagnation Pressure Coefficient C_{p_o}

defined as: $$C_{p_o} = (p_o - p_i) / \left(\frac{\rho_i V^2}{2}\right)$$

where:

p_o = stagnation pressure

p_i = freestream pressure = $1/\gamma$

γ = ratio of specific heats

ρ_i = freestream density

V = velocity magnitude

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
freestream velocity magnitude	constant variable or constant number

Pitot **Pressure**

PresPitot (any part(s), density, total energy, velocity, ratio of specific heats)

Computes a scalar variable which is Pitot Pressure p_p defined as:

$$p_p = sp$$

$$s = \frac{\left(\left(\frac{\gamma+1}{2}\right)\left(\frac{V^2}{\gamma(\gamma-1)\left(\frac{E}{\rho}-\frac{V^2}{2}\right)}\right)\right)^{(\gamma/(\gamma-1))}}{\left(\left(\left(\frac{2\gamma}{\gamma+1}\right)\left(\frac{V^2}{\gamma(\gamma-1)\left(\frac{E}{\rho}-\frac{V^2}{2}\right)}\right)\right)-\left(\frac{\gamma-1}{\gamma+1}\right)\right)^{(1/(\gamma-1))}}$$

where

γ = ratio of specific heats

E = total energy per unit volume

ρ = density

V = velocity magnitude

p = pressure

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Note: *For mach numbers less than 1.0, the Pitot Pressure is the same as the Stagnation Pressure. For mach numbers greater than or equal to 1.0, the Pitot Pressure is equivalent to the Stagnation Pressure behind a normal shock.*

Pitot **Pressure** Ratio

PresPitotRatio (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound)

Computes a scalar variable which is Pitot Pressure Ratio p_{pr} defined as:

$$p_{pr} = s(\gamma - 1)\left(E - \frac{\rho V^2}{2}\right)$$

where s = (defined above in Pitot Pressure)
γ = ratio of specific heats
E = total energy per unit volume
ρ = density
V = velocity magnitude

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number

Total **Pressure**

PresT (any part(s), pressure, velocity, density)

Computes a scalar variable whose value is the total pressure p_t defined as:

$$p_t = p + \rho\left(\frac{V^2}{2}\right)$$

where ρ = density
V = velocity
p = pressure

Note: In literature, total pressure is used interchangeably with stagnation pressure. The total pressure (or stagnation pressure) use two different equations depending upon the flow regime: incompressible or compressible. EnSight has chosen to define Total Pressure using the incompressible flow equation (above), and Stagnation Pressure using the compressible flow equation (see Stagnation Pressure above).

pressure	scalar variable
velocity	vector variable
density	scalar variable, constant variable, or constant number

Q_criteria

Q_criteria (any part(s), Grad_Vel_x, Grad_Vel_y, Grad_Vel_z)

Computes a scalar variable which is the second invariant,or Q-criterion, of the velocity gradient tensor. Vortex shells may then be visualized as an iso-surface of Q-criterion > 0.

where

First you must calculate the intermediate variable:
Vel_x = Velocity[X] = x-component of the velocity vector
Vel_y = Velocity[Y] = y-component of the velocity vector
Vel_z = Velocity[Z] = z-component of the velocity vector

then calculate the gradient using the intermediate variable:
Grad_Vel_x = Grad(any part(s), Vel_x) = gradient of x component Velocity
Grad_Vel_y = Grad(any part(s), Vel_y) = gradient of y component Velocity
Grad_Vel_z = Grad(any part(s), Vel_z) = gradient of z component Velocity

with
Velocity = velocity vector variable

Note: A common mistake is to try to calculate the Gradient from the component of the velocity without using the intermediate Vel_x, Vel_y, and Vel_z variables. For example **this is wrong and will use only the velocity magnitude:**

Grad_Vel_x = Grad(any part(s), Velocity[X])

Note:

This is a User-Defined Math Function (UDMF) which may be modified and recompiled by the user. Please see the EnSight Interface Manual for more details (Help>Interface Manual, Chapter 4 User-Defined Math Functions).

Algorithm

The three gradient vectors of the components of the velocity vector constitute the velocity gradient tensor. Using the 9 components of this (anti-symmetric) velocity gradient tensor, ∇v, construct both the symmetric, S, and the anti-symmetric, Ω, parts of the velocity gradient tensor, the Q criteria is established as follows.

$$\nabla v = S + \Omega$$

where

$$S = \frac{1}{2}[\nabla v + (\nabla v)^T]$$

$$\Omega = \frac{1}{2}[\nabla v - (\nabla v)^T]$$

solving for Q (hence Q criteria) when

$$Q = \frac{1}{2}[|\Omega|^2 - |S|^2] > 0$$

which (in terms of our EnSight variables) literally reduces to

Q = - 0.5 * (DOT(Grad_Vel_x[X],Grad_vel_x[X]) +
DOT(Grad_Vel_y[Y],Grad_vel_y[Y]) + DOT(Grad_Vel_z[Z],Grad_Vel_z[Z]) +
2 * (Grad_Vel_x[Y] * Grad_Vel_y[X] + Grad_Vel_x[Z] * Grad_Vel_z[X] +
Grad_Vel_y[Z] * Grad_Vel_z[Y])) > 0

Now, to find the vorticies, create an isosurface where Q is positive (Q > 0). This is because an isosurface with positive Q isolates areas where the strength of the rotation overcomes the strain, thus making those surfaces eligible as vortex envelopes.

See also the Lambda2 calculator function.

References:

Dubief, Y and Delcayre, F., “On coherent-vortex identification in turbulence”, Journal of Turbulence, (jot.iop.org) 1 (2000) 11, pp.1-22.

Haller, G., “An objective definition of a vortex,” Journal of Fluid Mechanics, 2005, vol. 525, pp. 1-26.

Jeong, J. and Hussain, F., “On the identification of a vortex,” Journal of Fluid Mechanics, 1995, vol. 285, pp. 69-94.

Radiograph_grid *Radiograph_grid*(1D or 2D part(s), dir X, dir Y, dir Z, num_points, variable, [component]).

Computes a per element scalar variable on the designated 1D or 2D part(s), that is a directional integration from these parts of a scalar variable or vector component through the model. Think of rays being cast from the center of each element of the 1D or 2D parents in the direction specified (and long enough to extend through the model). Along each ray the desired variable is integrated and the integral value is assigned to the element from which the ray was cast. This function integrates the ray in a constant delta, grid-like fashion. You control the delta by the number of points that is specified in the integration direction. *(Please note that while this function is not generally as time consuming as the Radiograph_mesh funtion (and you have some relsolution control with the num_points argument), it still may take some computation time. You may want to set the Abort server operations performance preference to avoid being stuck in a computation loop that exceeds your patience.)*

The arguments are:
dir X = Integration direction vector x component
dir Y = Integration direction vector y component
dir Z = Integration direction vector z component
num_points = number of points along ray in the integration direction.
(integration delta wil be ray length divided by num_points)
variable = Variable that is integrated along the ray.
component = If vector variable, component to be integrated.
[X] for x component
[Y] for y component
[Z] for z component
[] for magnitude

dir X	constant number
dir Y	constant number
dir Z	constant number
num_points	constant number
component	[X], [Y], [Z], or []

Note that this function will not work properly for Server of Servers (SOS). Each portion will only give its local value.

Radiograph_mesh *Radiograph_mesh*(1D or 2D part(s), dir X, dir Y, dir Z, variable, [component]).
Computes a per element scalar variable on the designated 1D or 2D part(s), that is a directional integration from these parts of a scalar variable or vector component through the model. Think of rays being cast from the center of each element of the 1D or 2D parents in the direction specified (and long enough to extend through the model). Along each ray the desired variable is integrated and the integral value is assigned to the element from which the ray was cast. This function integrates the ray at each domain element face intersection. *(Please note that this can be a very time consuming process. You may want to set the Abort server operations performance preference to avoid being stuck in a computation loop that exceeds your patience. The Radiograph_grid function will generally be considerably quicker.)*

The arguments are:
dir X = Integration direction vector x component
dir Y = Integration direction vector y component
dir Z = Integration direction vector z component
variable = Variable that is integrated along the ray.
component = If vector variable, component to be integrated.
[X] for x component
[Y] for y component
[Z] for z component
[] for magnitude

dir X	constant number
dir Y	constant number
dir Z	constant number

component	[X], [Y], [Z], or []

Note that this function will not work properly for Server of Servers (SOS). Each portion will only give its local value.

Rectangular To Cylindrical Vector

RectToCyl (any part(s), vector)

Produces a vector variable with cylindrical components according to frame 0. (Intended for calculation purposes)

x = radial component, y = tangential component, z = z component

Server Number

ServerNumber (any part(s))

Produces a per-element scalar variable that is the server number which contains the element. Useful for decomposed models using Server-of-Servers, so the distribution can be vizualized.

Shock Plot3d

ShockPlot3d(2D or 3D part(s), density, total energy, velocity, ratio of specific heats).

computes a scalar variable *ShockPlot3d*, whose value is:

$$ShockPlot3d = \frac{V}{c} \bullet \frac{grad(p)}{|grad(p)|}$$

where V = velocity

c = speed of sound

p = pressure

$grad(p)$ = gradient of pressure

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Mesh Smoothing

SmoothMesh(any 1D or 2D part(s), number of passes, weight)

Performs a mesh “smoothing” operation. The function returns a vector variable which, when applied to the mesh as a displacement, will result in a “smoother” mesh representation. The function computes new node locations resulting from a “normalization” of the mesh elements. As a tendency it results in a mesh with equal sized elements. The algorithm applies a form of convolution to the mesh edges repeatedly (number of passes) using a weighting factor to control how much change in position is allowed in each pass. In most cases, the weight is supplied as a constant, but the weight can be specified as a nodal scalar array. This allows for local control over the region of the mesh to be smoothed. The algorithm is fully threaded. Note: nodes on the outer boundary of a mesh (or are bounded by ghost elements) are not allowed to move. A good set of initial parameters might be 50 passes with a weight constant of 0.05.

For each pass, the following formula is applied:

$$x_{i+1} = x_i + w \sum_{j=0}^{n} (x_j - x_i)$$

where x = nodal position at pass (i)

w = nodal weight

n = edge connected nodes

number of passes	the number of smoothing passes to be applied: constant
weight	fraction of the length of a node’s edges a nodel is allowed to move with each pass: nodal scalar variable or constant

SOS Constant *SOSConstant*(any part(s), variable, reduction operation (0-3))

(Note: generally this function should not be necessary. The SOSConstant functionality has been pulled into the server/SOS infrastructure. It remains for backward compatibility.) Computes a constant variable whose value is the result of applying a reduction operation on that constant variable over the values on each of the servers. If there is no SOS involved or only a single server, the result is the same as the constant variable value on the single server. The selected part is used to select the case from which the constant variable is used. The constant variable itself is specified (this can be from the dataset or a omputed value. The actual operation to be performed is selected as an integer from 0 to 3. The operation can be a simple summation of the values from each of the servers, an average of the values from the servers (note that the weight given to each server in the average is the same, so this is essentially the sum operation divided by the number of servers) or the minimum/ maximum of the values on each of the servers.

variable	constant variable (from the data or computed)
reduction operation	value from 0 to 3 that selects from the following operations: 0=sum 1=average 2=minimum 3=maximum

Spatial Mean *SpaMean*(any part(s), scalar or (vector, component))

Computes a constant variable whose value is the volume (or area) weighted mean value of a scalar (or vector component) at the current time. This value can change with time. The component is not requested if a scalar variable is used

The spatial mean is computed by summing the product of the volume (3D, or area 2D) of each element by the value of the scalar (or vector component) taken at the centroid of the element, for each element over the entire part. The final sum is then divided by the total volume (or area) of the part.

$$SpatialMean = \frac{\sum s_i vol_i}{\sum vol_i}$$

where: s_i = Scalar taken at centroid of element i

vol_i = Volume (or Area) of element i

[component]	if vector variable, magnitude is the default, or specify [x], [y], or [z]

Speed *Speed* (any part(s), velocity)

Computes a scalar variable whose value is the Speed defined as:

$$Speed = \sqrt{u^2 + v^2 + w^2}$$

where: u,v,w = velocity components in the x,y,z directions.

velocity	vector variable

Sonic Speed

SonicSpeed(any part(s), density, total energy, velocity, ratio of specific heats).

Computes a scalar variable c, whose value is:

$$c = \sqrt{\frac{\gamma p}{\rho}}$$

where γ = ratio of specific heats

ρ = density

p = pressure

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number

Statistics Moments

StatMoment (any part(s), v, function)

Computes a constant by applying a selected statistical function over all of the nodes or elements of the selected parts, given the seleted scalar or constant variable. Five functions are defined as:

$$sum = \sum_{i=1}^{N} v_i$$

$$mean = \frac{1}{N}\sum_{i=1}^{N} v_i$$

$$var = \frac{1}{N-1}\sum_{i=1}^{N} (v_i - mean)^2$$

$$skew = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{v_i - mean}{\sqrt{var}}\right)^3$$

$$kurt = \left\{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{v_i - mean}{\sqrt{var}}\right)^4\right\} - 3$$

If the variable (v) is a constant, the operation will be computed as if the variable were a nodal variable with the given value at all nodes. If the computation is over an element variable, the size of the element is not used in the computation. If volume or area weighting is desired, the variable must be pre-weighted.

Note that StatMoment(plist,scalar,0) should be used in place of the example user-defined math function, udmf_sum because StatMoment is threaded and properly handles ghost cells. The function parameters are defined as:

v	scalar variable, constant variable, or constant number
function	constant number selecting the moment to compute (0=sum, 1=mean, 2=variance, 3=skewness, 4=kurtosis)

Statistics Regression

StatRegSpa (any part(s), y, x0, x1, x2, x3, x4, weight)

Performs classical multivariate linear regression, predictiing y = f(x0,x1,x2,x3,x4). The regression is performed at the current timestep using all of the nodes/elements of the selected parts. At each node/element, the input values y, x0, x1, x2, x3, x4 and weight are evaluated and added as an observation to the regression with the supplied weight (in the range [0.0-1.0]). If the model does not require 5 inputs, any of them can be specified as the constant number 0.0 to drop it out. If the constant 1.0 is supplied as an input, an intercept will be computed. One is cautioned to avoid co-linearity in the inputs (especially easy when supplying constants as regressors). An example, to model simple linearity: y = Ax0

+ B, the function parameters would be: StatRegSpa(plist, yvar, xvar, 1., 0., 0., 0., 1.). The example specifies that all observations be weighted the same. If weighting by element volume were desired, compute a field variable of element volume, normalized by the largest individual element volume and pass that variable as the weight. The function returns a scalar constant whose value is the R-squared value for the regression.

The function parameters are defined as:

y	scalar variable, constant variable or constant number
x0, x1, x2, x3, x4	scalar variable, constant variable or constant number
weight	scalar variable, constant variable or constant number

A full set of estimated values and statistical diagnostic output are available, see: *StatRegVal1*, *StatRegVal2*

Statistics
Regression info

StatRegVal1 (any part(s), regression_variable, function)

This function returns basic statistical diagnostics for a regression computed using StatRegSpa(). The function is passed the output variable of a previously computed StatRegSpa() and the function number of a specific statistical quantity to return. The values include the standard sum of squares values for the regression as well as the R-squared value.

The function parameters are defined as:

regression_variable	a scalar variable which is the output of an earlier StatRegSpa() function
function	the statistical quantity to return (0=sum of squares error, 1=sum of squares total, 2=sum of squares model, 3=R-squared)

See also: *StatRegSpa*, *StatRegVal2*

Statistics
Regression info

StatRegVal2 (any part(s), regression_variable, function, selection)

This function returns statistical diagnostics specific to individual input coefficients for a regression computed using StatRegSpa(). The function is passed the output variable of a previously computed StatRegSpa(), the function number of the specific statistical quantity to return and the coefficient selected. The values include the sum of squares and partial sum of squares for the individual coefficients as well as the estimated coefficient itself and its standard error.

The function parameters are defined as:

regression_variable	a scalar variable which is the output of an earlier StatRegSpa() function
function	the statistical quantity to return (0=the estimated coefficient, 1=sum of squares for the variable, 2=partial sum of squares for the variable, 3=standard error for the coefficient)
selection	constant variable or constant number which selects the specific coefficient for which to retrieve the statistical quantity (0=x0, 1=x1, 2=x2, 3=x3, 4=x4)

See also: *StatRegSpa*, *StatRegVal1*

Stream Function

Stream (any 2D part(s), velocity, density)

Computes a scalar variable whose value is the Stream Function Ψ defined as:

$$d\psi = -\rho v dx + \rho u dy$$

where: u, v = velocity components in X, Y directions
ρ = density

velocity	vector variable
density	scalar variable, constant variable, or constant number

Surface Integral See Surface Integral under **Integrals**.

Computes a constant variable whose value is the integral of the input variable over the surface of the specified 2D part(s).

Swirl *Swirl* (any part(s), density, velocity).

Computes a scalar variable *Swirl*, whose value is:

$$Swirl = \frac{\Omega \bullet V}{\rho V^2}$$

where: Ω = vorticity

ρ = density

V = velocity

density	scalar variable, constant variable, or constant number
velocity	vector variable

Temperature *Temperature* (any part(s), density, total energy, velocity, ratio of specific heats, gas constant)

Computes a scalar variable whose value is the temperature T defined as:

$$T = \frac{\gamma - 1}{R}\left(\frac{E}{\rho} - \frac{1}{2}V^2\right)$$

where: m = momentum

E = total energy per unit volume

ρ = density

V = velocity = m/ρ

γ = ratio of specific heats (1.4 for air)

R = gas constant

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
gas constant	constant variable or constant number

Normalized **Temperature** *TemperNorm* (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, gas constant)

Computes a scalar variable which is Normalized Temperature T_n defined as:

$$T_n = \frac{T}{T_i}$$

where: T = temperature

T_i = freestream temperature

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
gas constant	constant variable or constant number

Log of Normalized Temperature

TemperLogNorm (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, gas constant)

Computes a scalar variable which is the natural log of Normalized Temperature

defined as: $lnT_n = ln(T/T_i)$

where: T = temperature

T_i = freestream temperature

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
gas constant	constant variable or constant number

Stagnation Temperature

TemperStag (any part(s), density, total energy, velocity, ratio of specific heats, gas constant)

Computes a scalar variable which is the Stagnation Temperature T_o

defined as: $T_o = T\left(1 + \left(\frac{\gamma - 1}{2}\right)M^2\right)$

where: T = temperature

γ = ratio of specific heats

M = mach number

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
gas constant	constant variable or constant number

Normalized Stagnation Temperature

TemperNormStag (any part(s), density, total energy, velocity, ratio of specific heats, freestream density, freestream speed of sound, freestream velocity magnitude, gas constant)

Computes a scalar variable which is Normalized Stagnation Temperature T_{on}

defined as: $T_{on} = T_o/T_{oi}$

where: T_o = stagnation temperature

T_{oi} = freestream stagnation temperature

density	scalar variable, constant variable, or constant number
total energy	scalar variable
velocity	vector variable
ratio of specific heats	scalar variable, constant variable, or constant number
freestream density	constant variable or constant number
freestream speed of sound	constant variable or constant number
freestream velocity magnitude	constant variable or constant number
gas constant	constant variable or constant number

Temporal Mean

TempMean (any model part(s), scalar or vector, timestep1, timestep2)

Computes a scalar or vector variable, depending on which type was selected, whose value is the mean value at each location (node or element) of a scalar or vector variable over the interval from timestep1 to timestep2. Thus, the resultant scalar or vector is independent of time. The temporal mean is the discrete integral of the variable over time (using the Trapezoidal Rule) divided by the total time interval. ***Because any derived parts may vary in size over time, this function is only allowed on model parts. Model parts with changing connectivity are also not allowed***.

timestep1	constant number
timestep2	constant number

Temporal Minmax Field *TempMinmaxField* (any model part(s), scalar or vector, timestep1, timestep2, 0 or 1, 0 = compute minimum, 1 = compute maximum)

Computes a scalar or vector variable, depending on which type was selected, whose value is the minimum or maximum at each location (node or element) of a scalar or vector variable over the interval from timestep1 to timestep2. Thus, the resultant scalar or vector is independent of time. If input variable is a vector then the max or min is the max or min of each component of the vector. ***Because any derived parts may vary in size over time, this function is only allowed on model parts. Model parts with changing connectivity are also not allowed***.

timestep1	constant number
timestep2	constant number

Tensor:

Tensor Component *TensorComponent*(any part(s), tensor, tensor row(1-3), tensor col(1-3))

Creates a scalar variable which is the specified row and column of a tensor variable.

S = Tij

i = given row (1 to 3)
j = given column (1 to 3)

tensor row	constant number (1 to 3)
tensor col	constant number (1 to 3)

Tensor Determinate *TensorDeterminant*(any part(s), Tensor or 3 Principals or 6 Tensor Components)

Computes the determinant of a tensor variable, three principal scalar variables, or six tensor component scalar variables. The function will require either 1 or 6 entries beyond the parts, as indicated below:

If computing from a tensor variable, a single tensor variable will be needed.
ex) TensorDeterminant(plist, Stress)

If computing from 3 principals, three scalar variables representing sigma_1, sigma_2, and sigma_3 will be needed. Additionally, you must enter a -1 constant for the last three entries.
ex) TensorDeterminant(plist, sigma_1, sigms_2, sigma_3, -1, -1, -1)

If computing from 6 tensor components, six scalar variables will be needed. They must be the following (and must be in the order shown):
T11, T22, T33, T12, T13, T23.
ex) TensorDeterminant(plist, t_11, t_22, t_33, t_12, t_13, t_23)

Tensor Eigenvalue *TensorEigenvalue*(any part(s), tensor, which number(1-3))

Computes the number (1-3) eigenvalue of the given tensor. The first eigenvalue is always the largest, while the third eigenvalue is always the smallest.

Tensor Eigenvector *TensorEigenvector*(any part(s), tensor, which number(1-3))

Computes the number (1-3) eigenvector of the given tensor.

Tensor Make *TensorMake*(any part(s), T11, T22, T33, T12, T13, T23)

Create a tensor from six scalars.

Tensor Make Asymmetric

TensorMakeAsym(any part(s), T11,T12,T13, T21,T22,T23, T31,T32,T33)

Create a tensor from 9 scalars.

Tensor Tresca

TensorTresca(any part(s), Tensor or 3 Principals or 6 Tensor Components)

Computes Tresca stress/strain from a tensor variable, three principal scalar variables, or six tensor component scalar variables. The function will require either 1 or 6 entries beyond the parts, as indicated below:

If computing from a tensor variable, a single tensor variable will be needed.
ex) TensorTresca(plist, Stress)

If computing from 3 principals, three scalar variables representing sigma_1, sigma_2, and sigma_3 will be needed. Additionally, you must enter a -1 constant for the last three entries.
ex) TensorTresca(plist, sigma_1, sigms_2, sigma_3, -1, -1, -1)

If computing from 6 tensor components, six scalar variables will be needed. They must be the following (and must be in the order shown):
T11, T22, T33, T12, T13, T23.
ex) TensorTresca(plist, t_11, t_22, t_33, t_12, t_13, t_23)

The basic equation is shown below. If needed, the principal stresses/strains are first computed from the tensor or its components.

$$\sigma_{yp} = |\sigma_1 - \sigma_3|$$

where: σ_{yp} = yield stress

σ_1 = greatest principal stress/strain

σ_3 = least principal stress/strain

Tensor Von Mises

TensorVonMises(any part(s), Tensor or 3 Principals or 6 Tensor Components)

Computes Von Mises stress/strain from a tensor variable, three principal scalar variables, or six tensor component scalar variables. The function will require either 1 or 6 entries beyond the parts, as indicated below:

If computing from a tensor variable, a single tensor variable will be needed.
ex) TensorVonMises(plist, Stress)

If computing from 3 principals, three scalar variables representing sigma_1, sigma_2, and sigma_3 will be needed. Additionally, you must enter a -1 constant for the last three entries.
ex) TensorVonMises(plist, sigma_1, sigms_2, sigma_3, -1, -1, -1)

If computing from 6 tensor components, six scalar variables will be needed. They must be the following (and must be in the order shown):
T11, T22, T33, T12, T13, T23.
ex) TensorVonMises(plist, t_11, t_22, t_33, t_12, t_13, t_23)

The basic equation is shown below. If needed, the principal stresses/strains are first computed from the tensor or its components.

$$\sigma_{yp} = \sqrt{\frac{1}{2}((\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2)}$$

where: σ_{yp} = yield stress

σ_1 = greatest principal stress/strain

σ_2 = middle principal stress/strain

σ_3 = least principal stress/strain

Velocity *Velo* (any part(s), momentum, density)

Computes a vector variable whose value is the velocity V defined as:

$$V = \frac{m}{\rho}$$

where ρ = density
m = momentum

momentum	vector variable
density	scalar variable, constant variable, or constant number

Volume *Vol* (3D part(s))

Computes a constant variable whose value is the volume of 3D parts.

Volume Integral See Volume Integral under **Integrals**.

Vorticity *Vort* (any 2D or 3D part(s), velocity)

Computes a vector variable with components ζ_x , ζ_y , ζ_z defined as:

$$\zeta_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \qquad \zeta_y = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \qquad \zeta_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

where u,v,w = velocity components in the X, Y, Z directions.

velocity	vector variable

VorticityGamma *VortGamma* (2D clip part(s), velocity, gamma function number, k (1 or 2), proximity radius, proximity option)

Computes a dimensionless scalar variable on a 2D clip part, whose value is the vorticity-gamma function Γ_k(P), defined at each node (or element centroid for cell centered data), P as follows:

$$\Gamma_k(P) = \frac{1}{S}\int_S sin(\Theta_M)dS = \frac{1}{S}\int_{(M \in S)} \left(\frac{(\overrightarrow{PM} \times \vec{V}_M) \cdot \hat{n}}{\|\overrightarrow{PM}\| \cdot \|\vec{V}_M\|}\right) dS$$

where Γ_1 = (gamma function number k=1) is a (non-Galilean invariant) vortex center approximation method "...a dimensionless scalar, with $|\Gamma_1|$ bounded by 1. It can be shown that this bound is reached at the location of the vortex centre if the vortex is axisymmmetrical. Thus, this scalar function provides a way to quantify the streamline topology of the flow in the vicinity of P and the rotation sign of the vortex. … Typically, near the vortex centre, $|\Gamma_1|$ reaches values ranging from 0.9 to 1.0" [ref.2, 1424-5]

Γ_2 = (gamma function number k=2) a (Galilean invariant) vortex boundary approximation method resulting in a dimensionless scalar, "… a local function depending only on Ω and μ, where Ω is the rotation rate corresponding to the antisymmetrical part of the velocity gradient at P and μ is the eigenvalue of the symmetrical part of this tensor. (see Note below)" [ref.2, 1425]

k = Gamma function number, 1 or 2 used to determine V_M.

P = Base node (or element centroid for per-element data) around which the proximity area (or zone of influence) is being considered.

S = Proximity area (or zone of influence) surrounding P, determined by a proximity radius measured from the base P and the proximity option. The proximity option is used to determine which set of elements to include in S as follows. If the proximity option is 0, then S includes all elements with any nodes within the proximity radius. If the proximity option is 1, then S includes only elements with every node within the proximity radius. Both options also include all elements which contain P.

M = A node (or element center) within S.

PM = The vector from the base node P to M.

V(P) = Velocity vector at P.

V(M) = Velocity vector at each M.

V_M = If the gamma function number k = 1, then $V_M = V(M)$. If k=2, $V_M = V(M) - V(P)$.

n = A unit vector normal to the 2D clip plane parent part.

Θ_M = The angle between V_M and PM. Since $-1 \le \sin(\Theta_M) \le 1$ (and n is a unit vector), then $-1 \le \Gamma_\kappa(P) \le 1$.

velocity	vector variable
gamma function number	single integer (k=1 or k= 2) which determines which value of V_M to use. A value of 1 is useful for finding vortex cores (centers) and a value of 2 is useful for finding vortex boundaries.
proximity radius	(greater than or equal to 0.0) Used to determine the proximity area around each base node or element P over which the vorticity gamma is calculated on the 2D clip part. The larger the proximity radius, the more nodes (or elements) that are used to calculate Γ and the slower the calculation. A proximity radius less than or equal to 0.0 will always use a proximity area of only elements that contain P and is the lower bound of this parameter resulting in the smallest proximity area around P (and the fastest calculation). A radius of 0.0 is a good value for the first run. **WARNING**: As the proximity radius approaches the parent plane size this calculation approaches using every node (or element) in the calculation for each node (or element) resulting in a n^2 operation whose solution may be measured in calendar time rather than wristwatch time. The radius should be large enough to sample sufficient elements for a meaningful average, but a small enough so the vortex result remains a local calculation reported at each element. Again, a radius of 0.0 is a good value for the first run, and a radius with a small scaling of the element size is a good second run.

proximity option	0 to include all cells with any nodes in the proximity area, 1 to include only cells entirely located in the proximity area. Use this option along with the radius to control the number of nodes (or elements) used in the calculation for each node (or element) P. Consider using option 0 as the radius gets small relative to element size, and 1 as the radius is enlarged. At a minimum, the proximity area will always include elements that contain P.

Note:

Recall that Ω is the rotation rate for the antisymmetrical part of the velocity gradient and that μ is the eigenvalue of the symmetric part of the tensor. The local character of the flow may be classified for Γ_2 in the following manner (based on figure 4 in [ref.2, 1425] which plots Γ_2 as a function of the ratio of Ω/μ):

$\vert \Omega/\mu \vert < 1$:	flow locally dominated by strain,	$\vert\Gamma_2\vert < 2/\pi$
$\vert \Omega/\mu \vert = 1$:	pure shear,	$\vert\Gamma_2\vert = 2/\pi$
$\vert \Omega/\mu \vert > 1$:	flow locally dominated by rotation,	$\vert\Gamma_2\vert > 2/\pi$.

References:

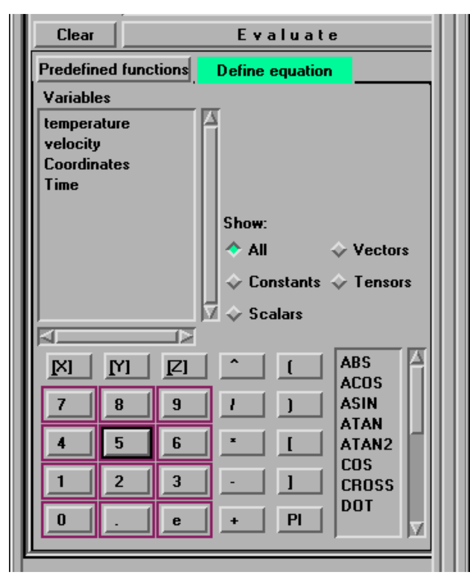
1. Jeong, J. and Hussain, F., “On the identification of a vortex,” Journal of Fluid Mechanics, 1995, vol. 285, pp. 69-94.

2. Laurent Graftieaux, Marc Michard, & Nathalie Grosjean "Combining PIV, POD and vortex identification algorithms for the study of unsteady turbulent swirling flows", Institute Of Physics Publishing Ltd in UK, Measurement Science & Technology, 12 (2001) 1422-1429

3. PSA via Distene (personal communication).

Define equations Tab Under this tab, you are provided with a variable list (which you can restrict to desired types), a calculator pad with math operators, and a list of Math functions. These can be used to construct your own equations.

Show: Can constrain the types of variables shown in the list.



Math Functions

Math functions use the syntax: function (value *or expression*). All angle arguments are in radians. For most functions the value can be either a constant, scalar, or vector and the result of the function will be of corresponding type. When you select a math function from the list, the function name and the opening “(“ appears in the Working Expression for you. However, after defining the argument(s) for the function, you have to manually provide any commas needed and a closing “)”. The Math functions include:

<table>
<tr><td colspan="2">Routines which accept argument(s) of type constant, scalar, or vector and produce the corresponding type of result: (function works on each component of a vector)</td></tr>
<tr><td>ABS(constant) absolute value = constant
or (scalar) scalar
or (vector) vector</td><td>ACOS(constant) arccosine = radian constant
or (scalar) radian scalar
or (vector) radian vector</td></tr>
<tr><td>ASIN(constant) arcsine = radian constant
or (scalar) radian scalar
or (vector) radian vector</td><td>ATAN(constant) arctangent = radian constant
or (scalar) radian scalar
or (vector) radian vector</td></tr>
<tr><td colspan="2">ATAN2(y, x) calculates ATAN(y/x) where the signs of both variables are used to determine the quadrant of the result. Returns the result in radians which is between -PI and PI (inclusive). So:

ATAN2(constant, constant) = radian constant
or (constant, scalar) = radian scalar
or (constant, vector) = radian vector
or (scalar, scalar) = radian scalar
or (scalar, vector) = radian vector
or (vector, vector) = radian vector where:
ATAN2(vector1,vector2) = (ATAN2(vector1x/vector2x), ATAN2(vector1y/vector2y), ATAN2(vector1z/vector2z))</td></tr>
<tr><td>COS(radian constant) cosine = constant
or (radian scalar) scalar
or (radian vector) vector</td><td>CROSS(vector, vector) cross product = vector</td></tr>
<tr><td colspan="2">CDF_CHISQU(constant,constant) evaluates the cumulative Chi-Squared distribution at the value v with k degrees of freedom. The result is a constant.

CDF_CHISQU(v,k) = $\dfrac{\gamma\left(\frac{k}{2}, \frac{v}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}$</td></tr>
<tr><td colspan="2">CDF_F(constant, constant, constant) evaluates the cumulative F distribution at the value v with j and k degrees of freedom. The result is a constant.

CDF_F(v, j, k) = $I_{\frac{jx}{jx+k}}\left(\frac{j}{2}, \frac{k}{2}\right)$</td></tr>
<tr><td colspan="2">CDF_NORM(constant) evaluates the cumulative normal distribution at the value v. The result is a constant.

CDF_NORM(v) = $\frac{1}{2}\left(1 + erf\left(\frac{v}{\sqrt{2}}\right)\right)$</td></tr>
<tr><td colspan="2">CDF_T(constant, constant) evaluates the cumulative Student’s T distribution at the value v with k degrees of freedom. The result is a constant.

CDF_T(v, k) = $\frac{1}{2} + v\Gamma\left(\frac{k+1}{2}\right)\left(\dfrac{\sum_{n=0}^{\infty}\left(\dfrac{\left(\frac{1}{2}\right)_n\left(\frac{k+1}{2}\right)_n}{\left(\frac{3}{2}\right)_n}\right)\left(\dfrac{\left(-\frac{v^2}{k}\right)^n}{n!}\right)}{\Gamma\left(\frac{k}{2}\right)\sqrt{\pi k}}\right)$</td></tr>
</table>

<table>
<tr><td>DOT(vector, vector) dot product = scalar
Note: DOT(velocity,velocity) is a scalar not equal to velocity^2 which is a vector.</td><td>EXP(constant) e^{value} = constant
or (scalar) scalar
or (vector) vector</td></tr>
<tr><td colspan="2">GT(constant,constant) greater of = constant
or (constant,scalar) scalar
or (constant,vector) vector
or (scalar,scalar) scalar
or (scalar,vector) vector
or (vector,vector) vector where:
GT(vector1,vector2) = (GT(vector1x, vector2x), GT(vector1y, vector2y), GT(vector1z, vector2z))</td></tr>
<tr><td>LOG(constant) ln = constant
or (scalar) scalar
or (vector) vector</td><td>LOG10(constant) $\mathbf{log_{10}}$ = constant
or (scalar) scalar
or (vector) vector</td></tr>
<tr><td colspan="2">LT(constant,constant) lesser of = constant
or (constant,scalar) scalar
or (constant,vector) vector
or (scalar,scalar) scalar
or (scalar,vector) vector
or (vector,vector) vector where:
LT(vector1,vector2) = (LT(vector1x, vector2x), LT(vector1y, vector2y), LT(vector1z, vector2z))</td></tr>
<tr><td>IF_CMP(constant,constant) compare the two values and return a constant
return value for IF_COMP(a,b):
(a < b) returns -1, (a = b) returns 0, (a > b) returns 1</td><td>IF_EQ(constant,constant)compare the two values and return a constant
return value for IF_EQ(a,b):
(a = b) returns 1, otherwise returns 0</td></tr>
<tr><td>IF_LT(constant,constant)compare the two values and return a constant
return value for IF_LT(a,b):
(a < b) returns 1, otherwise returns 0</td><td>IF_GT(constant,constant)compare the two values and return a constant
return value for IF_GT(a,b):
(a > b) returns 1, otherwise returns 0</td></tr>
<tr><td colspan="2">PDF_CHISQU(constant, constant) evaluates the Chi-Squared probability density at the value v with k degrees of freedom. The result is a constant.
PDF_CHISQU(v, k) = $\frac{1}{2^{\left(\frac{k}{2}\right)}\Gamma\left(\frac{k}{2}\right)} v^{\frac{k}{2}-1} exp\left(-\frac{v}{2}\right)$</td></tr>
<tr><td colspan="2">PDF_F(constant, constant, constant) evaluates the F probability density at the value v with j and k degrees of freedom. The result is a constant.
PDF_F(v, j, k) = $\frac{\sqrt{\frac{(jv)^j k^k}{(jv+k)^{(j+k)}}}}{xB(\frac{j}{2},\frac{k}{2})}$</td></tr>
<tr><td colspan="2">PDF_NORM(constant) evaluates the normal probability density at the value v. The result is a constant.
PDF_NORM(v) = $\frac{1}{\sqrt{2\pi}}\left(exp\left(-\frac{v^2}{2}\right)\right)$</td></tr>
<tr><td colspan="2">PDF_T(constant, constant) evaluates the Student’s T probability density at the value v with k degrees of freedom. The result is a constant.
PDF_T(v, k) = $\frac{1}{B\left(\frac{1}{2},\frac{k}{2}\right)\sqrt{k}}\left(1+\frac{v^2}{k}\right)^{-\left(\frac{k+1}{2}\right)}$</td></tr>
</table>

RMS(vector) **root-mean-square** (magnitude) = scalar. RMS(vector) is the same as SQRT(vector[X]*vector[X] + vector[Y]*vector[Y]+vector[Z]*vector[Z]) and the same as SQRT(DOT(vector,vector)) but NOT the same as SQRT(vector^2)	***RND***(constant) **round to nearest** = constant or (scalar) scalar or (vector) vector
SIN(radian constant) **sine** = constant *or* (radian scalar) scalar *or* (radian vector) vector	***SQRT***(constant) **square root** = constant *or* (scalar) scalar *or* (vector) vector
TAN(radian constant) **tangent** = constant *or* (radian scalar) scalar *or* (radian vector) vector	

Next>>> Click this button when so prompted by the dynamic instructions. It basically signals the completion of various intermediate tasks for general functions.

Calculator This on-screen calculator can usually be used in place of typing on your keyboard.

Button	Function
0 to 9	number digits
.	decimal
e	e for exponential notation
+	plus operator
–	minus operator
*	multiplication operator
/	division operator
^	exponentiation operator
PI	value for π
(	opening parentheses. For function arguments and general grouping
)	closing parentheses. For function arguments and general grouping
[	opening brackets. For components and node/element numbers
]	closing brackets. For components and node/element numbers
[X]	X component
[Y]	Y component
[Z]	Z component

(see How To Create New Variables)

6 Main Menu

This chapter describes the functions available from the Main Menu.

Figure 6-1
EnSight Main Menu

6.1 File Menu Functions

Clicking the File button in the Main Menu opens a pull-down menu which provides access to capabilities which enable you to record and play command files, connect the EnSight Client process to an EnSight Server process, setup or join collaboration sessions with other users, read data into the EnSight Server, load or re-load parts, print and save images, record currently running animations, save and restore various files, archives, and scenarios, and quit from EnSight.

File Pull-down Menu

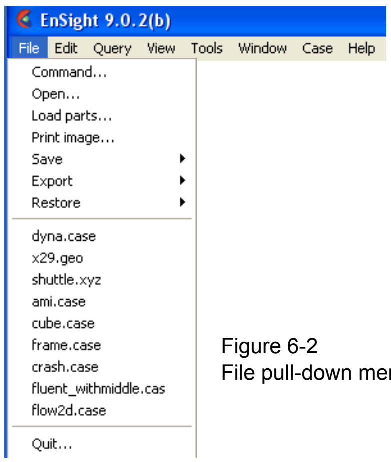


Figure 6-2
File pull-down menu

Command...

Opens the Command dialog which is used to record and play Command Files or enter command language directly
Access: Main Menu > Command...

(see Section 2.5, Command Files and How To Record and Play Command Files)

Open...

Opens the Open... dialog for use in the data loading process. Based on the suffix of the selected file, and the mapping in the `ensight_reader_extension.map` file, which is located in the EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac), the correct reader can be invoked and the data will be loaded. Or you can take more control in the specifying of the data format, files to read, and parts to load. Note that you can edit the map file to map more data file suffixes to readers.
Access: Main Menu > Open...

(see Reading and Loading Data Basics, in Section 2.1 and How to Read Data)

Load parts...

Opens the Data Part Loader dialog which is used to load parts into EnSight.
Access: Main Menu > Data (part loader)...

(see Reading and Loading Data Basics, in Section 2.1 and How To Read Data)

Print

Opens the Print/save image dialog which is used to print or save images from EnSight.
Access: Main Menu > Print/Save image...

(see Section 2.12, Saving and Printing Graphic Images, How To Print/Save an Image, and How To Output for Povray)

Save

Opens a pull-down menu which allows you to choose between the following Save options: Current Animation, Context, Commands from this session, Full backup, Geometric entities, or Scenario.
Access: Main Menu > File > Save

Session... Opens the Save current session dialog where you can specify the name of a session file to be created. This file saves information needed to restore EnSight back to the current state. This is faster than replaying a command file because it skips the intermediate steps and takes you directly to the end..
Access: Main Menu > File > Save > Session...
(See How To Save or Restore a Context File)

Context... Opens the Save current context dialog where you can specify the name of a context file to be created. This file saves information needed to reproduce the same basic imagery on a different set of data. This is faster than replaying a command file because it skips the intermediate steps and takes you directly to the end, but not all formats restore a context properly.
Access: Main Menu > File > Save > Context...
(See How To Save or Restore a Context File)

Command from current session... Opens the File selection dialog where you can specify the name of a file into which the commands used up to this point in the session will be placed.
Access: Main Menu > File > Save > Commands from this session...
(See How To Record and Play Command Files)

Full backup... Opens the Save full backup archive dialog which is used to save an entire session as an archive file which can later be used to restore EnSight to the same condition present when the archive file was made. Note that some formats do not support backup and restoring archives. Note also that a full backup is a memory dump to disk and is suitable for a fast restore of the session the next day, but is not a good option for long term archival.
Access: Main Menu > File > Save > Full backup...
(see Section 2.6, Archive Files and How To Save and Restore an Archive)

Export Opens a pull-down menu which allows you to choose between the following Export options: Current Animation, Scenario, Image, or Geometric entities.
Access: Main Menu > File > Export

Animation... Opens the Export animation dialog which is used to record the running flipbook or animated trace animation. It will not be available if no animation is currently running.
Access: Main Menu > File > Export > Current Animation...
(See How to Animate Transient Data, How to Create a Flipbook Animation, and How To Animate Particle Traces.)

Scenario... Opens the Export scenario dialog where you can create a scenario file which can be viewed by CEI's EnLiten product. EnLiten can display any scene created with EnSight and can be run standalone or be embedded in Microsoft applications.
Access: Main Menu > File > Export > Scenario...
(See How To Save Scenario)

Image.... Opens the Export image dialog which is used to save images from EnSight.
Access: Main Menu > Export > Image...

Geometric entities... Opens the Export geometric entities dialog which is used to save selected part geometric information and active variable values from EnSight. EnSight Gold, VRML 2.0, Brick of Values, or User-defined writer formats can be selected.
Access: Main Menu > File > Export > Geometric entities...
(see Section 2.10, Saving Geometry and Results Within EnSight and How To Save Geometric Entities)

Restore Opens a pull-down menu which allows you to choose to restore a context or a full backup archive file.
Access: Main Menu > File > Restore

Context... Opens the Restore context from file dialog where you can specify the name of a context

file to be applied and which case to apply it to. First read in your data, then restore the context. This will do its best to create the same basic imagery (as that when the context file was saved) to your current model.
Access: Main Menu > File > Restore > Context...

(See How To Save or Restore a Context File)

Full backup... Opens the File selection dialog which is used to specify the archive file to be restored into EnSight. This should bring EnSight to the same condition present when the archive file was saved.
Access: Main Menu > File > Restore > Full backup...

(see Section 2.6, Archive Files and How To Save and Restore an Archive)

Open recent data file Provides a pulldown of recently accessed data files. Note Windows just has a list of Most Recently Used Filenames.
Access: Main Menu > File > Open Recent Data...

Quit... Opens the Quit confirmation dialog which allows you to save a command file or/and an archive file before exiting EnSight.
Access: Main Menu > Quit...

(see Section 2.6, Archive Files)

6.2 Edit Menu Functions

Clicking the Edit button in the Main Menu opens a pull-down menu which provides access to the following features:

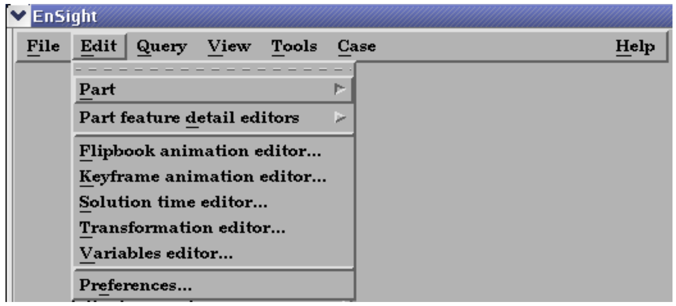


Figure 6-3
Edit pull-down menu

Part

Opens a pull-down menu which allows you to choose between the following part operations:

- Select All (see Section 3.4, Part Operations and How To Select Parts)
- Select... (see Section 3.4, Part Operations and How To Select Parts)
- Delete... (see Section 3.4, Part Operations and How To Delete a Part)
- Assign to single new viewport (see Section 3.4, Part Operations)
- Assign to multiple new viewports (see Section 3.4, Part Operations)
- Group... & Ungroup (see Section 3.4, Part Operations and How To Group Parts)
- Copy (see Section 3.4, Part Operations and How To Copy a Part)
- Extract (see Section 3.4, Part Operations and How To Extract Part Representations)
- Merge (see Section 3.4, Part Operations and How To Merge Parts)

Access: Main Menu > Edit > Part

Part feature detail editors

Opens a pull-down menu which allows you to choose between the following options to open the Feature Detail Editor:

- Selected part type... (see Section 3.1, Part Overview and Introduction to Part Creation)
- Contours... (see Section 3.3, Part Editing, Section 7.7, Contour Create/Update, and How To Create Contours)
- Clips... (see Section 3.3, Part Editing, Section 7.9, Clip Create/Update, How To Create Line Clips, How To Create Plane Clips, How To Create Quadric Clips, and How To Create IJK Clips)
- Developed surfaces...(see Section 3.3, Part Editing, Section 7.21, Developed Surface Create/Update, and How to Create Developed Surfaces)
- Elevated surfaces... (see Section 3.3, Part Editing, Section 7.14, Elevated/Offset Surface Create/Update, and How to Create Elevated Surfaces)
- Isosurfaces... (see Section 3.3, Part Editing, Section 7.8, Isosurface Create/Update, and How to Create Isosurfaces)

- Material Interface... (see Section 3.3, Part Editing, Section 7.19, Material Parts Create/Update, and How to Create Material Parts)
- Model parts... (see Section 3.3, Part Editing and Introduction to Part Creation)
- Particle traces... (see Section 3.3, Part Editing, Section 7.11, Particle Trace Create/Update, and How to Create Particle Traces)
- Profiles... (see Section 3.3, Part Editing, Section 7.13, Profile Create/Update, and How to Create Profile Plots)
- Separation/Attachment lines... (see Section 3.3, Part Editing, Section 7.17, Separation/Attachment Lines Create/Update, and How To Extract Separation/Attachment Lines)
- Shock regions/surfaces... (see Section 3.3, Part Editing, Section 7.16, Shock Surface/Region Create/Update, and How To Extract Shock Surfaces)
- Subset parts... (see Section 3.3, Part Editing, Section 7.12, Subset Parts Create/Update, and How to Create Subset Parts)
- Tensor glyphs... (see Section 3.3, Part Editing, Section 7.20, Tensor Glyph Parts Create/Update, and How to Create Tensor Glyphs)
- Vector arrows... (see Section 3.3, Part Editing, Section 7.10, Vector Arrow Create/Update, and How to Create Vector Arrows)
- Vortex cores... (see Section 3.3, Part Editing, Section 7.15, Vortex Core Create/Update, and How To Extract Vortex Cores)
- Point Parts... (see Section 3.3, Part Editing, Section 7.22, Point Parts Create/Update, How To Use Point Parts)
- Extrusion... (see Section 3.3, Part Editing, Section 7.15, Vortex Core Create/Update, and How To Extrude Parts)

Access: Main Menu > Edit > Part feature detail editors...

Flipbook animation editor... Opens the Flipbook animation editor in the Quick Interaction Area which is used to load, run, and modify Flipbook Animation sequences.

Access: Main Menu > Edit > Flipbook animation editor...

(see Section 7.2, Flipbook Animation and How To Create a Flipbook Animation)

Keyframe animation editor... Opens the Keyframe animation editor in the Quick Interaction Area which is used to create, save, restore, run, and modify Keyframe animation sequences.

Access: Main Menu > Edit > Keyframe animation editor...

(see Section 7.3, Keyframe Animation and How To Create a Keyframe Animation)

Solution time editor... Opens the Solution time editor in the Quick Interaction Area which is used to specify the currently displayed time step in a transient dataset.

Access: Main Menu > Edit > Solution time editor...

(see Section 7.1, Solution Time and How To Animate Transient Data)

Transformation editor... Opens the Transformation editor dialog which is used to precisely position parts, frames, and tools in the Graphics Window and to Save and Restore Views.

Access: Main Menu > Edit > Transformation editor...

(see Chapter 9, Transformation Control)

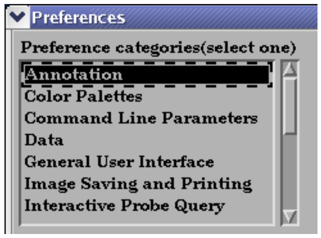
Variables editor... Opens the Feature Detail Editor (Variables) dialog which is used to obtain information about variables, change the information, and to create new variables.

Access: Main Menu > Edit > Variables editor...

(see Chapter 4, Variables)

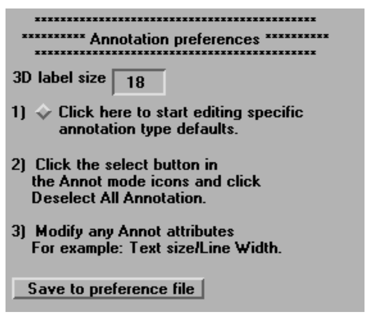
Copy Image Puts a copy of the graphics window into the system clipboard for use pasting elsewhere.

Preferences... Opens the Preferences dialog which is used to set or modify preference values for the various categories within EnSight.



In this area you can set default attributes and preferences which will be used for the current EnSight session. You may also save any of these to various preference file(s) so that they will be the defaults for future invocations of EnSight. Each of the preference categories will now be explained.

Annotation Preferences



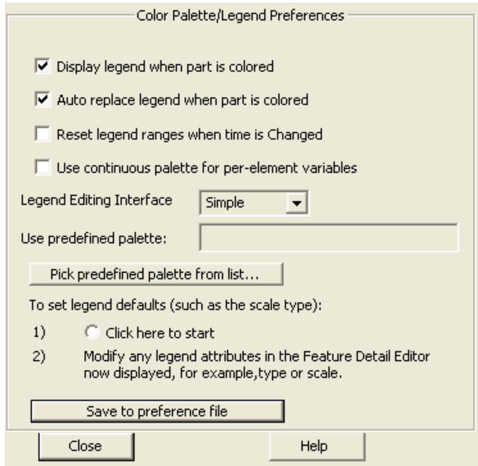
3D label size Controls the size of Node and Element label text.

Click here to start Will place you in Annotation mode in EnSight. Perform step 2) so that you are editing defaults. You can then change any annotation attributes desired and they will become the defaults for the session.

Save to preference file Will write the current annotation preferences to the preference file for future EnSight sessions.

(see How To To Set Annotation Preferences:)

Color Palettes Preferences



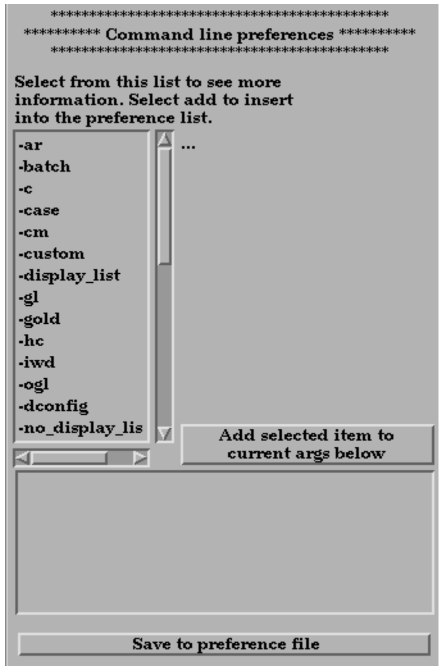
Display legend when part is colored	Will cause the legend to automatically appear when you color a part by a variable.
Auto replace legend when part is colored	Will cause legends to be automatically replaced when the current legend is no longer in use (i.e. no parts are colored by the variable) and a new variable is in use.
Reset legend ranges when time is changed	Will cause legend ranges to be reset according to variable values at the current time.
Use continuous palette for per element vars	By default the legend for Per Element variables has a “Type” of Constant. Toggle this on to change the default “Type” to Continuous.
Legend editing interface	Can be EnSight’s *Simple* or *Advanced* interface.
Use predefined palette	Allows you to enter a predefined palette name if you have predefined color palettes.
Pick predefined palette from List...	Allows you to pick from your predefined palette list.

To set legend defaults (such as the Scale Type):

Click here to start	Will allow you to modify legend default attributes.
Save to preference file	Will write the current legend and palette preferences to the preference file for future EnSight sessions.

(see How To To Set Color Palette Defaults:)

Command Line Parameter Preferences

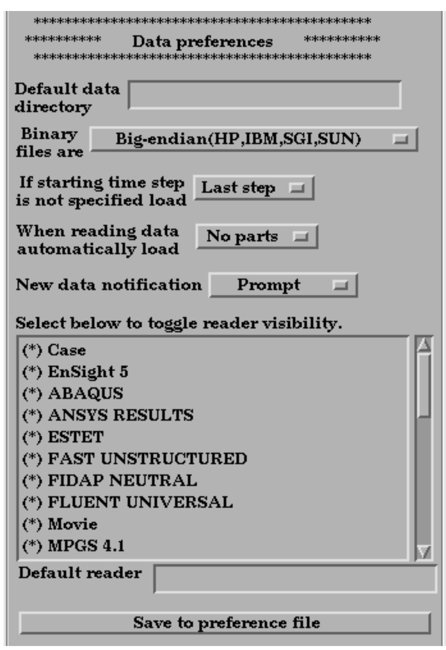


By selecting arguments from the list and hitting:

Add selected item to current args below	You can build customized command line preferences.
Save to preference file	Will save the command line preferences to the preference file for future invocations of EnSight.

(see How To To Set Command Line Preferences:)

Data Preferences



Default data directory	Will allow you to specify a default directory for data files.

Binary files are Allows you to specify the default byte order for binary files. The allowable settings are *Big-endian*, *Little-endian*, or *Native to server machine*.

If starting time step is not specified load Can be set so that the default starting time step for transient data can be either *Last step* or *First step*.

When reading data automatically load Allows you to have EnSight automatically load *All parts, First part, Last part,* or *No parts* at startup. If *No parts* is specified, the Part Loader dialog will be presented to the user at startup.

New data notification Options for dealing with notification of a change in the model dataset while EnSight is running. Please contact CEI support if you have need of this.

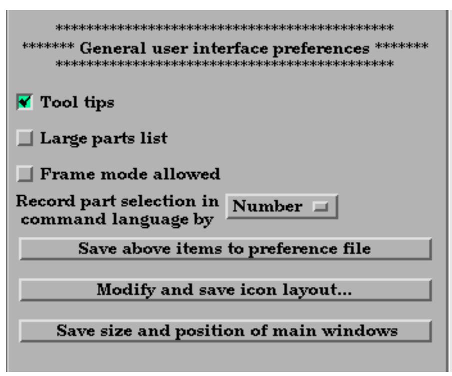
Select below to toggle reader visibility Allows you to specify which data formats will appear in the "Format" pull-down of the Data Reader dialog.

Default reader Allows you to specify the default data reader format.

Save to preference file Will save the data preferences to the preference file for future invocations of EnSight.

(see How To To Set Data Preferences:)

General User Interface Preferences



Tool tips Will cause pop-up help information to appear when the mouse is placed over certain icons while running EnSight.

Large parts list Will cause the parts list to expand to a larger size down the left of the desktop.

Frame mode allowed Will display Frame as a managed mode.

Record part selection in command language by Allows you to specify whether the part selections recorded in command language will be by part *Name* or by part *Number*.

Save above items to preference file Will save the preferences above to the preference file for future invocations of EnSight.

Modify and save icon layout.... Opens the Icon bar preferences dialog

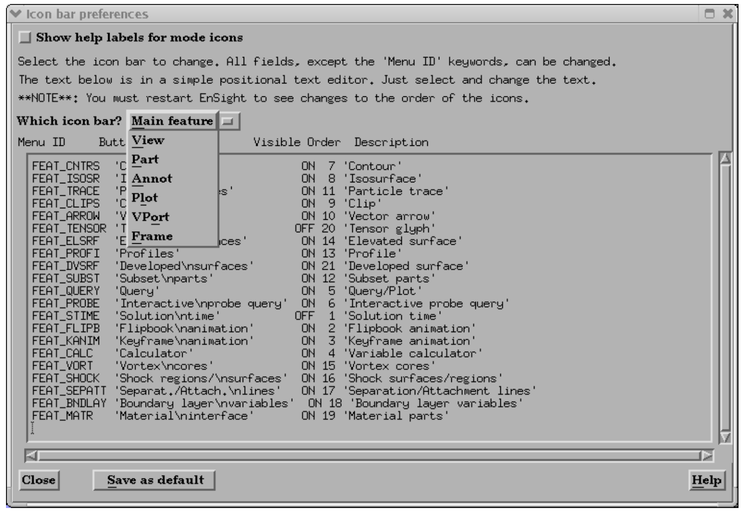


Figure 6-4
Icon Bar Preferences

Show help labels for mode icons When toggled on, the Icon name will appear beneath each icon in the Mode Icon Bar. You can customize the EnSight GUI by specifying which icons appear and their order in the Feature and Mode Icon Bars. Do **NOT** modify the Menu ID for any function. The other fields for each function may be edited within the dialog. Customization options are:

Button Name Describes the function of the icon which would be displayed if EnSight was started with no icons (command line function). Further, this is the name which will appear below the each Mode Icon when Show Help Labels For Mode Icons is toggled on.

Visible Determines the visibility of a feature icon. Must be either ON or OFF.

Order Determines the order in which the icons appear. A value of 1 will cause the icon to appear leftmost in the Main Feature Icon Bar and uppermost in the Mode Icon Bars.

Description The text description of the button which will be displayed in the Message Area when the icon is selected. You must click the Save As Default button to save any changes you have made. The Button Name and Order, if modified, will not take effect until you restart EnSight. Changes to Visibility, Description, and Show Help Labels however, will be implemented immediately upon clicking the Save as Default button (and will control these options in future EnSight sessions as well).

When EnSight is started, the icon preferences are initially read from the `$CEI_HOME/ensight92/site_preferences` directory and are then overwritten by any information in the user's preferences directory.
(see How To Customize Icon Bars)

Save size and position of main windows	Will record the location and size of the main GUI, and all dialogs that have been opened during the session or are currently open and will make those locations and sizes the default for future sessions of EnSight. Be aware also that if you had a turn-down section open in a dialog (such as General Attributes in the Feature Detail Editor dialog) when you closed it earlier in the session or at the time you choose Save window positions, this will be recorded as well and opening that dialog in future sessions will also open that turn-down section within the dialog. (see How To Save GUI Settings)

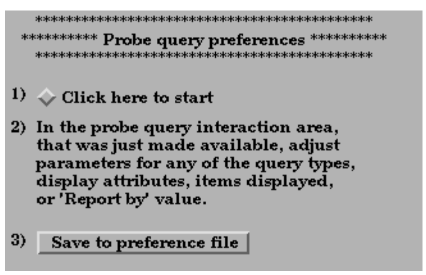
(see How To To Set General User Interface Preferences:)

Image Saving and Printing Preferences

Click here to start	Will allow you to modify default attributes for image saving and printing.
Save to preference file	Will write the current print/save preferences to the preference file for future EnSight sessions.

(see How To To Set Image Saving and Printing Preferences:)

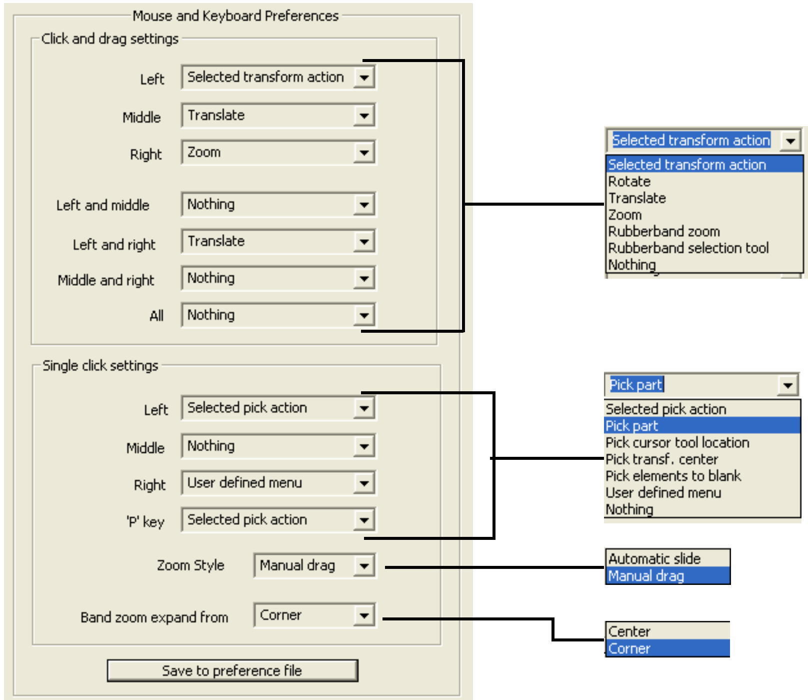
Interactive Probe Query Preferences



Click here to start	Will allow you to modify default attributes for interactive probe queries.
Save to preference file	Will write the current interactive probe query preferences to the preference file for future EnSight sessions.

(see How To To Set Interactive Probe Query Preferences:)

Mouse and Keyboard Preferences



Here you can specify the transformation and picking actions of the three mouse buttons and the keyboard 'P' key. Select the option you wish to assign to each button or combination of buttons. The options are as follows:

Selected transform action	When this option is chosen (it is the default for the left button), depressing the button and moving the mouse will perform the transformation (rotate, translate, zoom) currently selected in the Transformation Control Area on the desktop.
Rotate	When this option is chosen, depressing the button and moving the mouse will perform a rotate transformation on the model.
Translate	When this option is chosen, depressing the button and moving the mouse will perform a translate transformation on the model.
Zoom	When this option is chosen, depressing the button and moving the mouse will perform a zoom transformation on the model.
Rubberband zoom	When this option is chosen, depressing the button will cause the rubberband zoom rectangle to appear and dragging it will modify the zoom area.
Rubberband selection tool	When this option is chosen, depressing the button will bring up the rubberband selection tool that you can then manipulate.
Pick Part	When this option is chosen, depressing the button will pick the part in the part list.
Pick cursor tool location	When this option is chosen, depressing the button will pick the location for, and move the cursor tool.

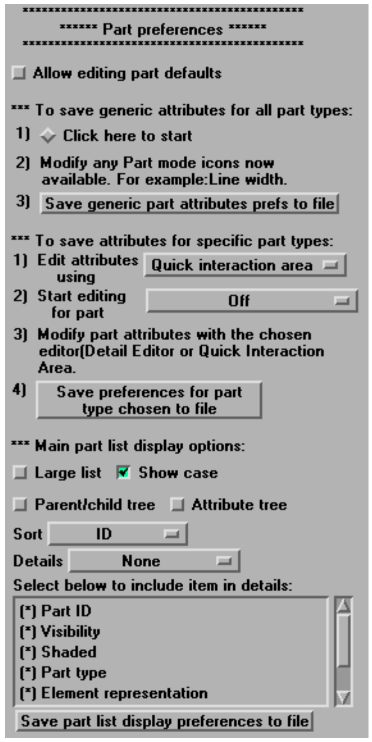
Pick transf. center	When this option is chosen, depressing the button will pick the location for the new center of transformation. Subsequent rotations, etc. will be about this picked location.
Pick elements to blank	When this option is chosen, depressing the button will blank the element under the pointer (if element blanking is enabled).
User defined menu	When this function is chosen, a user defined menu will appear as defined by the `user_defined_menus` directory contained in `$CEI_HOME/ensight92/site_preferences` or in the user's EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac).
Nothing	When this option is chosen, no function is mapped to the mouse button.

Note: One of the Mouse buttons must be assigned to Selected transform action. Macros cannot be assigned to a mouse key which has a function assigned to it. (see How To Customize Mouse Button Actions)

Zoom style	Choose method to use for zoom action. For either option, zooming stops when the mouse button is released.
Manual drag	Zoom DISTANCE is based on the distance you move your mouse when the mouse button is pressed.
Automatic slide	Zoom Velocity is based on the distance the mouse is moved when the mouse button is pressed.
Band zoom expand from	Choose method to use for rubber-band area action. For either option, area modification stops (and zooming will occur) when the mouse button is released.
Center	Zoom area will shrink and expand about the center of the rectangle.
Corner	Zoom area will shrink and expand about the selected corner of the rectangle. The opposite corner will be fixed.
Save to preference file	Will write the current mouse and keyboard preferences to the preference file for future EnSight sessions.

(see How To To Set Mouse and Keyboard Preferences:)

Parts



Allow editing part defaults	In EnSight 8 you can change a default attribute, such as line width, by making sure there are no parts selected in the main part list and changing an icon under Part mode. We used to pop up a message to warn the user that they are changing a default instead of a particular part if they forgot to select one. This preference now allows or disallows changing a default attribute. These default attributes will be used for any part created or loaded in the future, so that the user doesn't have to keep changing each new part.

Generic Attributes:

Click here to start	Will allow you to modify default visual part attributes which apply to all part types.
Save general part preferences to file	Will write the current generic part preferences to the preference file for future EnSight sessions.

Attributes For Specific Part Types:

Edit attributes using	Allows you to specify which area to use for default attribute modification - the *Quick interaction area* or the *Detail editor dialog*.
Start editing for part	Allows the user to specify the part type for which default attributes will be modified.
Save preferences for part type chosen to file	Will write the current specific part type preferences to the preference file for future EnSight sessions.

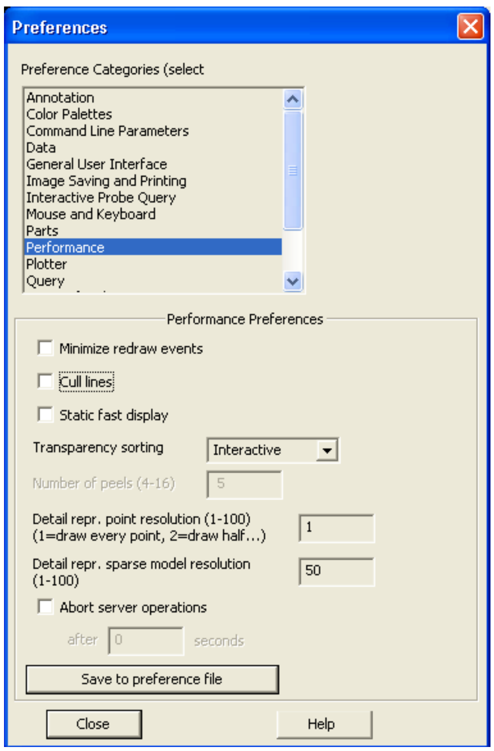
Main Part List Display Options:

Large List	Allows user to set default part list length to large if toggled on.
Show Case	Allows user to set the show case in the part list.

Parent/Child Tree	Allows the user to have the part list default to parent/child tree.
Attribute Tree	Allows the user to have the part list default to attribute tree.
Sort	Allows the user to have the part list sorted alphabetically, chronologically, or by part id by default.
Details	Allows the user to have the part list include abbreviated, full, or no details by default.
Items to include in Details:	Allows the user to select which detail items to display when details are being displayed. Can chose from Part ID, visibility, shaded, type, element representation, node ID visibility, and element ID visibility.
Save part list display preferences to file	Will write the current specific part list preferences to the preference file for future EnSight sessions.

(see How To To Set Part Preferences:)

Performance Preferences



Minimize redraw events	Will use pixel saving and restoring to advantage for window moves, etc. Useful if graphics card does fast pixel read/writes.
Cull lines	Will only draw shared lines between polygons once.
Static fast display	Will cause the fast representation to always be displayed. If this is off (the default), fast display will only be active during a transformation.

Transparency re-sort — "Interactive" causes polygons to be resorted with each transformation - so the image is always correct. Can be expensive.

"Delayed"causes the polygons to not be resorted while the mouse is down during transformations, but will be resorted when the mouse is released.

"Depth peeling", which is only available on graphics cards that support the OpenGL Shading Language, causes sorting to be done on the graphics card on a per-pixel basis.

Detail Repr. Point Resolution — Allows specification of fraction of nodes to display in Fast Display, point representation. ("1" indicates all nodes, "2" would be every other node, "3" every third node, etc.)

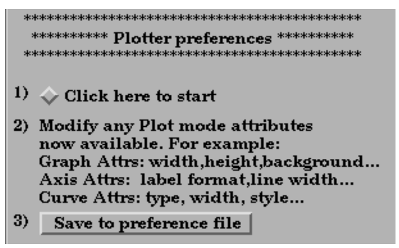
Detail Repr. Sparse Model Resolution — Allows specification of the percentage of the model geometry that will be displayed. (immediate mode only)

Abort Server Operations — Causes a timer to be set which will abort some of EnSight's CPU-intensive server operations (for example some calculator options, clipping, isosurface, isovolume, particle tracing and cuts) after the set amount of clock seconds have passed.

Save to preference file — Will write the current performance preferences to the preference file for future EnSight sessions.

(see How To To Set Performance Preferences:)

Plotter Preferences

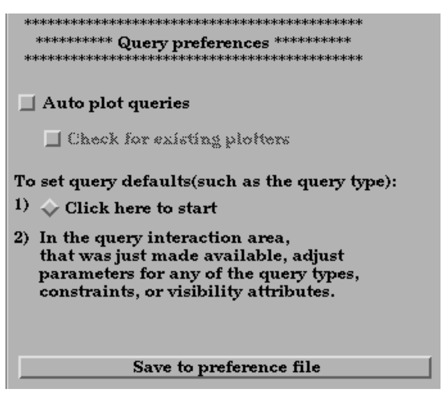


Click here to start — Will allow you to modify defaults for the various plotter graph, axis, and curve attributes.

Save to preference file — Will write the current plotter preferences to the preference file for future EnSight sessions.

(see How To To Set Plotter Preferences:)

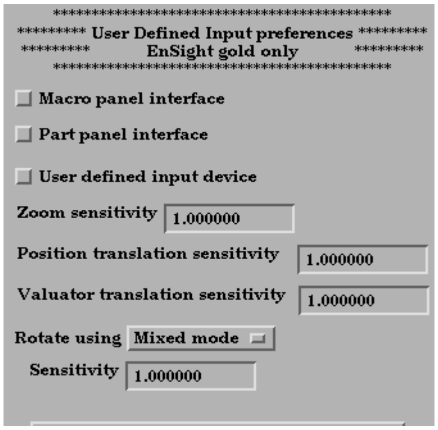
Query Preferences



Auto plot queries	Automatically plot a newly created query to a plotter. The plot will go into a new or an existing plotter, depending on the state of the "*Check for Existing Plotters*" toggle.
Check for existing plotters	Check for plotters of the same type (same X and Y units) as the new query is being autoplotted. If one exists, plot the query to the existing plotter.
Click here to start	Will allow you to modify defaults for the various query attributes.
Save to preference file	Will write the current query preferences to the preference file for future EnSight sessions.

(see How To To Set Query Preferences:)

User Defined Input Preferences



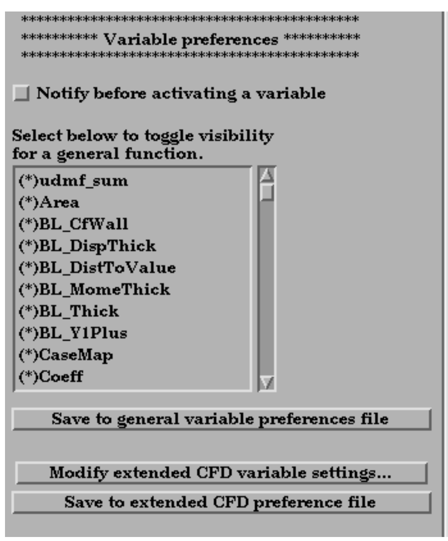
This area provides access to user defined input devices. The input devices include a Macro Panel Interface (a grid of commands that displays in the Main Graphics window and executes EnSight command files upon selection), and/or a User Defined Input Device (a virtual input device designed for - but not limited to - use with VR environments such as an Immersadesk)

Macro panel interface	Toggles on/off the user defined macro panel to the Main Graphics window. The user defined macro panel is defined in your EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac) in the `hum.define` file. (An example hum.define file is located at `$CEI_HOME/ensight92/src/cvf/udi/HUM/hum.define` on your client system.).
Part panel interface	Display a part list in the graphics window. This is helpful when in full screen mode or in a VR environment, to allow picking of parts that can be operated on via macros.
User defined input device	Toggles on/off the User Defined Input Device that is linked via a runtime library. (Steps outlining the implementation of this library and input device are found in the file: `$CEI_HOME/ensight92/src/input/README` on your client system.).

Zoom sensitivity	Specifies a positive scalar value that adjusts the Sensitivity of the zoom input device (i.e. values < 1 are slower, and values > 1 are faster).
Position translational sensitivity	Specifies a positive scalar value that adjusts the Sensitivity of a positional translation input device (i.e. values < 1 are slower, and values > 1 are faster).
Valuator translational sensitivity	Specifies a positive scalar value that adjusts the Sensitivity of a valuator translation input device (i.e. values < 1 are slower, and values > 1 are faster).
Rotate using	Opens a pull-down menu for selection of the type of input device used to record rotation transformations.
	Mixed mode — A device that returns virtual angle values where the Z rotations correspond to (literal) movement of the input device about its local Z (or roll) axis; and where the X and Y rotations correspond to translational movements of the input device with respect to its local X and Y axes.
	Direct mode — A device that returns virtual angle values that correspond to (literal) rotational movements of the input device about its local X, Y, and Z axes.
Sensitivity	Specifies a positive scalar value that adjusts the Sensitivity of the type of rotation input device selected in Rotate Using (i.e. values < 1 are slower, and values > 1 are faster). (see How To Enable User Defined Input Devices)
Save to preference file	Will write the current user defined input preferences to the preference file for future EnSight sessions.

(see How To To Set User Defined Input Preferences:)

Variables Preferences



Notify before activating a variable	Will cause you to be notified before a variable, which was going to be automatically activated, is actually activated.

Select below to toggle visibility for a general function	Toggle visibility of functions in the General Functions list of the new variable calculator dialog. There are many general functions and this allows you to limit the list to only functions that you wish to use.
Save to general variable preferences file	Will write the variable notification preference to the preference file for future EnSight sessions.
Modify extended CFD variable settings...	Opens the Extended CFD variable settings dialog. If your data defines variables or constants for density, total energy per unit volume, and momentum (or velocity), it is possible to show new variables defined by these basic variables in the Main Variables List of the GUI by utilizing the capabilities of this dialog.

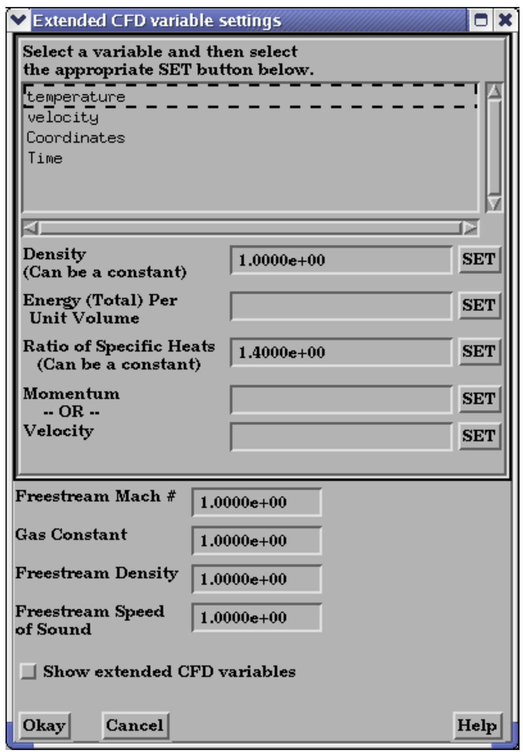


Figure 6-5
Extended CFD variable settings dialog

Density	Permits the selection of the density variable from the list (click SET after selection) or the specification of a constant value in the field provided.
Energy (Total) Per Unit Volume	Permits the selection of the energy variable from the list. Click SET after selection.
Ratio of Specific Heats	Permits the selection of the ratio of specific heats variable from the list (click SET after selection) or the specification of a constant value in the field provided.
Momentum OR Velocity	Permits the selection of the momentum or velocity variable from the list. Click SET after selection.

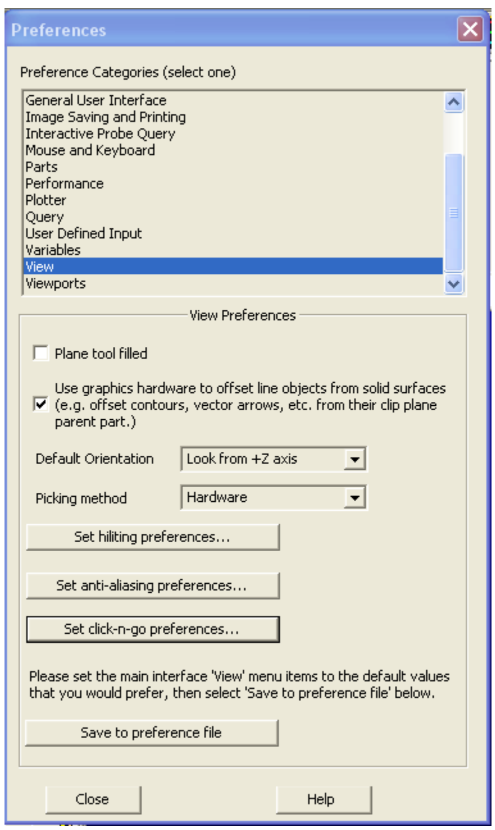
Freestream Mach #	Permits the specification of the freestream mach number in the field provided.
Gas Constant	Permits the specification of the gas constant in the field provided.
Freestream Density	Permits the specification of the freestream density value in the field provided.
Freestream Speed of Sound	Permits the specification of the freestream speed of sound value in the field provided.
Show Extended CFD Variables	When selected, all of the variables that can be derived from the information entered will be shown in the Main Variables List of the GUI. (Will not take effect until the Okay button is clicked.)
Okay	Clicking this button applies the changes made in the dialog.

(See How To Create New Variables)

Save to extended CFD preference file Will write the current extended CFD preferences to the extended CFD preference file for future EnSight sessions.

(see How To To Set Variable Preferences:)

View Preferences



Plane tool filled Will cause the plane tool to be a filled transparent surface. If it is off, the plane tool will be in line drawing mode. You can save this default to the preference file.

Use graphics hardware to offset line objects

There are two offsets employed in EnSight. This one, hardware offset, is perpendicular to the monitor screen, and done in hardware if this toggle is on. This will allow, for example, contour lines to appear closer to the viewer than their parent part so they are visible no matter what orientation the part is viewed from. The second offset is the display offset. The display offset can be set in the feature detail editor for line parts such as contour lines, particle trace lines, vector arrows, and separation/attachment lines. The display offset is the distance in the direction of the element normal (perpendicular to the surface).

Default orientation

The default axis for viewing can be selected and set.

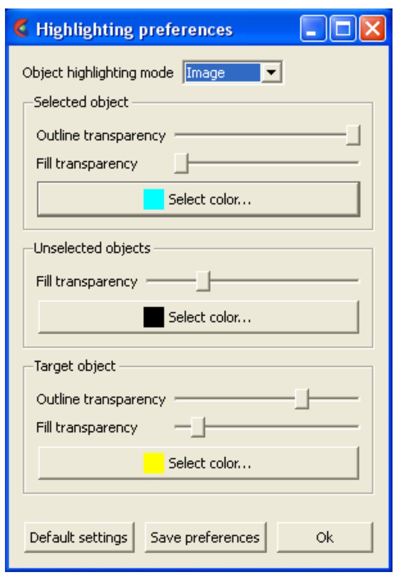
Picking Method

The default is to utilize the graphics hardware to give accelerated part pick information. Should the graphics software drivers be faulty, it may be desirable to instead to use software picking.

Set hiliting preferences

Opens the Highlighting preferences dialog where part selection and targeting feedback can be adjusted for color and intensity. To turn off highlighting completely, turn off the part highlighting toggle located above the graphics window and then Edit->Preferences->View and select the "Save to preferences file" button.

.

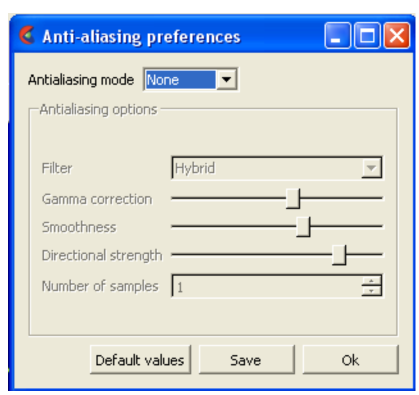


Object Highlighting Mode

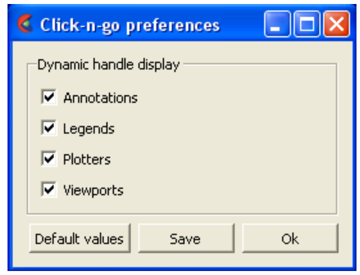
If set to Geometry will display selected objects (currently parts) by displaying the geometry as a cross hatch. This method requires a redraw of the scene when new parts are selected.

If set to Object highlighting, the following options are available: Selected Object, Unselected Objects, and Target Object.

Selected Objects	The selected object is displayed by a modified fill color as well as an outline. Both use the same color specified. When the Outline transparency slider is set to the left no outline will appear. When the Outline transparency is set to the right a full intensity outline of the color specified will appear around the object. If the Fill transparency slider is set to the left, the color of the object is not modified while if set to the right the full intensity of the color specified will be used.
Unselected Objects	Unselected objects will be blended with the color specified. When the slider is set to the left no color modification will occur. When the slider is set to the right the specified color will be shown at full intensity on the unselected objects.
Target Objects	The target object (the object under the mouse pointer) is displayed by a modified fill color as well as an outline. Both use the same color specified. When the Outline transparency slider is set to the left no outline will appear. When the Outline transparency is set to the right a full intensity outline of the color specified will appear around the target object. If the Fill transparency slider is set to the left, the color of the target object is not modified while if set to the right the full intensity of the color specified will be used.
Default Settings	Will restore the highlighting preferences to the default values
Save preferences	Will save the highlight preferences for the next session of EnSight.
OK	Closes the dialog.
Set anti-aliasing	Opens the Anti-aliasing preferences dialog, which is used to change parameters for on-screen smoothing jagged lines and edges. These anti-aliasing options apply only to on-screen, interactive use of EnSight. They do not apply to batch execution, nor to exporting an image. For example, when saving an image to file, anti aliasing is invoked separately from these preferences in the advanced tab using the number of passes.



antialiasing mode	There are three antialiasing modes in decreasing order of speed and increasing order of on-screen image quality: None, Filtered, and Multipass. Both Filtered and Multipass work only on modern graphics cards that support DirectX 10. Therefore, Anti-aliasing mode is disabled in software rendering mode (starting EnSight with the -X option). Further, zeroing the auxilliary buffers environmental variable (setenv CEI_NUM_AUX_BUFFERS 0) will disable this feature.
None	This is the default mode.
Multipass	Each time the scene needs to be redrawn, it is drawn multiple times from slightly different positions. The final image is a blend of all the images. Rendering performance will slow down in proportion to the number of samples
Number of Samples	Used with the multipass option. Indicates the number of times to redraw the scene. Rendering performance will slow down in proportion to the number of samples
Filtered	Each time the scene is drawn, an image filter is applied to the result. The shape of the filter is a gaussian curve. Filtered anti-aliasing has a very low performance cost, but is usually lower quality than multipass anti-aliasing
Filter Parameter	Used with filtered option. A pulldown that can be Small, Large, Hybrid
Small	Used with filtered option. A symmetric 3x3 filter.
Large	Used with filtered option. A symmetric 5x5 filter.
Hybrid	Used with filtered option. A 5x5 filter compressed in the direction perpendicular to the image gradient
Gamma Correction	Used with filtered option. Corrects some monitor-dependent artifacts. For example, without gamma correction, bright lines against a dark background can appear dimmer after filtering, but will appear to have the same overall brightness with the right gamma correction
Smoothness	Changes the shape of the gaussian curve. High values make the curve fall off slowly, making the image smoother and blurrier
Directional Strength	used with the hybrid algorithm to control how much the filter is compressed, perpendicular to the image gradient
Default Values	Restores the anti-aliasing preferences to the default values.
Save	Save the anti-aliasing preferences for subsequent EnSight sessions.
OK	Closes the dialog
Save to preference file	Will write the current view preferences to the preferences file for future EnSight sessions.
Set click-n-go preferences	Sets preferences for interactive interaction with created parts in the graphics window using graphical handles.



Annotations, Legends, etc Toggle ON (default) ability to interact with these items in the graphical user interface.

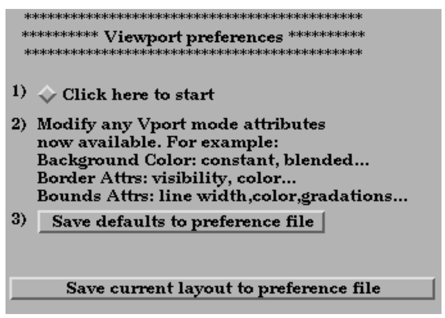
Default Values Restore selections back to default values.

Save Save settings to preferences file.

OK Close dialog

(see How To To Set View Preferences:)

Viewports Preferences



Click here to start Will allow you to modify default viewport attributes.

Modify any Vport mode attributes now available Select Vport along the left side of the main window. The attributes for viewports will appear in the iconbar for you to modify. Attributes as shown above and many others may be set.

Save defaults to preference file Will write the current viewport default values to a preference file for future EnSight sessions.

Save current layout to preference file Will write out the current viewport screen layout to a file for future EnSight sessions.

6.3 Query Menu Functions

Clicking the Query button in the Main Menu opens a pull-down menu which provides access to the following features:

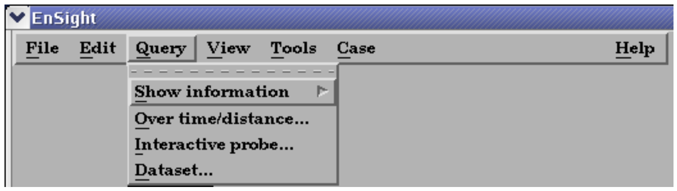


Figure 6-6
Query pull-down menu

EnSight provides several ways to examine information about variable values. You can, of course, visualize variable values with fringes, contours, vector arrows, profiles, isosurfaces, etc. Only parts with data residing on the Server host system may be queried. Thus, parts that reside exclusively on the Client host system (i.e. contours, particle traces, profiles, vector arrows) may NOT be queried.

(see Table 3–2 Part Creation and Data Location)

Show information

Opens the following pull-down menu:

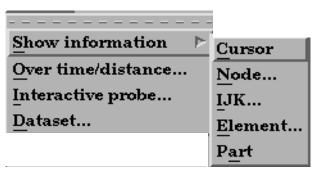


Figure 6-7
Show information pull-down menu

Access: Main Menu > Query > Show information

(see How To Get Point, Node, Element and Part Information)

Cursor

Provides the following information in the Status History Area about a Point inside of the selected Part(s) who's position you have specified with the cursor tool:

x,y,z coordinates, Frame assignment of Point, the Part that the Point is found in, the closest Node to the Point, the element id if it exists, and the active Variable values at the Point

Access: Main Menu > Query > Show information > Point

(see How To Get Point, Node, Element and Part Information and How To Use the Cursor (Point) Tool)

Node...

Opens the Query prompt dialog which is used to specify Node ID number.

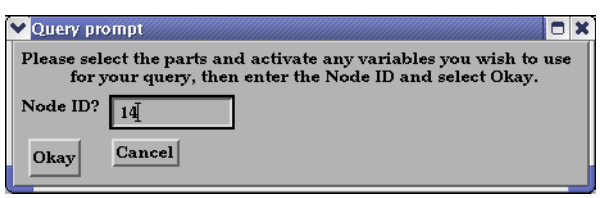


Figure 6-8
Query Prompt dialog

When the Okay button is pressed, the following information about the specified Node is shown in the Status History Area:

x,y,z coordinates, Frame assignment of Node, the Part that the Node is found in, the element id if it exists, and all active nodal Variable values at the Node

Access: Main Menu > Query > Show Information > Node...

(see How To Get Point, Node, Element and Part Information)

IJK... Opens the Query Prompt dialog which is used to specify IJK values.

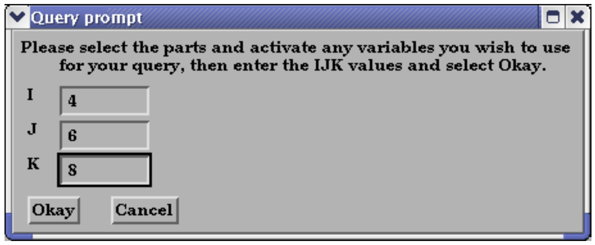


Figure 6-9
Query Prompt for IJK Values

When Okay button is pressed, the following information about the Node specified by the IJK values is shown in the Status History Area:

Node ID, Part in which the Node is located, x,y,z coordinates of the Node, Frame assignment of the Node, and the specified Variable value at the Node.

Access: Main Menu > Query > Show Information > IJK...

(see How To Get Point, Node, Element and Part Information)

Element... Opens the Query Prompt for Element ID.

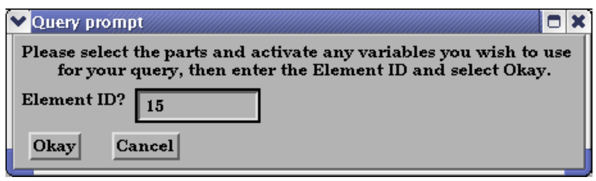


Figure 6-10
Query Prompt for Element ID

When Okay button is pressed, the following information about the Element is shown in the Status History Area:

Part in which Element is located, Type of Element, IJK bounds (if a structured mesh), Number of Nodes, Node ID numbers, information on neighboring Elements, and all active elemental Variable values at the Element.

Access: Main Menu > Query > Show Information > Element...

(see How To Get Point, Node, Element and Part Information)

Part Causes the following information about the Part to be shown in the Status History Area:

Part type (structured or unstructured), number of Nodes in Part, minimum and maximum x,y,z coordinates, Element type, min/max node labels if exist, min/max element labels if exist, and the number of Elements.

Access: Main Menu > Query > Show Information > Part

(see How To Get Point, Node, Element and Part Information)

Over time/distance... Opens the Query/Plot Editor in the Quick Interaction Area which is used to obtain information about variables and to create plots of the information.

Access: Main Menu > Query > Over time/distance...

(see Section 7.5, Query/Plot, How To Query/Plot)

Interactive probe... Opens the Interactive Probe Query Editor in the Quick Interaction Area which is used to obtain information interactively about variables.

Access: Main Menu > Query > Interactive probe...

(see Section 7.6, Interactive Probe Query and How To Probe Interactively)

Dataset... Opens the Query Dataset dialog which is used to obtain information about datasets for the selected case. This can be useful to verify the files that you are using, whether the geometry is static, changing coordinates, or changing connectivity, and the number of each element type in your dataset.

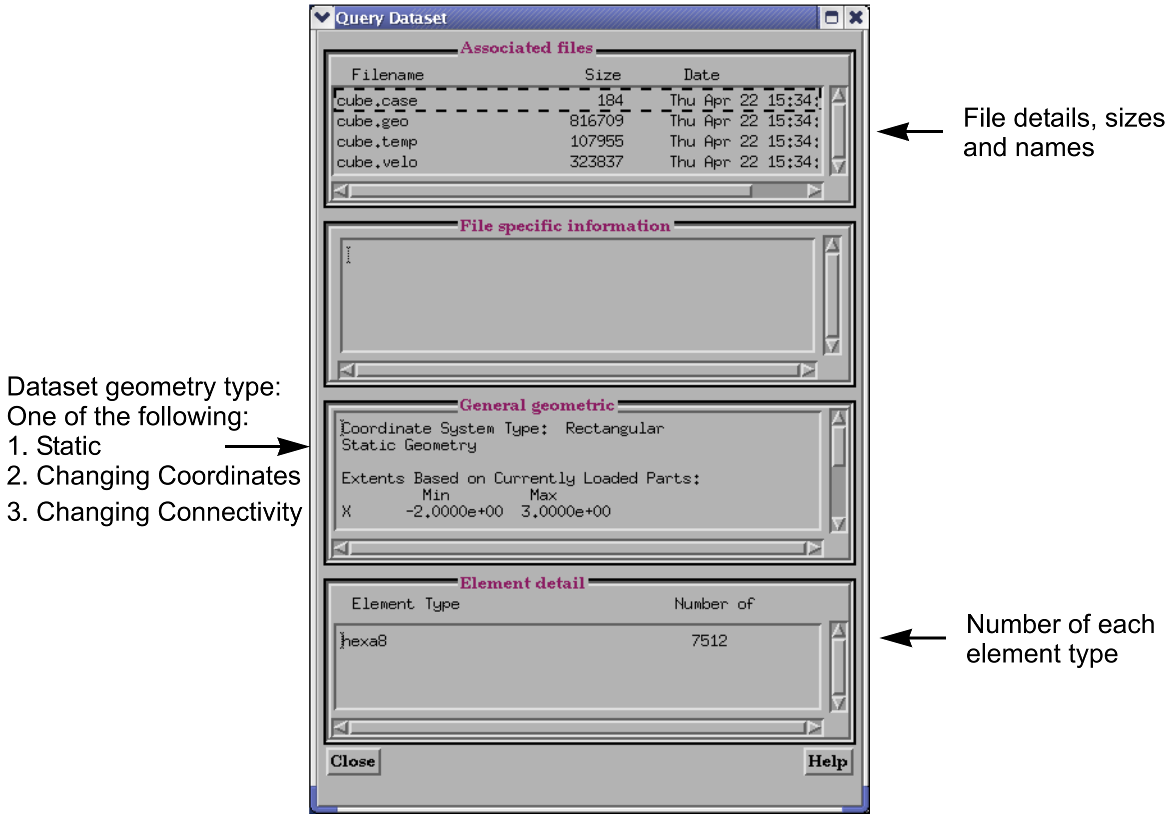


Figure 6-11
Query Dataset dialog

For the specified file, specific, general and detail information is provided.

Access: Main Menu > Query > Dataset...

(see Section 7.5, Query/Plot and How To Query Datasets)

6.4 View Menu Functions

Clicking the View button in the Main Menu opens a pull-down menu which provides access to the following features many of which can be accessed by the global toggles just above the graphics screen as indicated below:

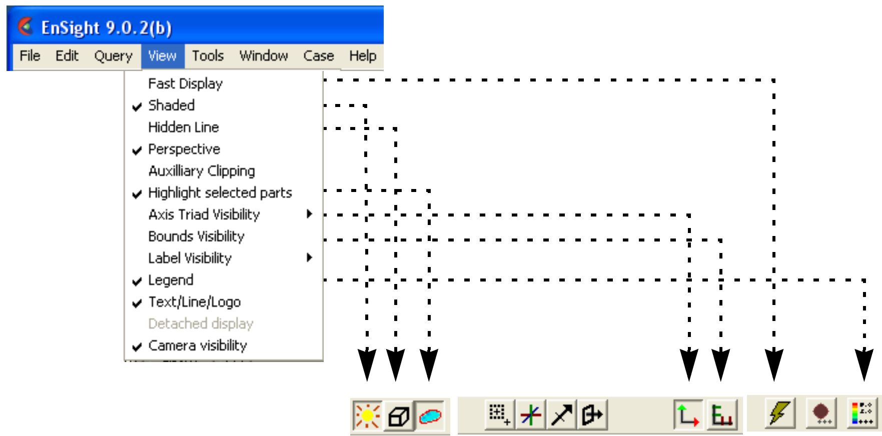


Figure 6-12
View pull-down menu

Fast display

Toggles the Fast display mode.
Access: Main Menu > View > Fast display

Fast Display in this pull-down is the same as the one located on the Desktop.

By default, EnSight displays all of the lines and elements for each part every time the Main View window redraws. If you have very large models (or if you have slow graphics hardware), each redraw can take significant time. As a result, interactive transformations become jerky and lag behind the motion of the mouse. Ironically, the slower the graphics performance, the harder it is to perform precise interactive transformations. To avoid this problem, you can tell EnSight to show a lesser detailed part representation, i.e, a bounding box surrounding each Part, or the Part as a point cloud. You can select to show the detail representation all the time, or only while you are performing transformations. This obviously displays much less information, but may be sufficient if you want to rotate a very large model.

A lesser detail display is also useful when experimenting with keyframe-animation rates. Using lesser detail, the display rate can be adjusted to approximate the video rate, thus you can see how your scene will transform on the video tape

The default setting is off, indicating that all lines and elements of all visible parts will be redrawn. When on, the redraw will show only the part's Fast Display Representation (by default a box). The fast display representation is only used while transformations are being performed. The fast display representation will be continuously displayed if the Static Fast Display option is turned on in:
Main Menu > Edit > Preferences > Performance > Static fast display.

Shaded Toggle

Toggles the *Global* Shaded mode for parts on and off. EnSight by default displays parts in line mode. Shaded mode displays parts in a more realistic manner by making hidden surfaces invisible while shading visible surfaces according to specified lighting parameters. Parts in Shaded mode require more time to redraw than when in line mode, so you may wish to first set up the Graphics Window as you want it, then turn on Shaded to see the final result. Shaded can be toggled on/off for individual parts by using the Shaded Toggle icon in the Part Mode Icon Bar.
Access: EnSight dialog > View > Shaded
or Desktop > Shaded

(see Section 8.6, Quick Desktop Buttons and How To Set Drawing Style)

Troubleshooting Hidden Surfaces and Shading

Problem	Probable Causes	Solutions
Main View shows line drawing after turning on Shaded.	Shaded is toggled off for each individual part.	Toggle Shaded on for individual parts with the Shaded Icon in Part Mode or in the Feature Detail Editor dialog.
	There are no surfaces to shade—all parts have only lines.	If parts are currently in Feature Angle representation, change the representation. If model only has lines, you can not display shaded images.
	The element visibility attributes has been toggled off for the part(s).	Toggle the element visibility on for individual parts in the Feature Detail Editor dialog.

Hidden line

Toggles the global Hidden line display for all parts on/off. This simplifies a line drawing display by making hidden lines - lines behind surfaces - invisible while continuing to display other lines. Hidden Line can be combined with Shaded to display both surfaces and the edges of the visible surface elements. Hidden Line can be toggled on/off for individual parts by using the Hidden Line Toggle icon in the Part Mode Icon Bar.
To have lines hidden behind surfaces, you must have surfaces (2D elements). If the representation of the in-front parts consists of 1D elements, the display is the same whether or not you have Hidden Lines mode toggled on.

Access: Main Menu > View > Hidden line

(see Section 8.6, Quick Desktop Buttons and How To Set Drawing Style)

Hidden line overlay dialog

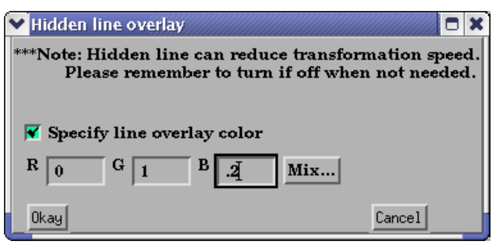


Figure 6-13
Hidden line overlay dialog

If you combine Shaded mode with Hidden line mode, the lines overlay the surfaces. The Hidden line overlay dialog will pop-up on the screen if the Shaded option is currently on and you then turn the Hidden Line option on. From this dialog you specify a color for the displayed lines (you do not want to use the same color as the surfaces since they then will be indistinguishable from the surfaces). The default is the part-color of each part, which may be appropriate if the surfaces are colored by a color palette instead of their part-color.

Specify line overlay color
: Toggle-on if you want to specify an overlay color. If off, the overlay line color will be the same as the part color.

R, G, B
: The red, green, and blue components of the hidden line overlay. These fields will not be accessible unless the Specify Overlay option is on.

Mix...
: Click to interactively specify the constant color used for the hidden line overlay using the

Color selector dialog.
(see Section 8.1, Color Selector) and How To Change Color)

Okay — Click to accept the hidden line overlay color options.

Perspective (Global)Toggle — Toggles the view within each of the viewports within the Graphics Window between a perspective view (the default) and an orthographic projection. *Perspective* is what gives you the sense of depth when viewing a three dimensional scene on a two dimensional surface. Objects that are far away look smaller and parallel lines seem to meet at infinity. *Orthographic projection* removes the sense of depth in a scene. Lines that are parallel will never meet and objects of the same size all appear the same no matter how far away they are from you. Orthographic projection mode often helps when you are positioning the Cursor, Line, and Plane tools using multiple viewports. This is the Global toggle. Each viewport also has a Perspective Toggle.
Access: Main Menu > View > Perspective

(see Section 8.4, VPort Mode and see How To Set Global Viewing)

Auxiliary clipping — Toggles the Auxiliary clipping feature on/off. (Default is Off). Like a Z-Clip plane, Auxiliary clipping cuts-away a portion of the model. When Auxiliary clipping is On, Parts (or portions of Parts) located on the back (negative-Z) side of the Plane Tool are removed. Parts whose Clip attribute you have toggled off (in the General Attributes section of the Feature Detail Editor dialog or with the Auxiliary clipping Toggle Icon in the Part Mode Icon Bar) remain unaffected.

Auxiliary clipping is interactive—the view updates in real time as you move the Plane Tool around

(see Section 6.5, Tools Menu Functions and How To Use the Plane Tool).

Unlike a Z-Clip plane, Auxiliary clipping applies only to the parts you specify, and the plane can be located anywhere with any orientation though it is always infinite in extent (see Section 9.6, Z-Clip and How To Set Z Clipping).

Auxiliary clipping is helpful, for example, with internal flow problems since you can “peel” off the outside parts and look inside. This capability is also often useful in animation.

The position of the Plane Tool and the status of Auxiliary clipping is the same for all displayed viewports.

Do not confuse Auxiliary clipping with a 2D-Clip plane, which is a created part whose geometry lies in a plane cutting through its parent parts or with the Part operation of cutting a part.

(see Section 3.4, Part Operations, How to Create Plane Clips, and How To Cut a Part).

Troubleshooting Auxiliary Clipping

Problem	Probable Causes	Solutions
The Plane Tool does not appear to clip anything	The Auxiliary Clipping toggle is off for all parts.	Turn the Auxiliary Clipping toggle on for individual parts in the Feature Detail Editor (Model) dialog General Attributes section.
	The Plane Tool is not intersecting the model	Change the position of the Plane Tool.
The Main View window shows nothing other than the Plane Tool after Clipping is toggled-on.	All of the part(s) is(are) on the back side of the Plane Tool and is(are) thus clipped	Change the position of the Plane Tool.

Hightlight selected part(s)...Highlight the selected parts in the graphics window. This often aids in the identification of parts.

Axis triad visibility Opens the pull-down menu which allows you to toggle on/off the visibility of the Global axis triad, the axis triads for all Frames, and the model axis triad.

Frame Toggle Toggles on/off (default is On) the display of all coordinate Frame axis triads. The visibility of individual coordinate Frame axes can be selectively turned on/off by clicking on the Frame's axis triad and then clicking on the Frame Axis Triad Visibility Toggle in the Frame Mode Icon Bar.
Access: Main Menu > View > Axis triad visibility > Frame
(see Section 8.5, Frame Mode)

Global Toggle Toggles on/off (default is Off) the display of the global coordinate frame axis. The global coordinate frame axis triad represents the Look-At Point.
Access: Main Menu > View > Axis triad visibility > Global

Model Toggle Toggles on/off the display of the model axis triad in the lower left of the screen. This triad is not at the origin of frame 0, but is aligned with it.
Access: Main Menu > View > Axis triad visibility > Model

(see Section 8.6, Quick Desktop Buttons)

Bounds visibility Toggles on/off (default is Off) the extents box for all parts.

Label visibility Opens the pull-down menu which allows you to toggle on/off the visibility of labels for Elements or Nodes.

Element labeling Toggle Toggles on/off (default is Off) the global visibility of labels (if they are available in the dataset) for elements in all parts. Visibility of element labels for individual parts can be controlled via the Node/Element labeling icon of Part Mode.
Access: Main Menu > View > Label Visibility > Element labeling

Node labeling Toggle Toggles on/off (default is off) the global visibility of labels (if they are available in the dataset) for nodes in all parts. Visibility of node labels for individual parts can be controlled via the Node/Element labeling icon of Part Mode.
Access: Main Menu > View > Label Visibility > Node labeling

Labeling attributes... Opens the Node/Element labeling dialog from which you can control the labeling visibility, color, and filtering for selected parts. This same dialog can also be reached from the Node/Element labeling icon in Part Mode.
Access:Main Menu > View > Label Visibility > Labelling attributes...

Legend Toggle Toggles on/off (default is on) the global visibility of all legends. Visibility of individual legends can be controlled by using the Edit legends button in Annot Mode.
Access:Main Menu > View > Legend

(see Section 4.2, Variable Summary & Palette, Section 8.2, Annot Mode and How To Create Color Legends.

Text/Line/Logo Toggle Toggles on/off global visibility for text strings and lines which have been created and logos which have been imported. Visibility of individual Text strings, Lines, or Logos can be controlled by selecting the item while in Annotation Mode and clicking the Visibility Toggle in the Annotation dialog. While in Annot Mode, you will notice that the item does not disappear, but turns transparent. Such items will not appear in the Graphics Window in any Mode except Annotation Mode, and then only if global visibility has been turned on.
Access: Main Menu > View > Text/Line/Logo

(see Section 8.2, Annot Mode, How To Create Lines and Arrows, How To Create Text Annotation, and How To Load Custom Logos.

Detached display This menu item is disabled unless using a detached display. If you have a detached display then it's automatically enabled and toggled on, and this menu item allows toggling on/off this detached display.

Camera Visibility

Toggles on/off the global visibility for cameras. Visiblity of individual cameras can be controlled by selecting the camera in the Transformation editor (Camera) dialog.
(see Chapter 9.9, Camera)

6.5 Tools Menu Functions

The Region selector, Cursor, Line, Plane, Box, and Quadric (cylinder, sphere, cone, and revolution) Tools in EnSight are used for a variety of tasks, such as: positioning of clipping planes and lines, query operations, particle trace emitters, etc. Collectively these tools are referred to as Positioning Tools. Clicking the Tools button in the Main Menu opens a pull-down menu which provides access to these features some of which are also available as global toggles above the graphics window as indicated below:

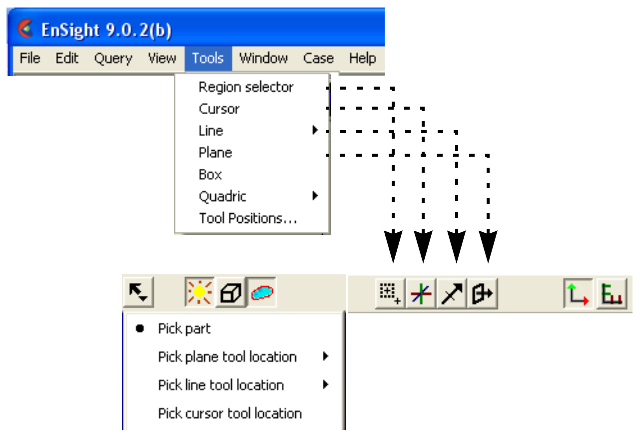


Figure 6-14
Tools pull-down menu and Pick Part pull-down menu

Region selector Toggle

Makes the Selection Tool (region selector) visible/invisible in the Graphics Window. The Selection Tool appears as a white rectangle with two symbols at the upper left of the tool (one is for zooming and the other for element blanking). It may be repositioned interactively in the Graphics Window by selecting and dragging it or by selecting any corner and rubber banding the corner. Note that a dotted rectangle, which stays at the same aspect ratio of the screen, will indicate the actual selection area as it is manipulated. Alternatively, you can reposition it precisely by specifying X and Y min/max coordinates in the Transformation Editor dialog (described in Tool Positions...Region Tool below).

Access: Main Menu > Tools > Region selector
or Desktop > Selection tool

(see Section 8.1, Part Mode and How to Use the Selection Tool)

Cursor Tool Toggle

Makes the Cursor Tool visible/invisible in the Graphics Window. The Cursor Tool appears as a three-dimensional cross colored red, green, and blue. The red axis of the cross corresponds to the X axis direction for the currently selected Frame, while green matches the Y and blue matches up with the Z. The Cursor Tool is initially located at the Look-At point and may be repositioned interactively in the Graphics Window by left-clicking and dragging it or right-clicking on it and choosing 'Edit' on the pulldown which opens the Tranformation Editor. If you have a specific location you want to click on and have the cursor moved there, then select Pick Cursor Location from the Pick Pull-down Icon menu in the Quick Desktop Button region.
(see Section 8.1, Part Mode and How to Use the Cursor (Point) Tool)

Line Tool Toggle

Makes the Line Tool visible/invisible in the Graphics Window. The Line Tool appears as a white line with a dotted-line axis system at the center point and an arrowhead on its 2nd endpoint. The Line Tool is initially centered about the Look-At point and sized so that it fills approximately 10% of the default view. There are a number of ways to manipulate the tools either interactively or via the transformation dialog. You can change its length and orientation interactively in the Graphics Window by selecting one of its end points. You

can rotate the line tool by clicking and dragging on the center axes. You can reposition it interactively in the Graphics Window by selecting its center and dragging it or by selecting Pick Line Location from the Pick Pulldown Icon menu in the Quick Desktop Button region. Alternatively, you can reposition it precisely by rotating, translating, or specifying coordinates in the Transformation Editor dialog by right-clicking on the line tool and selecting 'Edit'. If you have a precise location that you want to locate the line tool you can select Pick Line Tool Location from the Pick Pull-down Icon menu in the Quick Desktop Button region.
(see Section 8.1, Part Mode and How to Use the Line Tool)

Plane Tool

Makes the Plane Tool visible/invisible in the Graphics Window. (*Note: Its appearance (line or filled) is controlled under Main Menu > Edit > Preferences > View*)
The Plane Tool is shown with an X, Y, Z axis system, is initially centered about the Look-At point, and lies in the X-Y plane. You can reposition it interactively in the Graphics Window by selecting its center point in the Graphics Window and dragging it. You can change its orientation interactively in the Graphics Window by selecting the X, Y, or Z letters at the ends of the axes. You can resize the Plane Tool interactively in the Graphics Window by selecting any corner of the plane and dragging it. You can rubber-band any of the corners by holding the Ctrl key while selecting and dragging. You can reposition it precisely using the Tranformation Editor by right clicking on the plane tool center point and choosing 'Edit'. If you have a precise location that you want to locate the plane tool, you can choose Pick Plane Tool Location from the Pick Pull-down Icon menu in the Quick Desktop Button region.
(see Section 8.1, Part Mode and How to Use the Plane Tool)

Box Tool

Makes the Box Tool visible/invisible in the Graphics Window. The Box Tool is shown with an X, Y, Z axis system and is initially centered about the Look-At point. You can resize it interactively in the Graphics Window by selecting any of its corner points and dragging. You can reposition it interactively in the graphics window by selecting the origin of the box and dragging. You can reposition it precisely using the Transformation Editor by right-clicking on the box tool origin and choosing 'Edit'. You can perform these types of operations as well as rotations, in the Transformation Editor dialog (described in Tool Positions... Box Mode below). You can even reposition it precisely by specifying coordinates in the Transformation Editor dialog.
(see Section 8.1, Part Mode and How to Use the Box Tool)

Quadric

Opens a pull-down menu which allows you to choose one of the Quadric Tools and make it visible.

Access: Main Menu > Tools > Quadric

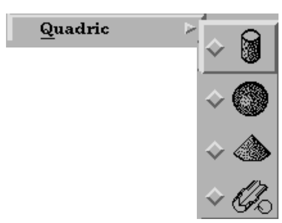


Figure 6-15
Quadric Tool pull-down menu

Cylinder Tool Toggle

Makes the Cylinder Tool visible/invisible in the Graphics Window. The Cylinder Tool appears as thick direction line with center point and center tool axis system, and a circle around the line at the mid and two end points. Thinner projection lines run parallel to the direction line through the three circles outlining the surface of the cylinder. The Cylinder Tool is initially centered about the Look-At point with the direction line pointing in the X direction. There are a number of ways to manipulate the tools either interactively or via the transformation dialog. You can change its length and orientation interactively in the Graphics Window by selecting one of its end points. You can rotate it in the Graphics Window by selecting the end of one of the tool axes. You can change its diameter by

selecting the circle about the mid point. You can reposition it interactively in the Graphics Window by selecting its center. You can reposition it precisely using the Transformation Editor by right-clicking on the cylinder tool origin and choosing 'Edit'.
(see How to Use the Cylinder Tool)

Sphere Tool Toggle	Makes the Sphere Tool visible/invisible in the Graphics Window. The Sphere Tool appears as thick direction line with center point and center tool axis system, and with several circles outlining the sphere. The Sphere Tool is initially centered about the Look-At point with the direction line pointing in the X direction. There are a number of ways to manipulate the tools either interactively or via the transformation dialog. You can change its radius and orientation interactively in the Graphics Window by selecting one of the thick direction line end points. You can rotate it in the Graphics Window by selecting the end of one of the tool axes. You can reposition it interactively in the Graphics Window by selecting its center. You can reposition it precisely using the Transformation Editor by right-clicking on the sphere tool origin and choosing 'Edit'. (see How to Use the Sphere Tool)
Cone Tool Toggle	Makes the Cone Tool visible/invisible in the Graphics Window. The Cone Tool appears as thick direction line with center point and a tool axis system at the apex. It has a circle at the opposite end point. Thinner projection lines run from the beginning point to the circle at the end point outlining the surface of the cone. The Cone Tool is initially centered about the Look-At point with the direction line pointing in the X direction. There are a number of ways to manipulate the tools either interactively or via the transformation dialog. You can change its length and orientation interactively in the Graphics Window by selecting one of the thick direction line end points. You can change its diameter by selecting the largest circle about the end point. You can rotate it in the Graphics Window by selecting the end of one of the tool axes. You can reposition it interactively in the Graphics Window by selecting its center. You can reposition it precisely using the Transformation Editor by right-clicking on the cone tool origin and choosing 'Edit'. Note: the cone tool always operates as if the tool extends infinitely from the origin at the half angle. The half angle of the cone tool is in degrees. (see How to Use the Cone Tool)
Revolution Tool Toggle	Makes the Surface of Revolution Tool visible/invisible in the Graphics Window. The Revolution Tool appears as thick direction line with center point and center tool axis system, and with several circles outlining each user defined point along the tool. Thinner projection lines run through the circles to outline the revolution surface. The Revolution Tool is initially centered about the Look-At point with the direction line pointing in the X direction. There are a number of ways to manipulate the tools either interactively or via the transformation dialog. You can change its length and orientation interactively in the Graphics Window by selecting one of the thick direction line end points. You can rotate it in the Graphics Window by selecting the end of one of the tool axes. You can reposition it interactively in the Graphics Window by selecting its center or alternatively, you can reposition it precisely by specifying coordinates in the Transformation Editor dialog (described in Tool Positions... Quadric below). Access: Main Menu > Tools > Quadric (see How to Use the Surface of Revolution Tool)

Tool positions...

Opens the Transformation Editor dialog which allows you to precisely position the various tools within the Graphics Window in reference to the selected Frame.

Access: Main Menu > Tools > Tool Positions...

Region Tool (Selection region)

Click on Editor Function in the Transformation Editor dialog and then selecting Tools > Select region from the pull-down menu configures the dialog as shown below:

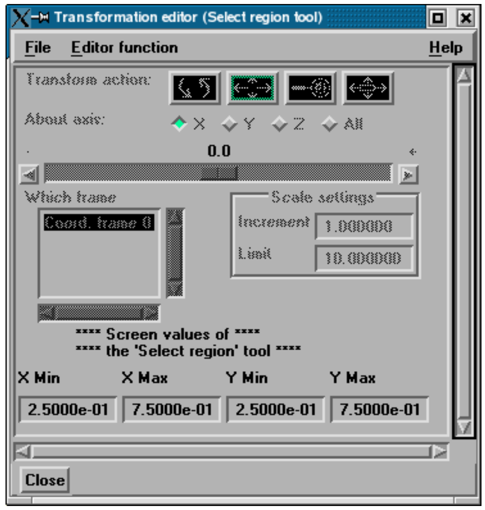


Figure 6-16
Transformation Editor (Select region)

To precisely position the Selection tool, enter the desired normalized screen coordinate values for X and Y minimum and maximum. The coordinates can be between 0.0 and 1.0.

Access: Main Menu > Tools > Tool Positions... > Tools > Select region

Cursor Tool

Clicking on Editor Function in the Transformation Editor dialog and then selecting Tools > Cursor from the pull-down menu configures the dialog as shown below. Even easier and quicker to get this dialog is to right-click on the cursor tool and choose 'Edit'.

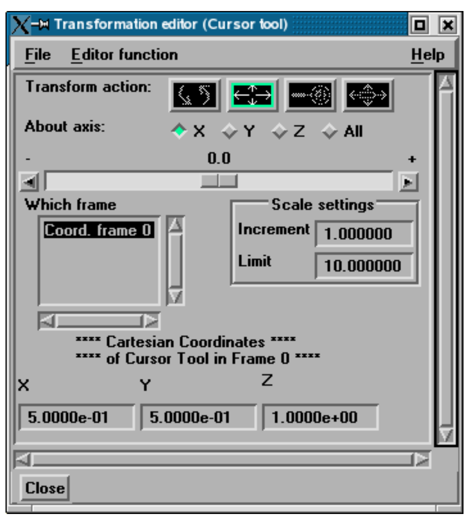


Figure 6-17
Transformation Editor (Cursor)

The Transformation Editor dialog provides three methods for the precise positioning of the Cursor Tool. First, the Cursor Tool may be positioned within the Graphics Window by entering coordinates in the X, Y, and Z fields. Pressing return causes the Cursor Tool to relocate to the specified coordinates in the selected Frame (or, if more than one Frame is selected, for Frame 0).

It is also possible to reposition the Cursor Tool from its present coordinate position by specific increments. The Axis Button allows you to choose the axis of translation (X, Y, Z, or All). The Slider Bar at Top allows you to quickly choose the increment by which to move the position of the Cursor Tool. Dragging the slider in the negative (left) or positive (right) directions and then releasing it will cause the X, Y, and Z coordinate fields to increment as specified and the Cursor Tool to relocate to the new coordinates. The number specified in the Limit field of the Scale Settings area determines the negative (-) and positive (+) range of the slider. If the Limit is set to 1.0 as shown, then the numerical range of the slider bar will be -1 to +1.

Alternatively, you can specify an increment for translation in the Increment field of the Scale Settings area. Pressing return while the mouse pointer is in the Increment field will cause the Cursor Tool to translate along the specified axis (or all axes) by the increment specified.

Access: Transformation Editor > Editor Function > Tools > Cursor

(see How to Use the Cursor (Point) Tool)

Line Tool

From the menu Tools>Line there are three options.

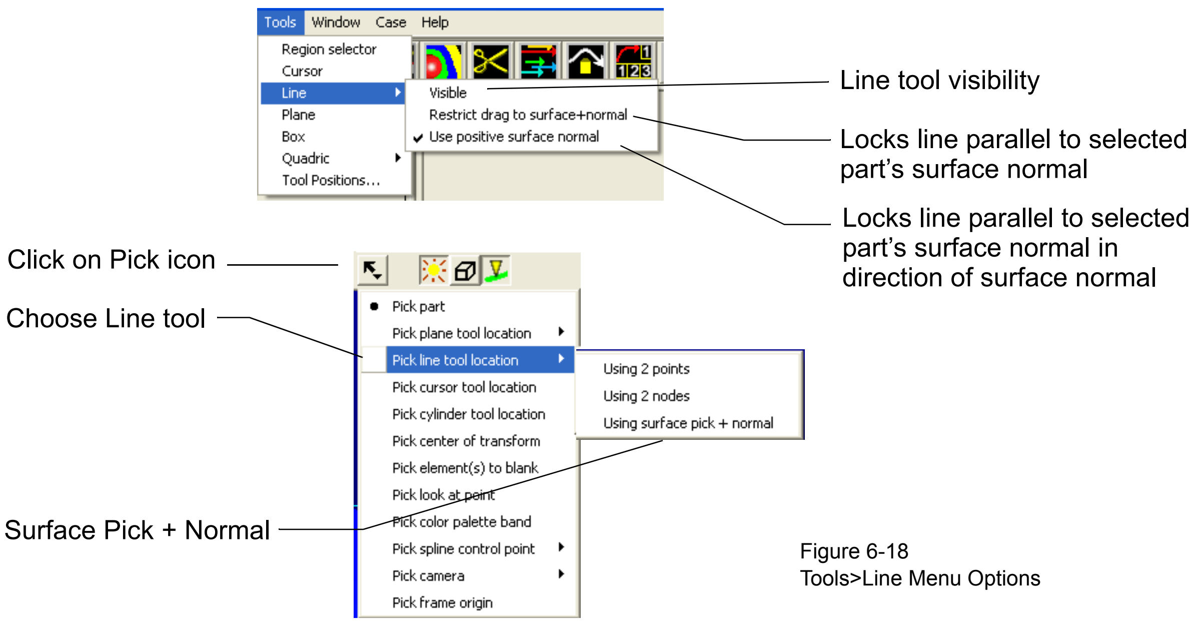


Figure 6-18
Tools>Line Menu Options

Visible

Toggles the line tool visibility

Restrict drag to surface + normal

After choosing this option, click on the Pick icon and choose to pick plane tool location using surface pick + normal and then put the cursor tool on the part surface and pick using the 'p' key and the line tool will appear aligned with the surface normal at the point picked with one end attached to the surface at the point of the picking.

Use positive surface normal

Same as above except the line tool will originate at the picked point location, but will be in the direction of the positive normal.

Another way that the line tool can be positioned is using the Transformation Editor. Click on the Transf button at the bottom of the graphics window. Then click on Editor Function in the Transformation Editor dialog and then select Tools > Line from the pull-down menu to see the dialog as shown below. Even easier to get this dialog is to right-click on the visible line tool in the graphics area and choose ‘Edit’.

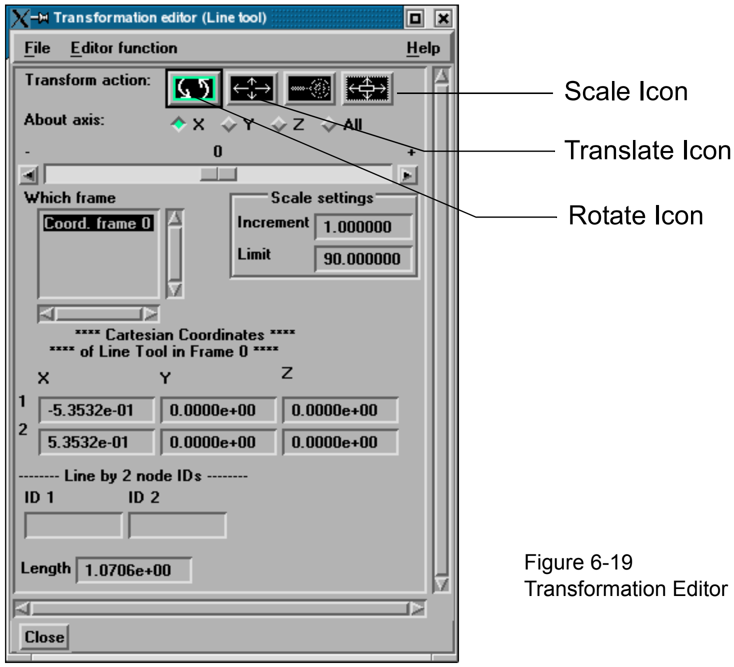


Figure 6-19
Transformation Editor (Line Tool)

The Transformation Editor can control precisely the position and size of the line tool.

Position The Transformation Editor dialog provides several methods for the precise positioning of the Line Tool. First, the Line Tool may be positioned within the Graphics Window by entering coordinates for the two endpoints in the X, Y, and Z fields. Pressing return after entering each coordinate value causes the Line Tool to relocate to the specified coordinates in the selected Frame (or if more than one Frame is selected, in Frame 0). Enter all three X, Y, Z fields for a endpoint and then press return once to cause the line tool to update it’s position to that endpoint.

You can also specify the node ID labels to use for the line endpoints.

It is also possible to reposition the Line Tool from its present coordinate position by specific increments. First click on the translate icon. The Axis Button allows you to choose the axis of translation for the center of the line (X, Y, Z, or All). The Slider Bar at Top allows you to quickly choose the increment by which to move the position of the center point of the Line Tool. Dragging the slider in the negative (left) or positive (right) directions and then releasing it will cause the X, Y, and Z coordinate fields to increment as specified and the Line Tool to relocate to the new coordinates. The number specified in the Limit field of the Scale Settings area determines the negative (-) and positive (+) range of the slider. If the Limit is set to 1.0 as shown, then the numerical range of the slider bar will be -1 to +1. The transformations are relative to the line tool axis system.

Alternatively, you can specify an increment for translation in the Increment field of the Scale Settings area. Pressing return while the mouse pointer is in the Increment field will cause the center point of the Line Tool to translate along the specified axis (or all axes) by the increment specified.

Orientation First click on the rotate icon. Next, pick an axis about which to rotate. Next pick an increment and limit (in degrees) and slide the slider to rotate the plane.

Scale First click on the scale icon. Next pick an increment and limit and slide the slider to scale the line about its center, along its length.

Access: Transformation Editor > Editor Function > Tools > Line

(see How to Use the Line Tool)

Plane Tool

Clicking on Editor Function in the Transformation Editor dialog and then selecting Tools > Plane from the pull-down menu configures the dialog as shown below. Even easier and quicker to get to this dialog is to right-click on the visible plane tool and choose 'Edit'.

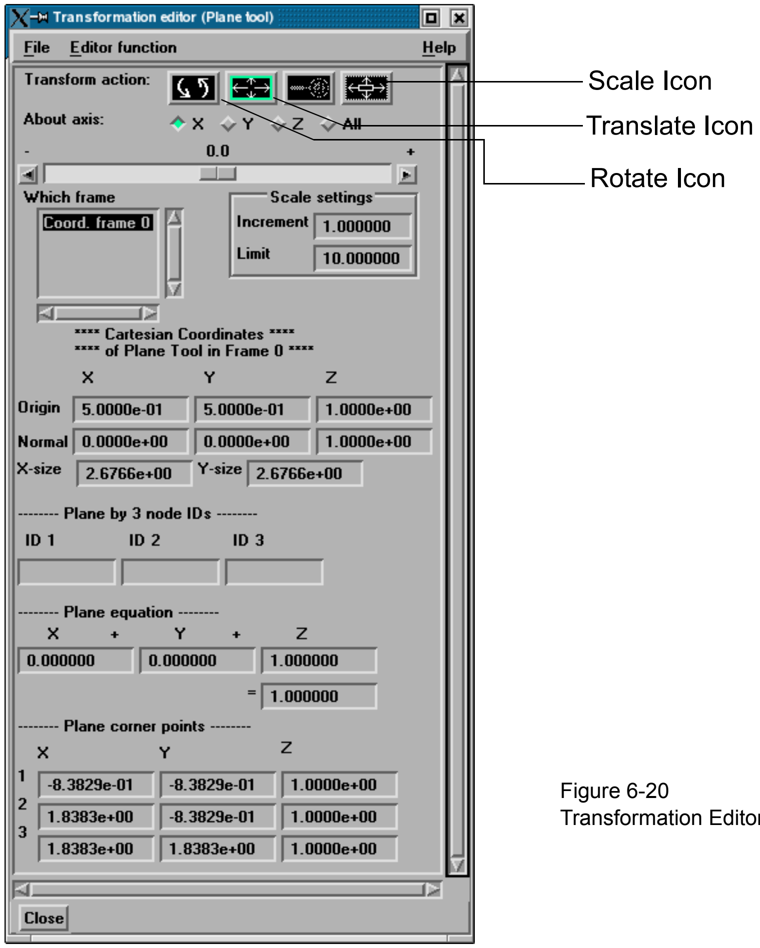


Figure 6-20
Transformation Editor (Plane Tool)

The Transformation Editor can control precisely the position, orientation, and size of the plane tool.

Position

The Transformation Editor dialog provides several methods for the precise positioning of the Plane Tool. First, the Plane Tool may be positioned within the Graphics Window by entering coordinates for the three corners of the plane in the X, Y, and Z fields. Corner 1 is defined as the -X, -Y corner of the plane, Corner 2 is defined as the +X, -Y corner of the plane, and Corner 3 is defined as the +X, +Y corner of the plane. Pressing return causes the Plane Tool to relocate to the specified coordinates in the selected Frame (or if more than one Frame is selected, in Frame 0). For your convenience, you can enter values into all 9 fields and then press return once to update the plane tool position.

You can also position the Plane Tool be entering the id for three nodes. The Plane Tool will then remain tied to these three nodes - even as the nodes move in a transient geometry model.

You can also position the Plane Tool by entering a plane equation in the form $A_x + B_y + C_z = D$ in the four fields and then pressing Return. For convenience, enter in all four then press return. The coefficients may then be normalized, but the equation of the plane will be the same as the one you entered. The coefficients of the plane equation are in reference to the selected Frame (or if more than one Frame is selected, to Frame 0).

As with the Cursor and Line Tools, it is possible to reposition the Plane Tool from its present coordinate position by specific increments. First click the translate icon at the top of the Transformation Editor. The Axis Button allows you to choose the axis of translation (X, Y, Z, or All) for the origin of the Plane Tool (intersection of the axes). The Slider Bar at Top allows you to quickly choose the increment by which to move the position of the origin. Dragging the slider in the negative (left) or positive (right) directions and then releasing it will cause the X, Y, and Z coordinate fields to increment as specified and the origin of the Plane Tool to relocate to the new coordinates. The number specified in the Limit field of the Scale Settings area determines the negative (-) and positive (+) range of the slider. If the Limit is set to 1.0 as shown, then the numerical range of the slider bar will be -1 to +1.

Alternatively, you can specify an increment for translation in the Increment field of the Scale Settings area. Pressing return while the mouse pointer is in the Increment field will cause the center of the Plane Tool to translate along the specified axis (or all axes) by the increment specified.

Orientation First click on the rotate icon. Next, pick an axis about which to rotate. Next pick an increment and limit (in degrees) and slide the slider to rotate the plane.

Scale First click on the scale icon. Next pick an axis direction to scale (X or Y only). Finally pick an increment and limit and slide the slider to scale the size of the plane.

Access: Transformation Editor > Editor Function > Tools > Plane

(see How to Use the Plane Tool)

Box Tool

Clicking on Editor Function in the Transformation Editor dialog and then selecting Tools > Box from the pull-down menu configures the dialog as shown below. Even easier and quicker to get to this dialog is to right-click on the visible box tool and choose 'Edit'.

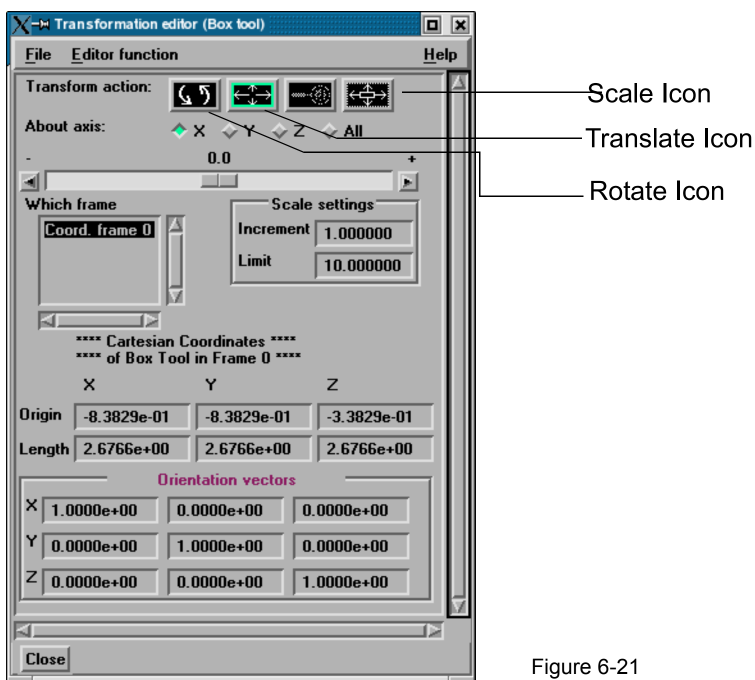


Figure 6-21
Transformation Editor (Box Tool)

The Transformation Editor can control precisely the position, orientation, and size of the box tool.

Position The Transformation Editor dialog provides several methods for the precise positioning of the Box Tool. First, the Box Tool may be positioned within the Graphics Window by entering coordinates for the origin of the box in the X, Y, and Z fields and the length of the each of the X, Y, and Z sides. Pressing return causes the Box Tool to relocate to the specified location in the selected Frame (or if more than one Frame is selected, in Frame 0). For your convenience, you can enter in all of the fields and then press return once to update the Box Tool position.

Additionally, you can modify the orientation of the Box Tool by entering the X, Y, and Z orientation vectors of the box axis in regards to Frame 0.

As with other Tools, it is possible to reposition the Box Tool from its present coordinate position by specific increments. First click the translate icon at the top of the Transformation Editor. The Axis Button allows you to choose the axis of translation (X, Y, Z, or All) for the origin of the Box Tool (intersection of the axes). The Slider Bar at Top allows you to quickly choose the increment by which to move the position of the origin. Dragging the slider in the negative (left) or positive (right) directions and then releasing it will cause the X, Y, and Z coordinate fields to increment as specified and the origin of the Box Tool to relocate to the new coordinates. The number specified in the Limit field of the Scale Settings area determines the negative (-) and positive (+) range of the slider. If the Limit is set to 1.0 as shown, then the numerical range of the slider bar will be -1 to +1.

Alternatively, you can specify an increment for translation in the Increment field of the Scale Settings area. Pressing return while the mouse pointer is in the Increment field will cause the origin of the Box Tool to translate along the specified axis (or all axes) by the increment specified.

Orientation First click on the rotate icon. Next, pick an axis about which to rotate. Next pick an increment and limit (in degrees) and slide the slider to rotate the Box Tool.

Scale First click on the scale icon. Next pick an axis direction to scale. Finally pick an increment and limit and slide the slider to scale the size of the Box Tool.

Access: Transformation Editor > Editor Function > Tools > Box

(see How to Use the Box Tool)

Cylinder or Sphere Tools

Clicking on Editor Function in the Transformation Editor dialog and then selecting Tools and then Cylinder or Sphere from the pull-down menu configures the dialog as shown below. Even easier and quicker to get to this dialog is to right-click on the visible tool and choose 'Edit'.

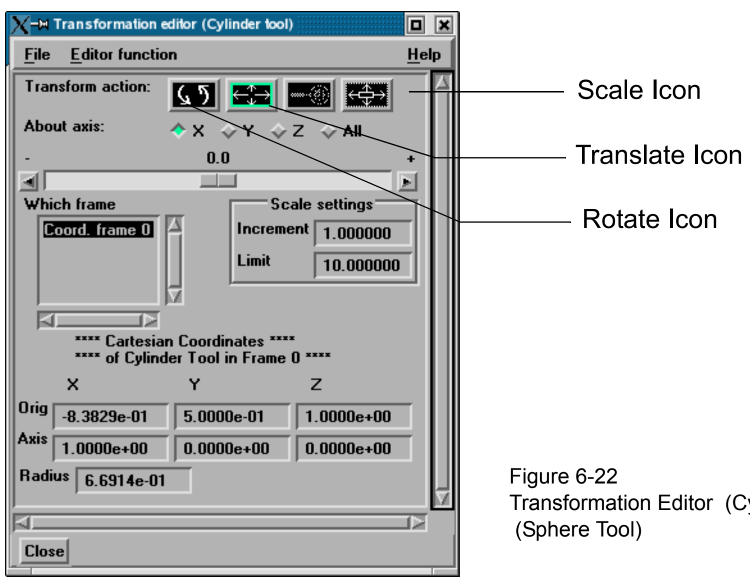


Figure 6-22
Transformation Editor (Cylinder Tool) or (Sphere Tool)

The Transformation Editor can control precisely the position and size of the cylinder tool.

Position The Transformation Editor dialog enables you to precisely control the coordinates of the Cylinder or Sphere Tool origin (center point of the thick direction line) by specifying them in the Orig. X, Y, and Z fields. You control the direction vector for the Cylinder or Sphere Tool direction axes by specifying the coordinates in the Axis X, Y, and Z fields of the selected Frame (or if more than one Frame is selected, in Frame 0). The Radius of each tool may be specified in the Radius Field.

It is possible to reposition the Cylinder or Sphere Tool origins by specific increments. First click on the translate icon. The Axis Button allows you to choose the axis of translation (X, Y, Z, or All) for the origin of the tool. The Slider Bar at Top allows you to quickly choose the increment by which to move the position of the origin. Dragging the slider it in the negative (left) or positive (right) directions and then releasing it will cause the X, Y, and Z coordinate fields to increment as specified and the origin of the Cylinder or Sphere Tool to relocate to the new coordinates. The number specified in the Limit field of the Scale Settings area determines the negative (-) and positive (+) range of the slider. If the Limit is set to 1.0 as shown, then the numerical range of the slider bar will be -1 to +1.

Alternatively, you can specify an increment for translation in the Increment field of the Scale Settings area. Pressing return while the mouse pointer is in the Increment field will cause the origin of the Cylinder or Sphere Tool to translate along the specified axis (or all axes) by the increment specified.

Orientation First click on the rotate icon. Next, pick an axis about which to rotate. Next pick an increment and limit (in degrees) and slide the slider to rotate the plane.

Scale First click on the scale icon. Next pick an axis direction to scale. Can only scale in the X (longitudinal) or Y (radial) directions. Finally pick an increment and limit and slide the slider to scale the size of the cylinder or sphere Tool.

Access: Transformation Editor > Editor Function > Tools > Cylinder or Sphere

(see How To Use the Cylinder Tool and How To use the Sphere Tool)

Cone Tool

Clicking on Editor Function in the Transformation Editor dialog and then selecting Tools and then Cone from the pull-down menus configures the dialog as shown below.

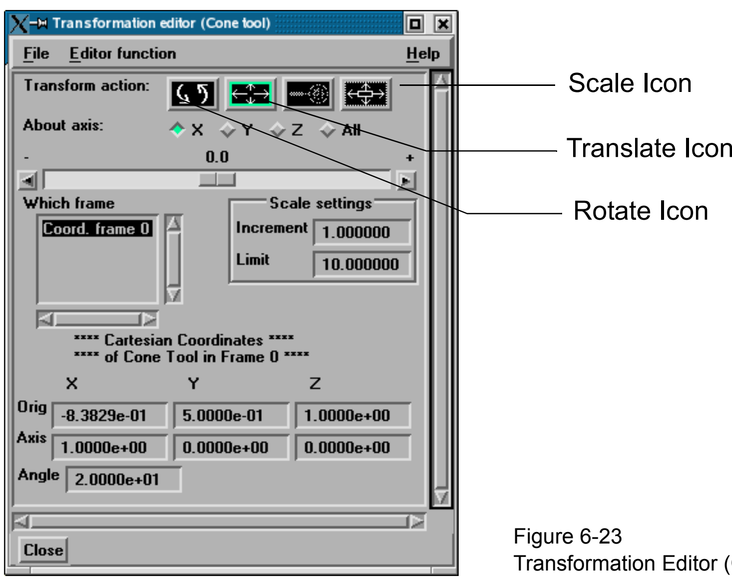


Figure 6-23
Transformation Editor (Cone Tool)

The Transformation Editor dialog enables you to precisely control the coordinates of the Cone Tool origin (the point of the cone) by specifying them in the Orig. X, Y, and Z fields. You control the direction vector for the Cone Tool direction axis by specifying the coordinates in the Axis X, Y, and Z fields for the selected Frame (or if more than one Frame is selected, in Frame 0). The conical half angle may be specified in degrees in the Angle Field.

Position It is possible to reposition the Cone Tool origin by specific increments. The Axis Button allows you to choose the axis of translation (X, Y, Z, or All) for the origin of the tool. The Slider Bar at Top allows you to quickly choose the increment by which to move the position of the origin. Dragging the slider in the negative (left) or positive (right) directions and then releasing it will cause the X, Y, and Z coordinate fields to increment as specified and the origin of the Cone Tool to relocate to the new coordinates. The number specified in the Limit field of the Scale Settings area determines the negative (-) and positive (+) range of the slider. If the Limit is set to 1.0 as shown, then the numerical range of the slider bar will be -1 to +1.

Alternatively, you can specify an increment for translation in the Increment field of the Scale Settings area. Pressing return while the mouse pointer is in the Increment field will cause the center of the Cone Tool to translate along the specified axis (or all axes) by the increment specified.

Orientation First click on the rotate icon. Next, pick an axis about which to rotate. Next pick an increment and limit (in degrees) and slide the slider to rotate the plane.

Scale First click on the scale icon. Next pick an axis direction to scale. Can only scale in the X (longitudinal) or Y (half conical angle) directions. Finally pick an increment and limit and slide the slider to scale the size of the cone tool.

Access: Transformation Editor > Editor Function > Tools > Cone

(see How to Use the Cone Tool)

Revolution Tool

Clicking on Editor Function in the Transformation Editor dialog and then selecting Tools and then Revolution from the pull-down menu configures the dialog as shown below.

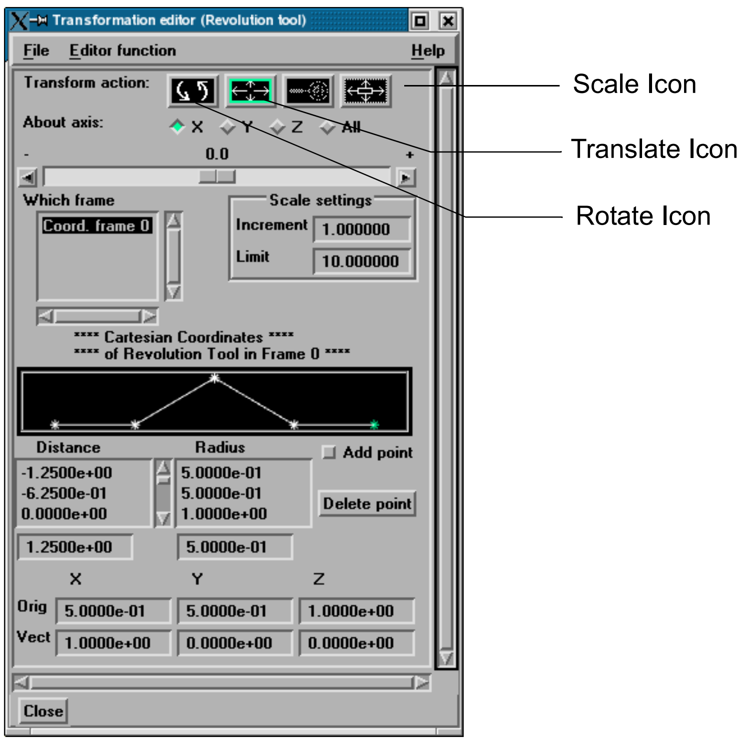


Figure 6-24
Transformation Editor (Revolution Tool)

For the Revolution Tool, you not only control the origin and direction vector, but the number of points and positions that are revolved about the axis.The desired coordinates of the Revolution Tool origin (center point of the thick direction line) are specified in the Orig. X, Y, and Z fields. The direction vector for the Revolution Tool direction axis is specified by entering the desired coordinates in the Vect X, Y, and Z fields for the selected Frame (or if more than one Frame is selected, in Frame 0).

Additional points may be added to the Revolution Tool by clicking on the Add Point(s) toggle and then clicking at the desired location in the schematic for the tool. There is no need to be overly precise in its placement since its location can be modified. Once you have added all of the new points you wish, the Add Point(s) toggle should be turned off. A point may be deleted by selecting it in the schematic area and then clicking the Delete button.

The position of any point may be modified interactively within the Revolution Tool schematic window, Simply click on and drag the point to the desired location. The precise location of any point may be specified by selecting the point in the schematic with the mouse and then entering the desired Distance (from the Revolution Tool origin) or Radius (from the axis) for the point in the text entry fields beneath the Distance and Radius Lists. Pressing return will enter the new value in the list above for the selected point.

The Transformation Editor can control precisely the position and size of the revolution tool.

Position It is possible to reposition the Revolution Tool origin by specific increments. First click on the translate icon. The Axis Button allows you to choose the axis of translation (X, Y, Z, or All) for the origin of the tool. The Slider Bar at Top allows you to quickly choose the increment by which to move the position of the origin. Dragging the slider in the negative (left) or positive (right) directions and then releasing it will cause the X, Y, and Z coordinate fields to increment as specified and the origin of the Revolution Tool to relocate to the new coordinates. The number specified in the Limit field of the Scale Settings area determines the negative (-) and positive (+) range of the slider. If the Limit is set to 1.0 as shown, then the numerical range of the slider bar will be -1 to +1.

Alternatively, you can specify an increment for translation in the Increment field of the Scale Settings area. Pressing return while the mouse pointer is in the Increment field will cause the center of the Revolution Tool to translate along the specified axis (or all axes) by the increment specified.

Orientation First click on the rotate icon. Next, pick an axis about which to rotate. Next pick an increment and limit (in degrees) and slide the slider to rotate the plane.

Scale First click on the scale icon. Next pick an axis direction to scale. Can only scale in the X (longitudinal) or Y (radial) directions. Finally pick an increment and limit and slide the slider to scale the size of the revolution tool.

Redraw This button will cause the Revolution Tool schematic window to re-center to the currently defined points of the tool.

Access: Transformation Editor > Editor Function > Tools > Revolution

(see How to Use the Surface of Revolution Tool)

Spline Tool

Selecting Spline from the Editor function Tools pulldown configures the dialog as shown below.

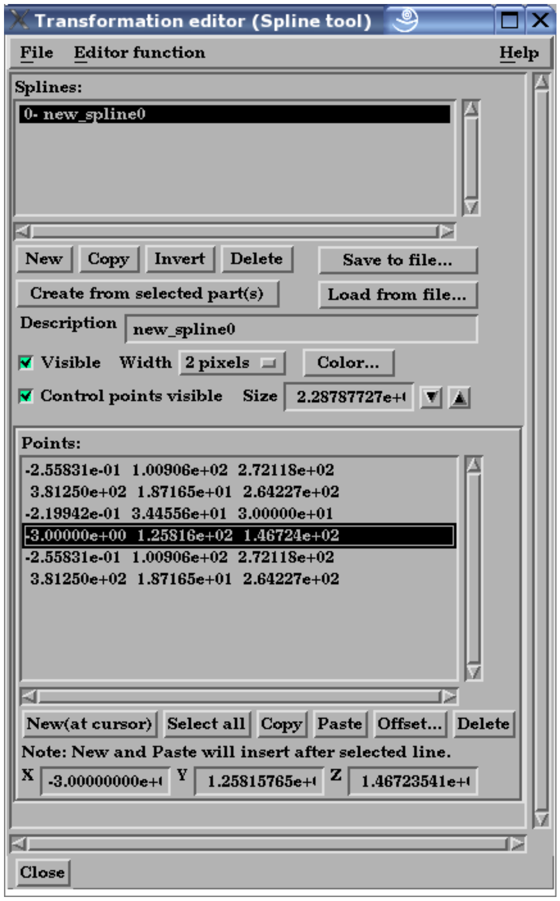


Figure 6-25
Transformation Editor (Spline Tool)

The Transformation Editor dialog enables you to create and edit the control points for a spline. A spline is used for one of three functions (a) path for a camera, (b) the path for a clip plane, and (c) the path for a distance vs. variable query.

New	Creates a new spline
Copy	Creates a new spline by copying the selected spline
Invert	Inverts the control points for the selected spline
Delete	Delete the selected spline(s)
Save to File	Save the selected spline(s) to a file
Load from File	Load splines from a file
Create from selected part(s)	Create a new spline and use all of the coordinates in the selected parts as the control points.
Description	Description of the spline
Visible	Toggles spline visibility
Width	Line width for the spline

Color...	Brings up the color chooser dialog to set the color for the spline
Show points	Toggles the visibility of the control points.
Size	The size of the control point markers in model coordinates
Points	The list of control points for the spline. Right click operations in this list include New, Copy, Paste, and Delete
New (at cursor)	Inserts a new control point in the selected spline after the selected point (or if no points are selected or exist at the end of the list) using the cursor tool as the location.
Select All	Selects all of the points in the list
Copy	Stores the coordinates of the selected points in preperation for a Paste operation
Paste	Paste the copied points and insert them immediately after the selected point (or if no points are selected or exist then at the end of the list).
Offset...	Brings up a dialog where a xyz offset value can be specified. The offset is added to the coordinates of the selected points.
Delete	Delete the selected points.
XYZ fields	Shows the XYZ values for the selected point. If modified will change the control point location.

(see How to Use the Spline Tool)

6.6 Window Functions

EnSight allows you to work concurrently with up to sixteen different sets of results data (computational or experimental). Each set of results data is read in as a “Case”.

Clicking the Case button in the Main Menu opens a pull-down menu which provides access to the following features:

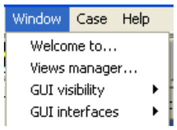
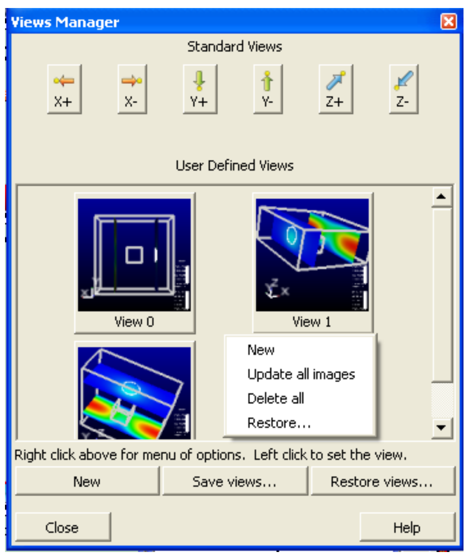


Figure 6-26
Window pull-down menu

Welcome — Displays the Welcome dialog.

Views manager — Displays the Views Manager dialog. Note that you can also click on the Views button below the graphics screen.



The dialog allows the user to create, save, restore, and apply views interactively. A View comprises all the viewing parameters and viewport parameters along with a snapshot of the user’s data. Along the top of the views manager are standard views designed to orient your model in the main graphics window. To create a new view right click on an empty area in the User Defined Views area and select New or click on the New button. A new view will be created in the User Defined Views area from your model viewing parameters in the main graphics window. To save your views to a folder, click on the Save Views button. To load (restore) views from a file, click on the Restore button. For more details, see How to Manage Views

GUI visibility — Opens the pull-down menu which allows you to toggle on/off the presence of the Feature Icons and the Feature quick interaction area in the GUI.

GUI Interfaces — Provides alternative Graphical User Interfaces. Two demonstrations are included with EnSight 9: A Simple Example GUI and CFD.

6.7 Case Menu Functions

EnSight allows you to work concurrently with up to sixteen different sets of results data (computational or experimental). Each set of results data is read in as a "Case".

Clicking the Case button in the Main Menu opens a pull-down menu which provides access to the following features:

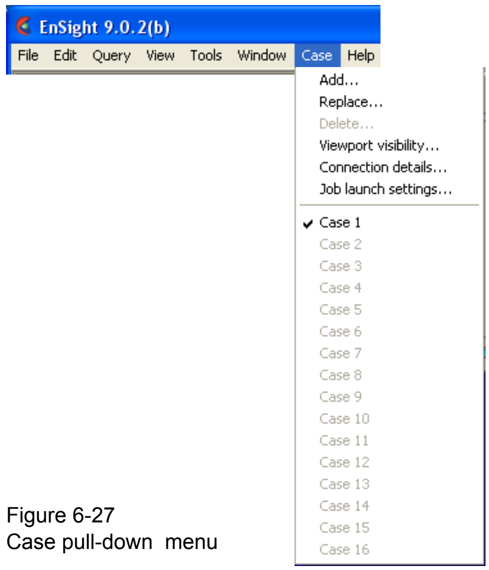


Figure 6-27
Case pull-down menu

Add... Opens a dialog which allows you to specify a name and other options for the new Case. .

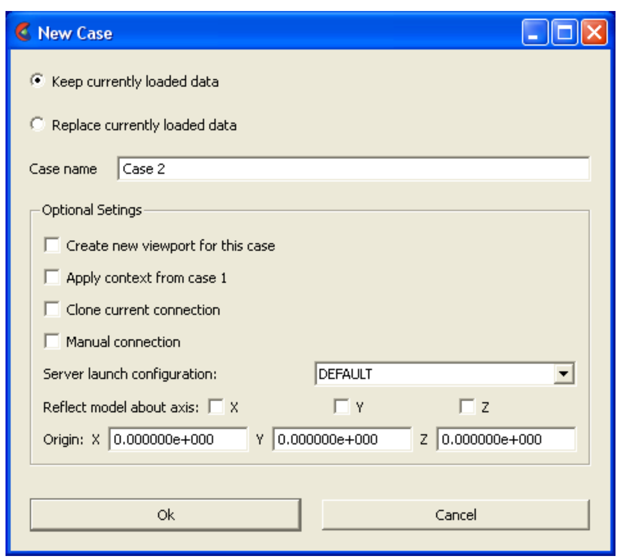


Figure 6-28
Add Case Dialog

The name will appear in the list of active Cases at the bottom of the Main Menu: Case pull-down menu as shown in Figure 6-27 above. Adding a Case actually starts a new EnSight Server and connects it to the EnSight Client. The Open... dialog will then open and you can read and load data files for the new Case and the data will be added to the data already present in the EnSight Client. When adding a case, you have some options as indicated.

Keep currently loaded data — The new case can be added to the existing one. This does not unload the current one and starts up a new server to load the added dataset

Replace currently loaded data — This deletes the existing dataset and loads the new data in place of it

Case name — The name that will appear at the bottom of the Case menu pulldown.

Create new viewport — The new dataset can be placed in a new viewport or added to the current.

Apply Context — The new dataset can have the context of case1 applied to it, which will cause it to basically inherit the positioning etc.of case 1.

Clone Current Connection — Use the same parameters to start up the new server as the existing server used This can be useful if you used a complicated set of options to start up the server for the original dataset and don't want to have to repeat them for this dataset.

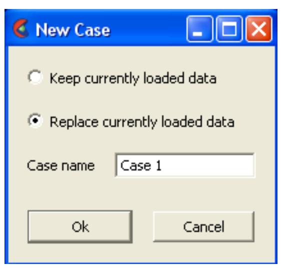
Manual Connection — If you would like to start up the EnSight server for your new case manually toggle this ON.

Server Launch Config — Choose a launch configuration for the server. You can define server launch configurations using the Case>Job Launch Settings...

Reflect Model — This allows you to load the model in already creating a reflection about the axis at load.

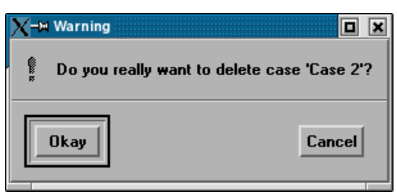
Origin — Specify the origin of the data.

Replace... — Replacing a Case causes all parts and variables associated with the active Case to be deleted. The Server will be restarted and assigned the new Case name. Clicking the Replace... button opens a small dialog which allows you to specify a name for the Case



you wish to use to replace the Case currently selected in the Main Menu: Case pull-down menu as shown in Figure 6-27 above. The Open... dialog will then open and you can read and load data for the new Case.

Delete — Deleting a Case causes all parts and variables associated with the Case to be deleted and terminates the Server associated with the Case. Clicking the Delete button opens a Warning Dialog which asks you to confirm that you wish to delete the Case currently



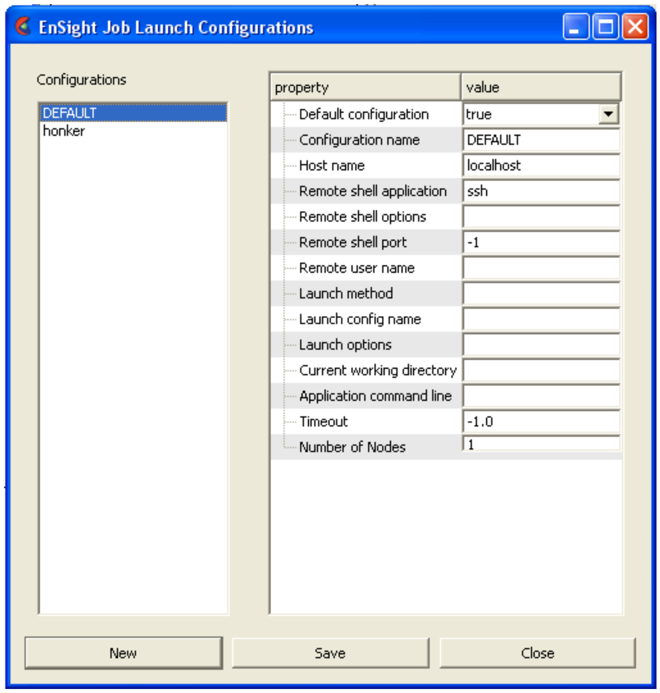
selected in the Main Menu: Case pull-down menu as shown in Figure 6-27 above.

(see How To Load Multiple Datasets (Cases))

Viewport visibility... — Opens the Case visible in which viewport dialog which allows you to specify in which Viewports (including the Main Graphics Window) you wish to make the parts associated with the currently selected Case visible. Parts associated with the selected Case will be visible in the viewports outlined in green and invisible in those outlined in red. Visibility for specific Parts can of course be toggled on/off using the Part Visibility Icon in the Part

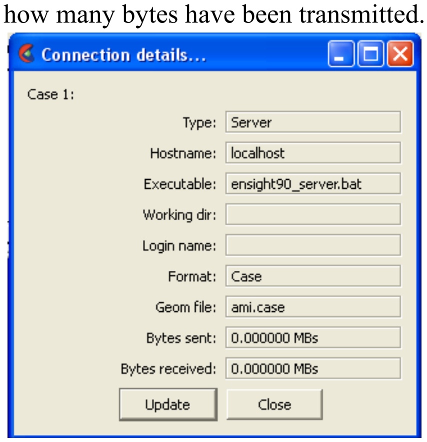
Mode Icon Bar.
(see Part Visibility Toggle Icon in Section 8.1, Part Mode)

Job Launch settings... Opens the Connection settings dialog. From this dialog you can perform the various actions for adding or replacing connections from the client to a server or an SOS on your favorite machines. You can control whether automatic or manual connections will occur, and can manage and save this information for future use.



(see How to Connect EnSight Client and Server)

Connection details... Opens the Connection details... dialog which gives information about the connection and how many bytes have been transmitted.



Finally, at the bottom of the pull-down menu you will find a list of active Cases, The toggle buttons allow the selection of only one Case at a time. In Figure 6-27 above, Case 1 is the currently selected Case. The current selected Case is the one which will be affected by the Data Reader, Querys, and many other operations.

6.8 Help Menu Functions

Clicking the Help button in the Main Menu opens a pull-down menu which provides access to the following features:

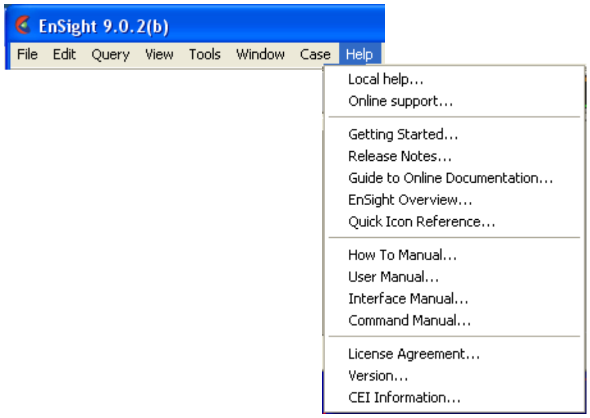


Figure 6-29
Help pull-down menu

Local help... Opens a site-specific, optional, local help document (of one exists). Simply place a .pdf file in your $CEI_HOME/ensight92/site_preferences directory named "LocalHelp.pdf". This is a hook for a sites that want to provide site-specific help for their users. This might include customized preferences, help manuals for local User-Defined Readers, instructions for using customized macros, etc.

Online Support... Opens the CEI Online Support tool for quick, easy, and comprehensive problem reporting, key requesting, and EnSight assistance. The tool has four tabs: System Info, Report Trouble, Key Request, Web Tool.

System Info — Click on this tab to review the information that has been collected about your system. This tab collects the information about the EnSight Application that you are using, your EnSight license, the graphics card you are using, your System Environment variables, and a Screen Shot EnSight's graphics window. This information is useful for more rapid troubleshooting of your EnSight issue.

Report Trouble — Click this tab to send a problem report to your EnSight Distributor. In this Tab, Toggle on the System Information to send, fill out the form, Select a Destination that is your CEI distributor (CEI in the U.S.), and click Send Report.

Key Request — Click on this tab to request a new or changed license key. Fill in the information and Select a Destination that is your CEI distributor (CEI in the U.S.), and click Request Key.

Web Tool — A web browser pointing to EnSight's extensive online assistance documents. In the Bookmarks pulldown, we have Training (information on upcoming training classes), Interfaces (how to get your data into EnSight) , Documentation (Online Documentation), Tips and Tutorials (step by step how to do stuff), Feature Requests (ask CEI to enhance EnSight), Change Logs (what was fixed/added to what version of EnSight?), and Knowledge Base articles (Frequently Asked Questions). These categories are available on our website, www.ensight.com and click support category.

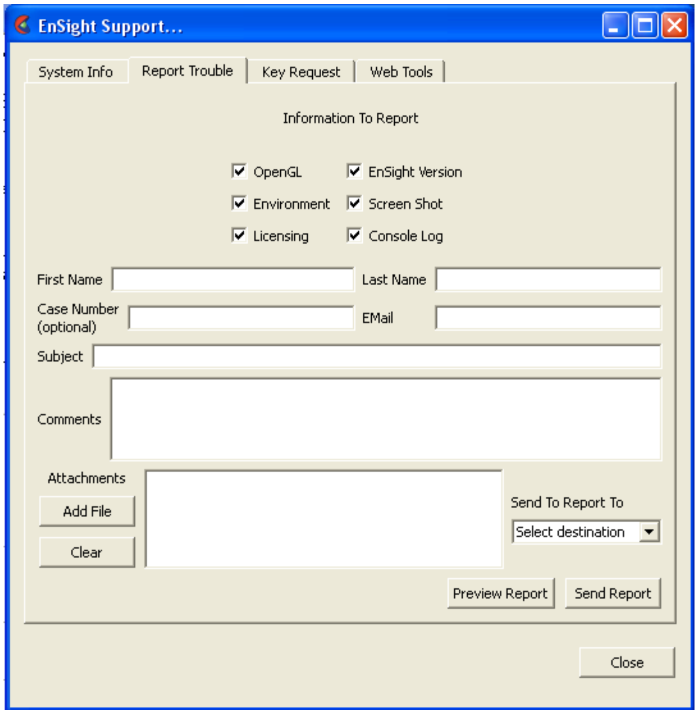


Figure 6-30
Online support tool

Getting started...	Opens the Getting Started Manual on-line. *Note that this document is not cross-referenced within itself or to other documents.*
Release notes...	Provides an overview of changes made since the last major EnSight release.
Guide to online documentation...	Provides a guide to the use of the On-Line Documentation.
EnSight overview...	Provides an overview of EnSight.
Quick icon reference...	Provides a quick reference guide to all EnSight GUI icons, many of which have links to appropriate How To documents
How to manual...	Opens the How To Manual on-line.
User manual...	Opens the User Manual on-line.
Interface manual...	Opens the Interface Manual on-line. The Interface Manual has the
Command manual...	Opens the Command Language Manual on-line.
License agreement...	Opens up On-Line Documentation to the text of the EnSight End User License Agreement and the EnSight Support and Maintenance Service Agreement.
Version...	Opens up the Version Information dialog which states the version number of the EnSight software currently running.
CEI information...	Opens up the CEI Information display which gives full CEI contact information.

7 Features

This chapter describes the functions available through the Feature Icon Bar.

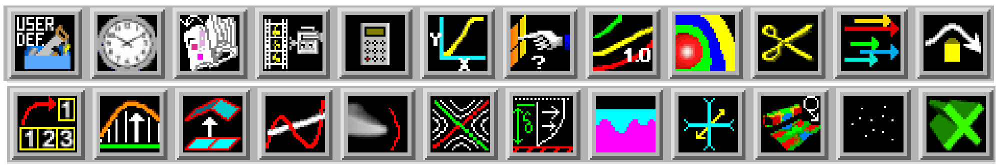


Figure 7-1
EnSight Feature Icon Bar

7.1 Solution Time

Many analyses contain time dependent information, such as automobile crash simulations and unsteady flow problems. The presence of time-dependent data is indicated to EnSight through an EnSight result file, case file, or is determined directly from the data files of other formats. EnSight has the capability of displaying the model and results at any time provided for in the data. Linear interpolation between given time steps is possible as long as the geometry does not have changing connectivity over time.

EnSight keeps track of which variables and Parts have been created so that if you change time steps, variables and Parts will update appropriately. For example, assume you have created a clip plane through the combustion chamber of an engine. From this clip plane you have created two constant variables Min Temperature and Max Temperature and are displaying them in the Main View. Now change time steps. First, the geometry updates to a new crank angle position. Second, the clip plane will automatically be recalculated to fit the new geometry. Third, the Min and Max values displayed in the Graphics Window are recalculated and updated. This is all performed automatically by EnSight after you change the current time value.

It is important to distinguish between time step and solution time. An example will best illustrate this concept.

Consider a model with data for 5 different times:

Time Step	Solution Time
0	1.0125
1	11.025
2	11.50
3	13.00
4	21.333

Note that the time steps coincide with the number of transient data files and are integers. The solution time at each time step comes from the analysis, and does not have to be at uniform intervals. The solution time can be in any units needed, but must be consistent with the solution files. That is, if a velocity file was in terms of meters per second, then the solution time must be in terms of seconds. Hence it is not possible, for example, to have the solution time reported in degrees crank angle for a combustion case unless the corresponding solution files were also in terms of crank angle (otherwise velocity would be reported in the meaningless units of meters-per-degree-crank-angle).

Important! EnSight will sort the solution time values and, after the sort, they must be monotonically increasing. An error will occur if any time values (represented as singe precision floats) are the same.

A special Solution Time dialog gives you control over time and relates time step to solution time. You can force the time information to conform to the actual time data given at the steps, or you can allow interpolation to occur between time steps. You must be aware of the implications of such an interpolation and choose the method that is appropriate.

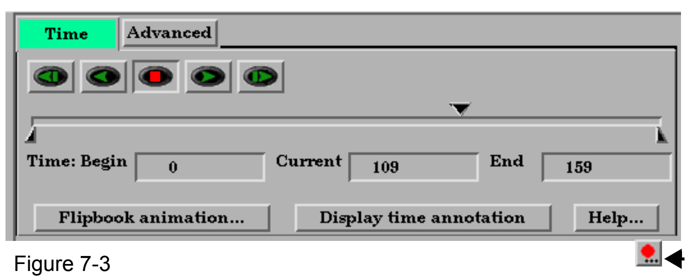
Also, you can see the ranges of time dependent data available and the current time that is set for the Main View. You can change time steps by either entering a new time to view, or using the Solution Time slider bar.

The Solution Time Dialog shows a composite timeline of all timesets from all cases. For any case, a number of different timesets can exist. Each timeset can be attached to multiple variables and/or geometry. This makes it possible to, for example, have one variable defined at t = 1.0, 2.0, 3.5 and another variable defined at t = 1.5, 2.01, 4.0. For each timeline, controls exist to specify how EnSight should interpolate the variables when time is set to a value not defined for a given timeset. Also a convergence time value can be defined to merge timesteps with time values that are very close, in the above example, 2.0 and 2.01 will be merged if the master timeline convergence time is set to 0.01.

There are other places within EnSight where time information is requested. These include, traces, emitters, animated traces, flip book transient data, key frame animation transient data, and Query/Plot. Each of these use the specified Beg/End values. For functions which do not explicitly specify the time step the current display time (as defined in the Solution Time Dialog) is used.

For data that is transient, the Solution Time Icon will be visible. For static data that does not change over time, there will be no Solution Time Icon on the Feature Icon Bar. Clicking once on the Solution Time Icon opens the Solution Time Editor in the Quick Interaction Area which is used to specify time information.

Figure 7-2
Solution Time Icon



Record Button
(Enabled with active
animation playing)

Figure 7-3
Quick Interaction Area - Solution Time Editor

The range of both Time Steps and Simulation Time is shown at the top of the Editor. The default Quick Interaction Area is the Time tab which is highlighted.

Beg — Value for Beginning Time Step or Simulation Time. The left slider below the rail can be used to select the beginning value.

Cur — Value for Current Time Step or Simulation Time. The slider above the rail can be used to select a value for the Current Time Step field.

End — Value for Ending Time Step or Simulation Time. The right slider below the rail can be used to select the ending value.

Video Controls — Click on the Video Playback control buttons to go back on timestep, to play in reverse, to stop, to play forward and to step forward one timestep, respectively. Note that this form of playing the animation is a solution time stream of the data. That is, each timestep is read from the data file without storing the animation in memory.

Record Button — Opens a dialog for recording transient animation to a file. This button is only enabled when an active animation is playing.

Flipbook Animation... — Opens the Flipbook Animation Editor.
(see Section 7.2, Flipbook Animation)

Display Time Annot — Toggles an annotation in the graphics area of the Time value. Click on the Annot Tab to the left of the graphics area to move, modify, etc. this annotation.

Figure 7-4
Quick Interaction Area - Solution Time Editor

# of Cycles — For cyclic transient analysis, the solution is often computed for one cycle only. It is often desirable to be able to visualize more than one cycle. This is possible only if the first and last timesteps contain the same information. By default, EnSight assumes one cycle.

Scale Type — Opens a pop-up menu to specify use of existing time steps, or allow EnSight to linearly interpolate to show any time step. Choices are:

Discrete	Can only change time to defined steps.
Continuous	Can change time to any time, including times between steps. Only available if do not have changing geometry connectivity transient case.

Units — Opens pop-up menu to specify whether to use and display:

Step	which will be an integer showing time as step data. Will show NOSTEP if in *Continuous* mode and current time is not at a given time step. Current time will automatically change to keep within range Begin/End range. The default beginning and ending simulation times correspond to the first and last time steps specified in the results.
Simulation Time	which will be a real number showing true simulation time. Current time will automatically change to keep within the Begin/End range. The default beginning and ending simulation times correspond to the first and last times specified in the results.

Increment — Specifies the incremental time which will be applied to the current time each time the slider stepper buttons are used.

Timeset Details... will open the Timeset Details dialog.

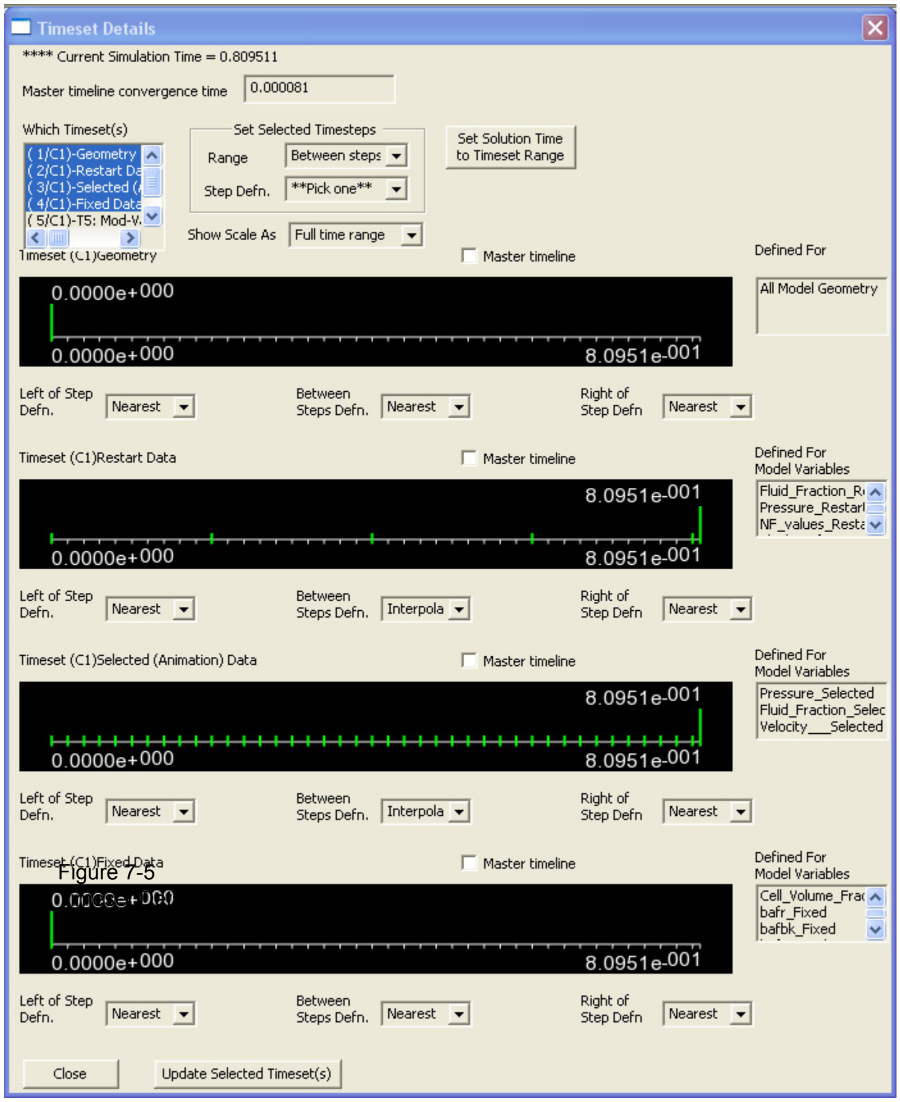


Figure 7-5

Master timeline Convergence time A tolerance value used to determine if two different times should be treated as the same time. For example 2.0 and 2.01 in two *different* timesets will be treated as the same time if the convergence time is set to 0.01. Time values in the same timeset are not merged.

Which Timeset(s) Selects the timesets to be viewed.

Modify All Selected Timeset(s) Allows modification of all selected timesets.

Range which time range to modify.
Update Step Defn. To Choose how to modify the selected Timeset's Range.

Set Solution Time To Timeset Range Will set the Solution Time Beg. and End. time values to correspond to the selected timeset.

Show Scale As "Full Time Range" will show the Timeset's values in relation to the full composite timeline. "Timeset's Range" will adjust the beginning and ending boundaries of the graphic timeset to correspond to the begin and end values for the timeset. The change will not take effect until the "Update Selected Timeset(s)" button is pressed.

Defined For Lists all of the variables and/or geometry attached to the Timeset.

Left/Right of Step Defn.	When the Current time is less than the Timeset's minimum time, the attached variables will use the Nearest values or become Undefined.
Between Steps Step Defn.	When the Current time is between the Timeset's minimum and maximum time values, but not defined, the attached variables will use the Right/ Left, Interpolate, or Nearest values, or become Undefined.
Update Selected	Must be selected in order to update any changed Timeset.

7.2 Flipbook Animation

There are four common animation techniques which are easily accomplished with Flipbook Animation. They are:

- animation of transient data, which can be any combination of scalar/ vector variables, geometry, and discrete Particles
- animation of mode shapes based on a mode-shape displacement variable
- animation of a Part moving or changing value during animation, such as sweeping a 2D-Clip Plane or changing the value of an isosurface.
- animation applying a linear interpolation of a vector displacement field value factor from 0 to 1.

You can combine any of these techniques with the animation of Particle traces discussed in the previous Section 7.4.

The concept of a flipbook is similar to the stick figures you have probably seen in books where each page contains a picture. When you flip through the pages quickly you get the sense of motion. Flipbook animation stores a series of "pages" in Client memory which are then rapidly played back to create the illusion of motion. Pages can be loaded as *graphic images*, which may playback faster; or as *graphic objects*, which can be transformed after creating the flipbook, even while the flipbook is running.

For animation to be of interest, something must change from page to page. For *transient-data* flipbooks, you must have visualized something about the model that changes over time. For *mode-shape* flipbooks, you need to have set the displacement attributes of the Parts for which you want to see mode shapes (see Section 7.10 Displacement On Parts). For *created-data* flipbooks, you need to have used the Start/Stop utility or specified Animation delta values for the Parts.

The number of pages in the flipbook determines the length and smoothness of the animation. You directly or indirectly specify how many pages to create. While the Server performs the calculations, the Client stores the flipbook pages in memory. Just how many pages you can store depends on the amount of memory installed on your Client workstation. Your choice to load graphic images or graphic objects affects memory requirements, but the complexity of the model and the size of the Graphics Window determine which will use less memory in any particular situation.

You can control which original model Parts and created Parts will be updated for each time increment as the user chooses. This feature takes all dependencies into account. For example if an elevated surface was created from a 2D clip plane, the clip plane would be updated first and then the elevated surface based on the new clip. The ability to choose which Parts are or are not updated allows before and after type comparisons of a Part.

After creating the flipbook, options for displaying it include: running all or only a portion of it, adjusting the display speed, running under manual control or automatically, and running from the beginning or cycling back-and-forth between the two ends.

It is important to know that objects in the flipbook cannot be edited. If you wish to change something in the flipbook, you must reload it. If you decide to regenerate a flipbook (after changing something), you can choose to discard all the old pages, or keep any old page with the same page number as a new page.

This is very useful when you first load every tenth frame then decide to load them all. EnSight will not have to reload every tenth frame that already exists. When you are done with a flipbook, remember to click Delete All Pages. This will free up memory for other uses.

Note that alpha transparency works with depth peel not sorted transparency.

Flipbook vs. Keyframe While you can implement any flipbook animation technique with keyframe animation (described in the next section), flipbook animation has three advantages. First, graphic-object-type flipbooks allow you to transform the model interactively to see from many viewpoints. Second, graphic-image-type flipbooks can be saved to a file and later replayed without having to have the dataset loaded, or even being connected to the Server. Third, the speed of display can be more interactive because the flipbook is in memory and can be flipped through automatically or stepped through manually.

Flipbook animation has a few disadvantages. First, you cannot change any Part attributes, except visibility and material properties, without regenerating the flipbook. Second, each page is stored in Client memory, which limits the number of pages and hence the duration of the animation.

Four Animation Techniques:

1.Transient Data Transient-data flipbooks have pages that correspond to particular solution times; i.e. step or simulation. You specify at which time value to start and stop the animation, and the time increment between each page. The time increment can be more than one solution-time value; this is useful in finding a range of interest or for a coarse review of the results. The increment can be a fraction, in which case the data for a page is interpolated from the two adjoining solution-time values.

2. Mode Shapes Mode-shape flipbooks are used to show primary modes of vibration for a structure. This is done by using a per node displacement, enabling the Part to vibrate. While you can use any vector variable for a displacement, to see actual mode shapes you need to have a Results-file vector variable corresponding to each mode shape you wish to visualize. Note that you can create copies of Parts and simultaneously display them with different mode-shape variables, or one at its original state and the other with displacement for comparison.

The first page of a mode-shape flipbook shows the full displacement (as it is normally shown in the Graphics Window). The last page shows the full displacement in the opposite direction. The in-between pages show intermediate displacements in proportion to the cosine of the elapsed-time of the animation.

3. Created Data Created-data flipbooks animate the motion of 2D-Clips and Isosurfaces according to their animation attributes. This animation allows you to show clipping planes sweeping through a model or to show a range of Isosurface values. The first page shows the Part's location as it appears in the normal Graphics Window. On each subsequent page, each 2D-Clip is regenerated at the new location found by adding the animation-delta displacement to the 2D-Clip's location on the previous page. Also, each Isosurface is regenerated with a new iso-value found by adding the animation-delta increment to the iso-value of the previous page.

4.Linear Load Linear-loaded flipbooks are used to animate a displacement field of a part by linearly interpolating the displacement field from its zero to its maximum value. The variable by which the part is colored also updates according to the linearly displaced values. Like Mode Shapes, this utilizes a per node displacement. The function can be applied to any static vector variable.

Clicking once on the Flipbook Animation Icon opens the Flipbook Animation Editor in the Quick Interaction Area.

Figure 7-6
Flipbook Animation Icon

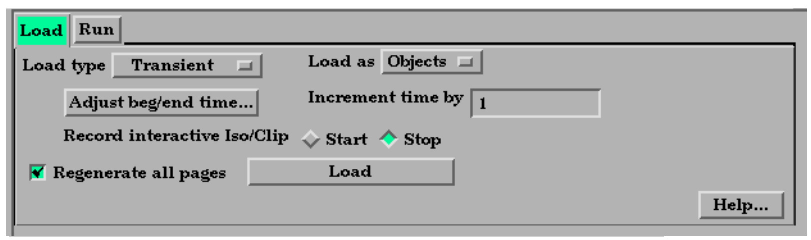


Figure 7-7
Quick Interaction Area - Flipbook Animation Editor

Load Type Opens a pop-up menu for the selection of type of flipbook animation to load. Options are:

Transient animates changes in data information resulting from changes in the transient data. For example, changes in coloration resulting from changes in variable values, or changes in displacement of Parts. See discussion in the introduction section.

Mode Shapes animates the mode shape resulting from a displacement variable. See discussion in the introduction of this section.

Create Data animates Parts having nonzero animation-delta values or which have been recorded with the Start/Stop utility. See discussion in the introduction of this section.

Linear Load animates the Displacement (vector) variable of a part by linearly interpolating the displacement field from its zero to its maximum value. The Color variables of the part also update according to the linearly displaced values.

Load As Opens a pop-up menu for the selection of whether to load flipbook pages as Graphic Images or Graphic Objects.

Graphic Objects flipbooks enable you to transform objects after creating the flipbook. Playback performance depends on the complexity of the model.

Graphic Images flipbooks may be saved for later recall, but they cannot be transformed, nor can the window be resized. Playback performance depends on the Graphics Window size.

Adjust Beg/End Time... Opens the Solution Time Editor in the Quick Interaction Area. To return to the Flipbook Animation Editor from the Solution Time Editor, click on Animate Over Time... (When loading Transient data, the flipbook will start and stop at the Beg/End values specified in the Solution Time Editor.
(see Section 7.2, Flipbook Animation)

Increment Time By — In this field you specify the increment of each transient-data flipbook page which corresponds to the range type specified in the Solution Time Editor,

Note: *If you enter a Begin, End, or Increment value not corresponding exactly to a Step or Simulation time value, EnSight will interpolate the values, affecting the appearance of each page (only if geometry connectivity is not changing).*

Record Interactive Iso/Clip — Allows you to define the change (isovalue change or clip plane movement) in an isosurface or clip plane which will take place during the Flipbook load. Only isosurfaces and clip planes which are modified in interactive mode are tracked.

Regen all pages — Toggling this on reloads all flipbook pages. Toggling it off saves load time if some pages have already been loaded, for example if you've just canceled the load, the pages already created during the load remain in memory. Also, you can quickly load by increment of 2 just to see if your animation looks good, then you can change the increment to 1, toggle off this toggle, and you'll only load the odd pages that have not yet loaded.

Load — Clicking this button starts the loading flipbook pages and opens a pop-up dialog which reports the progress of the load and then closes to signal load is complete. If you cancel the load, the pages already created during the load remain in memory.

Run Tab — Click on the Run tab at the top of the Flipbook quick interaction area to get the following dialog.

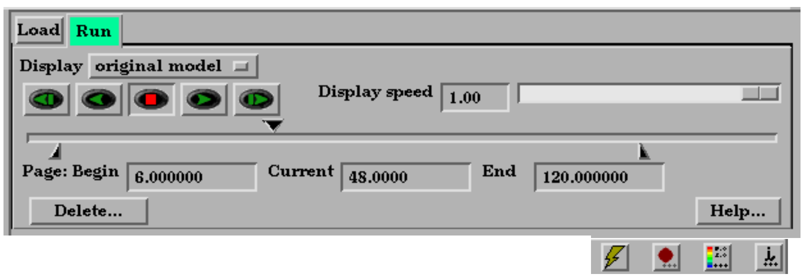


Figure 7-8
Quick Interaction Area - Flipbook Animation Editor

Record Button

Display — Opens a pop-up menu for selection and clarification of what is being viewed.
original model - This is the original model.
flipbook pages - These are the flipbook pages stored in memory.

Video Controls — Click on the Video Playback control buttons to go back on timestep, to play in reverse, to stop, to play forward and to step forward one timestep, respectively.

Display Speed — This field specifies the playback-speed factor. Varies from 1.0 (full speed of your hardware) to 0.0 (stopped). Change by entering a value or clicking the stepper buttons.

Beg — Value for Beginning Time Step or Simulation Time. Type a value in the field, or the left slider below the rail can be used to select the beginning value.

Cur — Value for Current Time Step or Simulation Time. Type a value in the field, or the slider above the rail can be used to select a value for the Current Time Step field.

End — Value for Ending Time Step or Simulation Time. Type a value in the field or the right slider below the rail can be used to select the ending value.

Delete... — Opens a pop-up warning dialog which asks you if you really wish to delete all loaded pages. Click Okay to delete all loaded flipbook pages and free the memory for other use.

Record Button — In order to save an *active* flipbook animation, click on the record button found below the Quick Interaction Area.

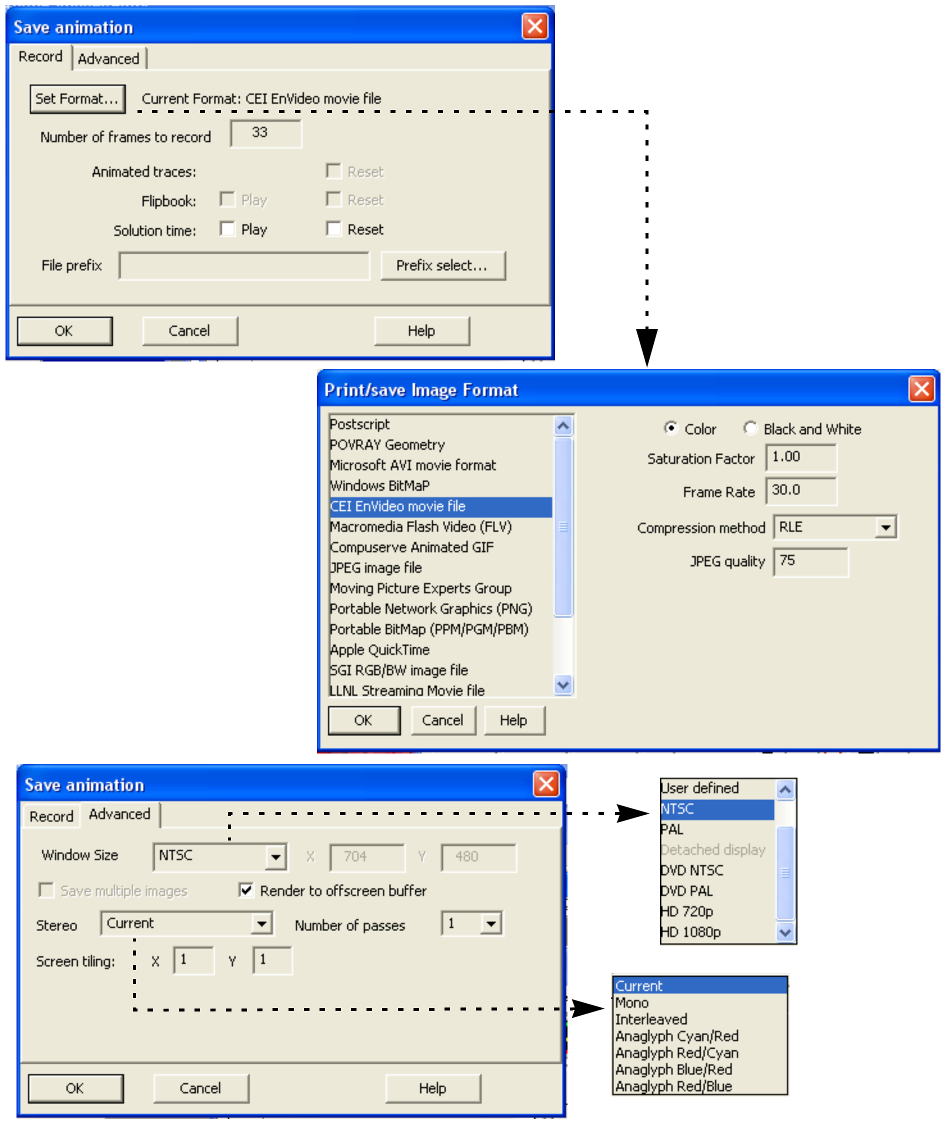


Figure 7-9
Save Flipbook Pages To dialog

Record Tab	Click on the Record Tab (comes up by default in this mode when you click on the Record Button).
Set Format...	Brings up the Image/Movie Format and Options dialog. The flipbook will be recorded using the selected format. (see Section 2.12, Saving and Printing Graphic Images and How to Print/Save an Image)
Number of frames	Choose the number of frames to record.
Reset Anim Traces	This will reset the animated traces to the beginning prior to starting the record. This toggle is only active if you have animated traces.
Play the flipbook	This will play the flipbook.
Reset Flipbook	This will reset the flipbook to the beginning prior to starting the record.
Play solution time	This will play through the solution time.
Reset Soln Time	This will reset the solution time to the beginning prior to starting the record.
File Prefix	The location and filename prefix for the recorded images. The appropriate suffix will be added automatically.

Advanced Tab	Click on the Advanced Tab.
Window Size	Brings up the Image/Movie Format and Options dialog. NTSC - (704 x 480 pixels) format suitable for older recording media PAL - (704 x 576 pixels) format suitable for older recording media Detached display - DVD NTSC - (720 x 480 pixels) DVD PAL - (720 x 576 pixels) HD 720p - (1280 x 720 pixels) HD 1080p - (1920 x 1080 pixels) Normal - (current graphics screen size) - machine/graphics/monitor dependent User Defined - (user entered horizontal x vertical pixels) - not limited to screen size. The flipbook will be recorded using the selected format. (see Section 2.12, Saving and Printing Graphic Images and How to Print/Save an Image)
Save multiple images	
Render offscreen buffer	Render offscreen (if your card supports this feature). Toggle this on to avoid having the image appear to your screen to be saved. This avoids the risk of saving overlapping windows onto your image.
Stereo	EnSight offers several options for saving stereo files. *Current* - Save according to the display settings *Mono* - Don't save in stereo *Interleaved* - Left and right stereo saved together. *Anaglyph Cyan/Red* - Save two files for anaglyph stereo (colored glasses) using cyan in the left eye and red in the right eye. *Anaglyph Red/Cyan* - Save two files for anaglyph stereo (colored glasses) using red in the left eye and cyan in the right eye. *Anaglyph Blue/Red* - Save two files for anaglyph stereo (colored glasses) using blue in the left eye and red in the right eye. *Anaglyp Red/Blue* - Save two files for anaglyph stereo (colored glasses) using red in the left eye and blue in the right eye.
Number of passes	Number of images to use in antialiasing algorithm. The higher the number, the smoother that jagged lines will appear but the longer that your image will take to save.
Screen Tiling	Graphics images can be saved in multiple sections called Tiles. These can be re-rendered back onto multiple displays, using, for example, our free EnVideo application.

Troubleshooting Flipbook Animation

Problem	Probable Causes	Solutions
No motion	No pages are loaded.	Load flipbook pages.
	All pages are the same visually.	In order to see motion there must be a difference between one page and the next. Reload with differing Part attributes, such as coloring by a variable, using displacements, etc.
	Run Type set to Step or Off	Select Run Type to be Auto
Speed too fast	Display Speed is set too fast.	Change speed.
Speed too slow	Display Speed is set too slow.	Change speed.
	Hardware bottleneck (computer simply isn’t sufficiently powerful)	Reduce the number of pages.Load pages as graphic images.
Speed erratic	Virtual memory is swapping pages to and from disk storage.	Only load the no.of pages that fit into the workstation’s main memory.
Mode Shape(s) not visible	Wrong Load Type setting	Change Load Type to Mode Shapes and reload.
	Displacement attributes are incorrect.	Change Displace by and Factor attributes for the Part to animate.
2D Clip plane(s) not moving	Wrong Load Type setting	Change Load Type to Create Data and reload.
	Plane was not moved interactively between Start and Stop.	
Isosurface(s) not moving	Wrong Load Type setting	Change Load Type to Create Data and reload.
	Isosurface was not moved interactively between Start and Stop.	
Transient data ignored	Wrong Load type	
	Solution time step specifications are incorrect.	Change Load Type to Transient and set Solution time values according to available time steps.
Pages lost	Show From or Show To pages are not at ends of flipbook.	
	Old pages are being regenerated.	Toggle-off Regen. All Pages
	Delete All Pages is clicked.	Recover using the session command file.
Transformations do not work	Flipbook pages are loaded as graphic images.	Reload flipbook pages as Graphic Objects

7.3 Keyframe Animation

Since its initial release in 1987, EnSight has been used extensively for animation, due to its easy-to-use keyframe animator and ability to handle transient data. This mechanism allows you to create your own movie sequence to present your results more easily. There seems to be two mind sets when it comes to animation. The first group of people believe animation to be totally trivial—something that can be completely finished in an hour or two by anyone. The other group of people seem to believe that animation is something that takes many days, if not weeks to finish and requires an “animation expert” to get done. Well, neither of these ideas are correct. While animation is not trivial, it is also not overly complicated. Most animation produced by EnSight is setup during a day, and recorded the same day or during the night to be complete by the next morning. Engineers create and record their own animations. EnSight is intended to be used by end users—this includes the animation module. We do acknowledge, however, that there is a difference between animation, and animation done well. The latter comes with time and experience.

EnSight uses a modified keyframe technique. This technique enables the user to define what the scene should look like at certain times called Keyframe. Each keyframe can be different from a previous keyframe by using any combination of rotate, translate, scale, zoom, look-at, or look-from operations. A given keyframe can also be the same as the previous frame (the purpose of which will be explained shortly). The keyframe technique only works on transformations, and is not used for other items related to what the scene looks like (i.e., when to turn on Parts, do isosurfaces, shading changes, etc.). EnSight actually keeps track of the transformation commands performed between keyframes and linearly interpolates these commands when creating frames between the keyframes. These in-between frames are referred to as subframes.

Each keyframe includes the following information: (1) a set of transformation matrix values, specifying each local frame, the global frame and the Look-At and Look-From Points; (2) the value of all isosurfaces and position of all clip Parts using the plane tool; (3) the specific keyframe attributes; and (4) the transformation commands and isosurface values to get the scene and clip Parts to the next keyframe.

When running keyframe animation, EnSight performs the following actions for each keyframe: (1) any command language commands associated with the keyframe are executed, (2) the specified number of subframes are displayed in sequence, interpolating the transformation commands to get to the next keyframe.

To begin the process of creating an animation sequence, first define the scene you desire for the first keyframe. Then, turn on keyframe animation and create this scene as your first keyframe. You can then proceed to modify the orientation of the model and create your other keyframes.

If you make mistakes during the keyframe definition, click Delete Keyframe... and enter the number of the last keyframe you were satisfied with. Then, proceed to define the subsequent keyframes again. As soon as you have at least two keyframes defined, you may play back the animation to see what it looks like. To do this, select the Run Animation button in the Quick Interaction Area. The animation process generally proceeds with some keyframe definitions, running what you have so far after some of those definitions, once in a while a delete back to operation, more keyframe definitions, etc., until you are satisfied with the entire animation sequence. You then set up the record information and set the process in motion to produce the images.

Note, that when playing back the animation, you do not have to always play the entire sequence. Run From, and To frame capability is provided. You also can abort an animation run by entering the “a” key in the graphics window.

In order to get the length of animation you want on video, you will need to adjust the number of sub-frames between keyframes in the Speed/Actions tab of the Run Attributes dialog. The total number of frames displayed during animation is the sum of the keyframes plus the sum of the subframes. The NTSC broadcasting standard calls for 30 frames displayed per second. It can be difficult to get a feel for how fast the animation will be once recorded. The speed of the playback on the workstation is related both to its graphics capability and the complexity of the scene, so reducing the complexity will speed things up. Accordingly, you might consider options like making all but a representative Part invisible, use the feature angle option to reduce the visual complexity of the Parts, and/or use the dynamic/static box drawing modes.

Anything that is currently on will be on during the animation. That is, if contours, vector arrows, Particle traces, shaded surfaces, flipbook animation, animated traces, etc. are on, they will be on during animation. If any Parts have an animation delta set or are dependent on a Part that has the delta set then they will be regenerated and change through the animation. This enables you to do any of the flipbook animation techniques within keyframe animation for recording purposes, including the use of transient data (See Flipbook Animation). The advantage for doing flipbook techniques within keyframe animation is that they can be recorded and the amount of memory used is smaller because the whole flipbook is not loaded into memory. This enables the recording of long sequences of changing information that would not be able to be shown fully with flipbook animation because of memory limits of the workstation. Short sequences that you have already loaded into the flipbook can also be used by making sure that the Flipbook Run toggle is on before running keyframe animation.

If dealing with transient data, you should set up the keyframes for display of the model first, play it back, edit, etc. Then, after you are satisfied with the model presentation, you can start dealing with displaying the transient data on the model. You should be careful in doing movement of the model while transient data is being displayed. It can be confusing to have the transient data changing at the same time that the model in the scene is moving. When dealing with transient data, we normally introduce the problem first with some keyframes, then run the transient data without any transformations by defining two successive identical keyframes. Between these two identical keyframes, we animate the transient data using one of the several methods available.

We have attempted to create the animation module to be able to run in a set-up-walk-away mode to create video. In order to do this, you can issue command language lines at each keyframe (except the last one). For example, if you had a case where you wanted to first show off some of the model, and then turn on fringes to show results, you could issue a “view: fringes on” command at the keyframe. It is also possible to play a command language file using this option. Care should be taken to *not* issue an `anim_keyframe: run` command as Part of this command language (which would cause an infinite loop).

When saving images to disk files, be aware that image files can take a great deal of disk space. The file system that you are writing images to should be monitored during the animation run to make sure it doesn't run out of space.

Clicking once on the Keyframe Animation Icon opens the Keyframe Animation Editor in the Quick Interaction Area.

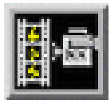


Figure 7-10
Keyframe Animation Icon

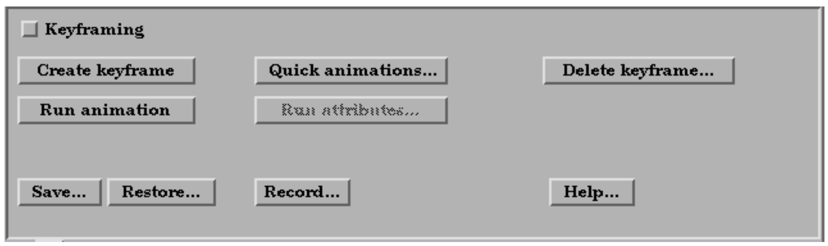


Figure 7-11
Quick Interaction Area - Keyframe Animation Editor

Keyframing Toggle
Toggles-on/off Keyframe animation feature. WARNING: If you toggle-off Keyframing, all the keyframes previously created will be lost (see Save... below).

Create Keyframe
Click this button to create a keyframe. If Keyframing toggle is not turned on then creating the first keyframe will turn it on automatically. Keyframes are automatically numbered in sequence of their creation. As each keyframe is created, a message appears in the Status History Area.

Quick Animations...
Opens the Quick Animations dialog which allows you to add keyframes of predefined movement to your animation. Currently implemented are "fly around", "rotate objects", and "exploded view"

Delete Keyframe…
Opens the Delete Keyframes pop-up dialog which allows you to specify the number of the last keyframe you wish to retain and then delete all keyframes back to that frame. The keyframe whose number you specify is not deleted. To delete all keyframes enter 0 at the prompt.

Run Animation
Click to run the keyframe animation. If you click Run more than once, the animation will play for the corresponding number of times. To abort the run, press the "a" key in the Graphics Window.

Run Attributes...
Gives the user access to the various controls of the animation, as explained in more detail below.

Speed/Actions

Opens the Speed(Subframes)/Actions(Commands) portion of the Keyframe Run attributes dialog.

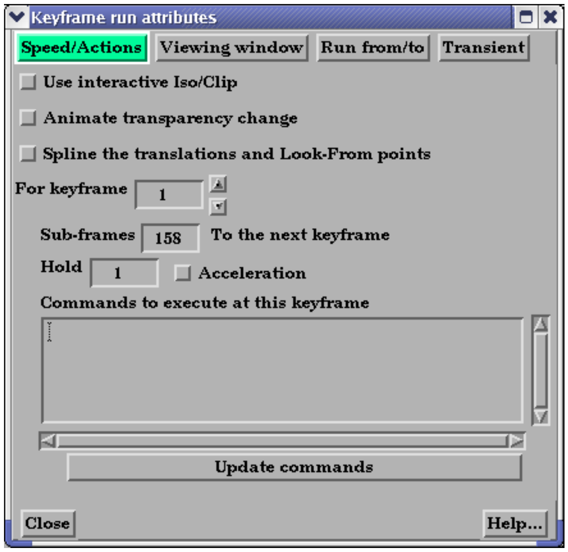


Figure 7-12
Speed(Subframes)/Actions(Commands) portion of the Keyframe Run Attributes dialog

Use Interactive Iso/Clip

By turning this toggle on, any clip or isosurface moved interactively during the keyframe will animate.

Animate Transparency Change

By turning this toggle on, transparency changes to parts during the definition of the keyframes will be part of the animation.

Spline the Translations and Look From Points

By turning this toggle on, translations and look-at/from transforms will be interpolated on a cubic spline.

For Keyframe

This field and the stepper buttons are used to select which keyframe to edit.

Sub-Frames

The Sub-Frames field specifies the number of subframes between that keyframe specified and the next one. More subframes make the transformations to the next keyframe smoother and slower.

Hold

This field specifies the number of frames to hold at the keyframe.

Acceleration

By turning on this toggle, acceleration will be used at the keyframe. If on for a previous keyframe it will decelerate.

Commands to Execute at Keyframe

This command text area is used to specify up to 20 commands to execute before displaying the keyframe referenced in the For Keyframe field (this can be used for all keyframes except the last one). You may use any command except commands corresponding to nonpermitted actions, such as loading another dataset. Also, there is no point in using `view_transf` commands that transform frames, change the Look At and Look From points, or move the Plane Tool since the next thing EnSight does is update the Graphics Window to match the transformation matrix information stored as Part of the keyframe. You may use `anim_keyframe` commands, for example, to toggle-on using transient data, but you should not use the `anim_keyframe: run` command since then the animation will enter an infinite loop. Commands frequently used here would be `view:` and `annotation:` commands. You may also play a command file (or a Python script), so there is really no limit as to how many commands you can execute. The `shell:` command is a special command to issue a UNIX command.

Update Commands

This button will accept the commands entered above.

Viewing Window — Opens the Viewing Window portion of the Keyframe Run Attributes dialog.

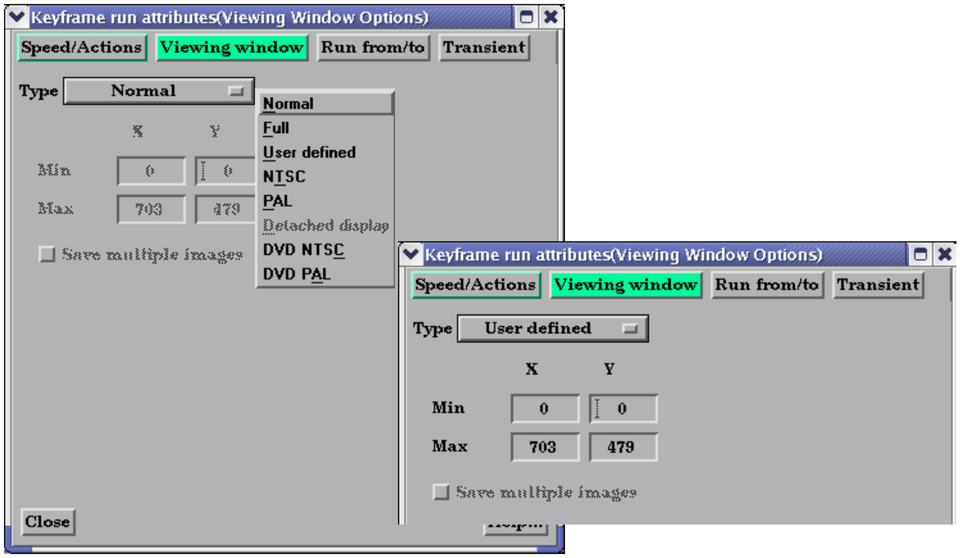


Figure 7-13
Viewing Window portion of Keyframe Run Attributes dialog

Type — Selection of image type, including standard video formats. Options are:

Normal type is appropriate for display in the Graphics Window.
Full type is appropriate for a full-screen graphics window.
User Defined type enables you to specify the screen dimensions (see below).
NTSC type is NTSC window size.
PAL type is PAL window size.
Detached Display enables the specification of the detached display as the destination of the animation rendering.
DVD NTSC type is a DVD standard window size 704 x 480 pixels.
DVD PAL type is a DVD standard window size 704 x 576 pixels.
HD720p type is an HD DVD standard window size 1280 x 720 pixels.
DVD PAL type is an HD DVD window size 1920 x 1080 pixels.

Min X Y
Max X Y — If Type: *User Defined*, these fields allow you to specify the pixel dimensions for user-defined type of screen. Allowed values for X are 0 to 1279 and for Y from 0 to 1023. Bottom left corner is 0,0. EnSight assumes a horizontal-to-vertical aspect ratio of 5-to-4. Other aspect ratios will distort the images.

Save Multiple Images — If Type: *Detached Display*, will save an image per display

Run From/To — Opens the Run From/To portion of the Keyframe Run Attributes dialog.

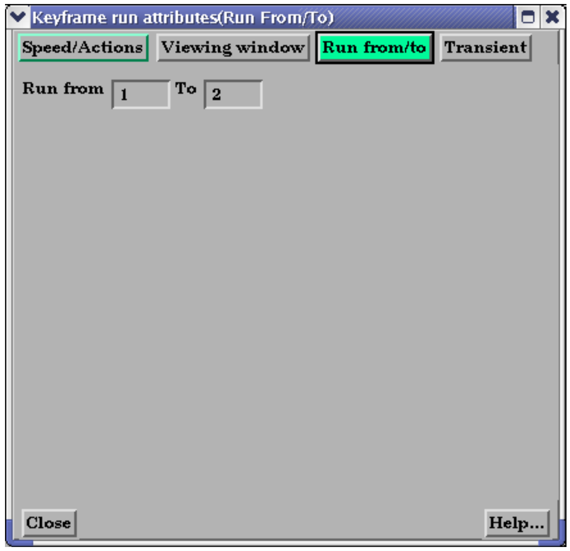


Figure 7-14
Run From/To portion of Keyframe Run Attributes dialog.

Run From / Run To — These fields specify the numbers of the keyframe to start from and the keyframe to run to when Run button is pressed. Must be integer numbers of already created keyframes. Default is Run From 1 and Run To number of keyframes you have created.

Transient — Opens the Transient data synchronization portion of the Keyframe Run Attributes dialog.

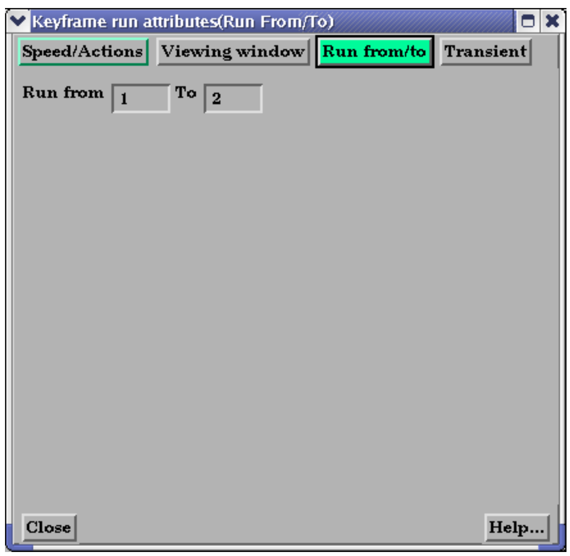


Figure 7-15
Transient Data Synchronization portion of Keyframe Run Attributes dialog.

Use Transient Data Toggle — Toggles-on/off transient data as defined in the timelines (see below).

Transient Timeline — Transient data is always used according to the definitions of the transient timelines. In a timeline you can specify the start and end time, how to increment time, etc.

By default a single timeline exists which spans the total number of keyframes. Only one timeline can exist for each keyframe.

The timeline shown is the one being edited. The up/down arrows will advance/decrement to the next/previous timeline if it exists.

New — Will create a new timeline. If the previous timeline currently spans all available keyframes a pop-up dialog will result and no new timeline will be created.

Delete Will delete the timeline indicated.

Start At Keyframe Transient data will start at the keyframe indicated.

End At Keyframe Transient data will end at the keyframe indicated.

Start Time Controls the time step to be shown at the Start Keyframe. The choices are:

Use Begin:	Use the Begin time as defined in the Solution Time dialog
Use End:	Use the End time as defined in the Solution Time dialog
Use Current:	Use the current time value
Specify:	Set the time value using the input field

End Time Controls the time step to be shown as the End Keyframe. The choices are the same as Start Time.

Specify Time Increment If off, EnSight will interpolate time such that the Start/End Time values match up with the Start/End keyframes specified. If on, you can specify the increment in time which occurs for each frame during the timeline.

When Arrive At Start/End Time If you have specified a Time increment this option controls what will happen if you arrive at the Start/End time. The choices are:

Loop	Jump to the begin/end time
Swing	Reverse direction

Quick Animations... Opens the Quick Animations dialog.

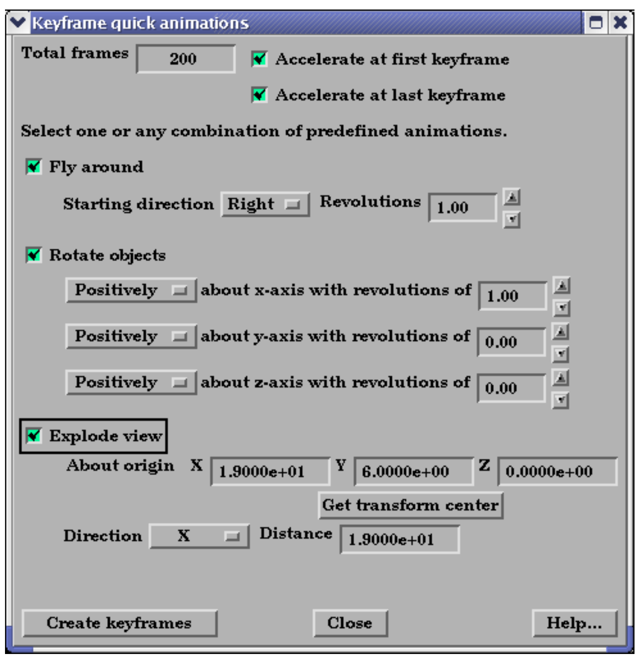


Figure 7-16
Keyframe Quick Animations dialog

This dialog allows you to add keyframes of predefined movement to your animation. Currently implemented are "fly around", "rotate objects", and "exploded view". One or a combination of these can be used.

Total Frames When the Create Keyframes button is pressed one or more keyframes will be created. The total frames (keyframes plus subframes) will be the number specified here.

Accelerate at first keyframe If on, transformations will accelerate out of the first keyframe created when Create Keyframes is pressed.

Accelerate at If on, transformations will de-accelerate at the last keyframe created when Create

last keyframe
Keyframes is pressed.

Fly Around
If on the look-from point will be revolved around the scene by the number of revolutions specified. The viewer can either rotate to the Right or the Left. The keyframes will be added when the Create Keyframes button is pressed.

Rotate Objects
If on the objects will be rotated about the x, y, and/or z axis by the number of revolutions specified. The keyframes will be added when the Create Keyframes button is pressed.

Explode View
If on the objects will be assigned local axis systems and animated Distance units in the Direction given about the origin specified. The global axis direction is used.

About Origin XYZ
The origin about which the explode will occur.

Get Transform Center
Sets the About Origin to be at the current transform center.

Direction
The direction of the explode transform. The choices are:

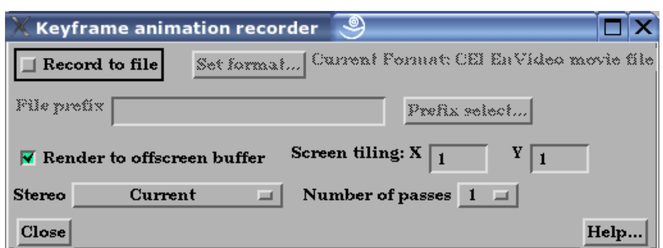
X	Translate in the x direction
Y	Translate in the y direction
Z	Translate in the z direction
XYZ	Translate in all three directions
Radial	Translate in the direction defined by a vector from the given origin through the part centroid.

For example, if the origin given is 0,0,0 and the explode direction is X, a part with a centroid at X=1 will translate in the positive X direction while a part with a centroid at X=-1 will translate in the negative X direction

Distance
The total translation to be used.

Create Keyframe
Creates the keyframe(s) given the selections in the dialog. If keyframe animation is currently not on it will turn it on and create an initial keyframe, then add the predefined transform indicated.

Record...
Opens the Keyframe Animation Recorder dialog.



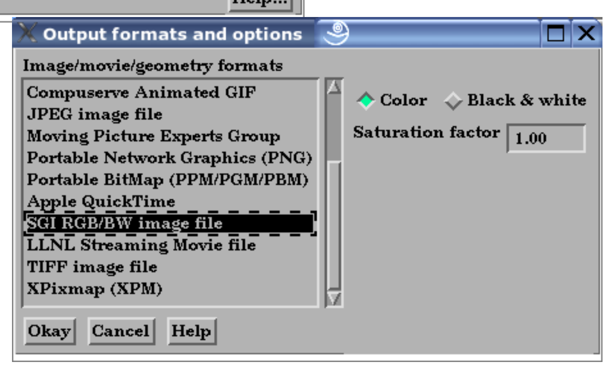


Figure 7-17
Keyframe Animation Recorder dialog

Record To File
When on, will record the keyframe animation images to the specified filename(s).

Set Format...
Brings up the Image/Movie Format and Options dialog. The flipbook will be recorded using the selected format. (see Section 2.12, Saving and Printing Graphic Images and How

	to Print/Save an Image)
File Prefix	Specification of filename prefix to use when saving frames to a disk file. Each frame is saved in a different file according to the File Numbering. The prefix can also have a directory path before it, such as /usr/tmp/prefix.
Render to offscreen buffer	If checked will render the animation to an offscreeen rendering context. If off will render to the graphics window and save from there which implies that you might also be saving popped up windows, screen savers, etc. It's highly recommended that if recording you record to the offscreen buffer.
Screen tiling	Will create multi-tile movies
Stereo	Sets the mono vs. stereo output
Current	Use mono if current graphics window is mono, use stereo interleaved if current graphics window is stereo.
Mono	Save the images as mono independent of the current graphics window state
Interleaved	Save the images as stereo interleaved independent of the current graphics window state
Anaglyph	Save the images as color separated anaglyph stereo images. Various color combinations available
Number of passes	Sets the amount of multi-pass anti-aliasing to use. The higher the number, the better the quality of the image.
Save…	Opens the File Selection dialog to allow you to save the specifications of the current keyframe into a file. This saves only the keyframe specifications, not the animation images or Part information. If you perform a Full Backup, the keyframe specifications are saved as Part of the Backup.
Restore…	Opens the File Selection dialog to allow you to restore keyframe specifications from a file. This restores only the keyframe specifications; you must also load Part data and set the Part attributes.
	(see How To Create a Keyframe Animation)

Troubleshooting Keyframe Animation

Problem	Probable Causes	Solutions
Graphics Window flashes at start of animation run.	New graphics window is opened to display the animation.	Hardware specific. Does not affect frames sent to recorder.
Lines “crawl” across the screen when I play the animation.	Lines are only 1 pixel wide which would cause crawling.	Use a line width greater than 1.
During playback the action of the video starts as soon as the picture comes up and it’s hard to recognize what is happening that quickly and then it goes away.	When creating a video it is best to have the model come up with a hold of 3 seconds or more before starting the animation. The animation should run for a reasonable length of time and then it should hold for 3 or more seconds again at the end. On complex models the hold may need to be as much as 10.	Holding a video at the beginning and the end and showing enough frames in-between will allow your audiences eyes to adjust and increase comprehension of the video. Adding annotation strings and pointers to point out areas of interest also helps. Also, showing the whole model with a hold and then zooming way in on the area of interest will help comprehension.
Video is too fast when played back, but it looked fine in EnSight.	Video playback speed is independent of model complexity. Rendering speed in EnSight is more dependent on graphics hardware.	Increase the number of frames recorded by adding more subframes or use speed controls in video player.
Transformation of my object in the animation is not smooth.	Not enough subframes.	Adding more subframes will cause more finely interpolated scene between keyframes. For instance the model should probably not rotate more than 3 degrees between frames being recorded.
Model is being clipped away as the animation proceeds.	Running into the Z-Clip plane or the regular plane tool with Clipping on.	Make sure the Z-Clip planes and the plane tool are far enough away from the model for the whole animation sequence. NOTE: The distance between the Z-Clip planes could affect the clarity of the image. The Z-Clip should be kept as close to the model and as close to each other as possible for better results.

7.4 Variable Calculator

While the Variable Calculator is located on the Feature Icon Bar, it's use is discussed in Chapter 4, **Section 4.3, Variable Creation**

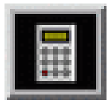


Figure 7-18
Variable Calculator Icon

7.5 Query/Plot

EnSight provides several ways to examine information about variable values. You can, of course, visualize variable values with fringes, contours, vector arrows, profiles, isosurfaces, etc. This section describes how to query variables *quantitatively*:

Over Distance

EnSight can query variables at points over distance for the following information:

- variable values inside Parts at evenly spaced points along a straight line
- variable values inside Parts at the nodes of a different 1D Part

Over Time

EnSight can query variables over time for the following information:

- minimum and maximum variable values for Parts
- variable values at any number of sample times at any point inside of a Part or at any labeled node or element.

Over-time queries can report actual variable values, or Fast Fourier Transform (FFT) spectral values at the positive FFT frequencies.

Variable vs. Variable

EnSight can produce a scatter plot of one variable vs. another.

Operations on

EnSight can scale query values and/or combine one set of query values with another set to produce a new set of values.

Importing

EnSight can import query values from external files.

Query Candidates

Only Parts with data residing on the Server host system may be queried. Thus, Parts that reside exclusively on the Client host system (i.e. contours, particle traces, profiles, vector arrows) may NOT be queried.

(see Section 3.1, Part Overview)

Clicking once on the Query/Plot Icon opens the Query/Plot Editor in the Quick Interaction Area which is used to query about the selected Variable on the selected Part and, if you wish, assign a query entity to a plotter.

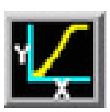


Figure 7-19
Query/Plot Icon

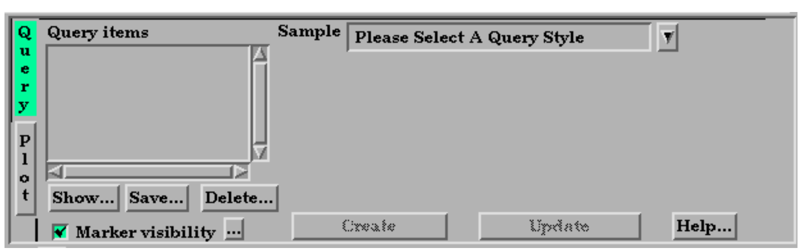


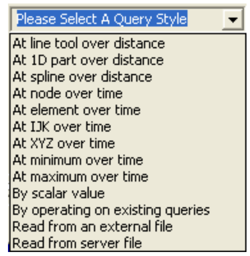
Figure 7-20
Quick Interaction Area - Query/Plot Editor

Query Items

This is the list of query items that currently exist in EnSight. After creating a query item, it will show up in this list and can be modified by selecting it in the list and changing the displayed values. (Note, it is best to deselect any query items in the list when creating a new one, otherwise you may accidently modify values for the item with the left mouse button.)

Sample This menu contains the types of queries that can be created. Selecting one of these changes the interface to display controls related to the type.

Figure 7-21
Query Sample Types



Please Select A Query Style is displayed until a Sample selection is made.

At Line Tool Over Distance queries at uniform points along the line tool.

At 1D Part Over Distance queries at the nodes of a 1D part.

At Spline Over Distance queries along an existing spline.

At Node Over Time queries at a node over a range of times.

At Element Over Time queries at an element over a range of times.

At IJK Over Time queries at an IJK location over a range of times.

At XYZ Over Time queries at the x, y, z location over a range of times.

At Minimum Over Time queries the minimum of a variable over a range of times.

At Maximum Over Time queries the maximum of a variable over a range of times.

By scalar value queries two variables at a given scalar variable value.

By Operating On Existing Queries forms new query by scaling and/or combining existing ones.

Read From An External File imports previously saved or externally generated queries. (This can be EnSight XY data format or MSC Dytran .ths files.)

Read from server file imports any queries that the server knows about.

General to Each Type of Query

Marker Visibility Toggles the visibility of the marker showing the location for the query. For distance queries, a sphere marker will be shown indicating the beginning location for the query.

... Opens the Query Display Attributes dialog for the specification of the display attributes of the query marker.

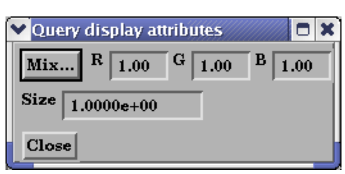


Figure 7-22
Query Marker Display Attributes dialog

Mix... Opens the Color Selector to specify the color of the marker.

RGB The red, green, and blue color for the marker.

Size The size of the sphere marker. The value is a scale factor. Values larger than 1.0 will scale the marker up, while values less than 1.0 (but greater than 0.0) will scale the marker down.

Update This button causes the query to be recomputed using any modified attributes or variables.

Save... Opens the Save Entity Query To dialog for the specification of the format, and file name in which you wish to save the query entity.

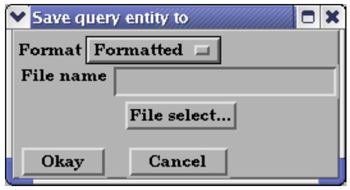


Figure 7-23
Save Query Entity To dialog

Format Opens a pop-up menu to allow specification of the format. Choices are:

Formatted Outputs the query information to the specified file in the same format as the Show Text button.

XY Data Outputs the query information in a generic format which could be used to export the information to a different plotting system.

File Name This field is used to specify the file name in which you wish to save the query entity.

File Select Opens up the File Selection dialog for specifying the File Name as an alternative to entering it manually in the File Name field.

Show... This button will display the results of the selected query in the EnSight Message Window.

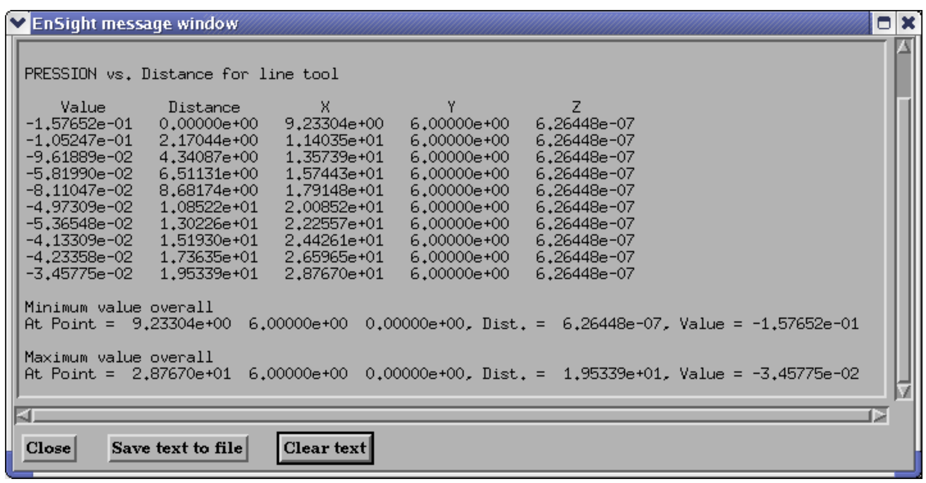


Figure 7-24
EnSight Message Window displaying query information

Save Text To File Opens the File Selection dialog for specification of filename to save to.

Delete... This button will delete the selected query items. You must confirm the deletion before it as actually done.

Create This button will create the query according to options and variables specified.

Plot

Changes the Quick Interaction Area into the plotting section.

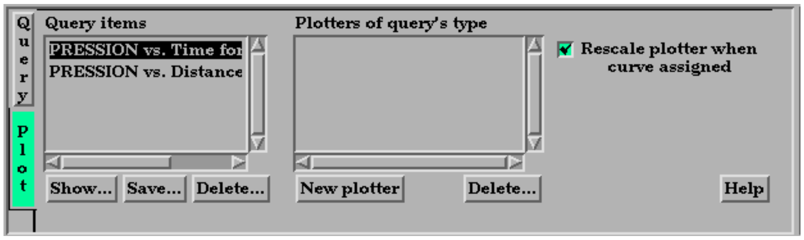


Figure 7-25
Plot Query Entity To dialog

Query Items — Shows a list of current query items. As you select these, the plotters of the same type (if any) will be displayed in the Plotters Of Query's Type list. And if the query has been assigned to one of these plotters, it will be highlighted.

Plotters Of Query's Type — Shows a list of currently defined Plotters which are of the same type as the selected query. The selected query can be plotted on these or on a new plotter. Max is 25 plotters.

Rescale plotter — If a query is assigned to an existing plotter by selecting one in the Plotter Of Query's Type list, toggling this option will rescale the axis of the Plotter to include all queries that are assigned to it.

New Plotter — Will create a new plotter, add the new plotter to the list, and display the plotter with its query curve in the graphics window.

Delete... — Will delete the selected plotters. You must confirm the deletion before it is actually done.

Query

At Line Tool Over Distance

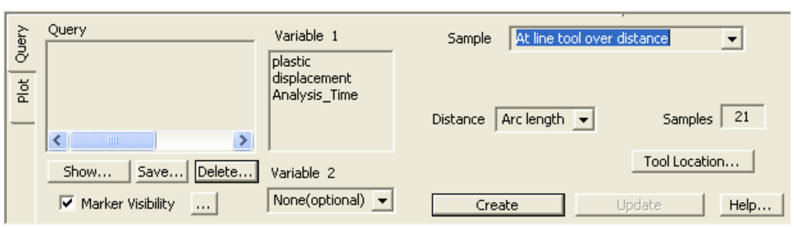


Figure 7-26
Quick Interaction Area - Query/Plot Editor - **At Line Tool Over Distance**

Variable: 1 — A list of variables that can be chosen for the query. Choose one variable. If plotted, this variable will be plotted along the Y-Axis.

Variable: 2 — If you leave this as "None", DISTANCE will be the default X- Axis variable. If you choose a variable from the list, a "scatter plot" query will result, and the X-Axis will be the variable you have chosen.

Distance — A menu of choices that control the distance parameter.

Arc Length	The distance along the part from the first node to each subsequent node (i.e. the sum of the 1D element lengths).
X Arc Length	The X coordinate value of each node accumulated from the start.
Y Arc Length	The Y coordinate value of each node accumulated from the start
Z Arc Length	The Z coordinate value of each node accumulated from the start.
From Origin	The distance from the origin.
X from Origin	The X distance from the origin.
Y from Origin	The Y distance from the origin.
Z from Origin	The Z distance from the origin.

Samples — For queries over Distance using the Line Tool, this field specifies the number of equally spaced points to query along the line.

Tool Loc... — Brings up the Transformation Editor (Line Tool) dialog for feedback and manipulation of the location of the line tool.

At 1D Part Over Distance

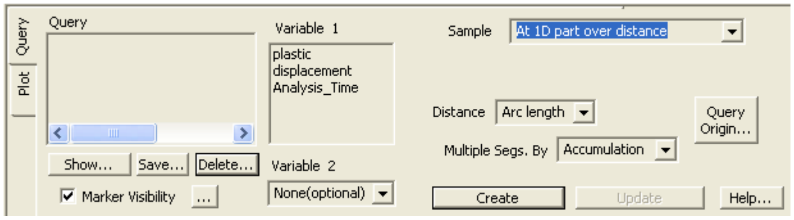


Figure 7-27
Quick Interaction Area - Query/Plot Editor - **At 1D Part Over Distance**

Note that the 1D part to use for the query must be selected from the Part's list.

Variable: 1 — A list of variables that can be chosen for the query. Choose one variable. If plotted, this variable will be plotted along the Y-Axis.

Variable: 2 — If you leave this as "None", DISTANCE will be the default X- Axis variable. If you choose a variable form the list, a "scatter plot" query will result, and the X-Axis will be the variable you have chosen.

Distance — A menu of choices that control the distance parameter.

Arc Length	The distance along the part from the first node to each subsequent node (i.e. the sum of the 1D element lengths).
X Arc Length	The X coordinate value of each node accumulated from the start.
Y Arc Length	The Y coordinate value of each node accumulated from the start
Z Arc Length	The Z coordinate value of each node accumulated from the start.
From Origin	The distance from the origin.
X from Origin	The X distance from the origin.
Y from Origin	The Y distance from the origin.
Z from Origin	The Z distance from the origin.

Multiple Segments By — When the selected 1D part contains more than one contiguous segment, these are handled by:

Accumulation	Each segment's query is appended to the previous. Thus a plot of this query will be one extended curve, but the extents of individual segment may not be obvious.
Reset Each	Each segment's query is treated like it is independent. Thus a plot of this query will appear as several curves.

Query Origin... — Brings up the Query Origin dialog for feedback and manipulation of the location of the query origin.

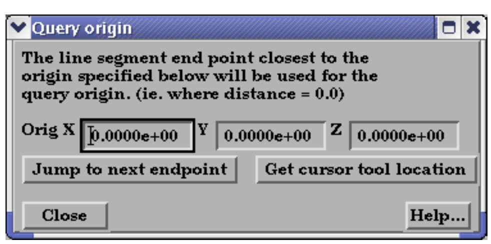


Figure 7-28
Query Origin dialog

Orig XYZ — Coordinates of the location to use for query origin determination. The endpoint closest to the origin specified will be used as the "origin" of the query, i.e., where distance is zero. If the 1D part is s closed loop (i.e. there are no end points), the closest point on the loop is used as the "origin".

Jump To Next Endpoint — When multiple segments are present, clicking this button jumps to the beginning of the next segment, placing that location in to the Orig XYZ fields.

Get Cursor Tool Location — Places the current cursor tool location into the Orig XYZ fields so that point can be used as the query origin.

At Spline Over Distance

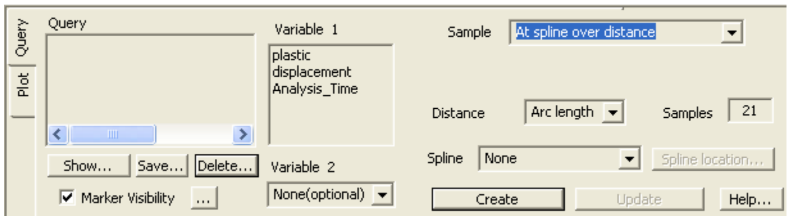


Figure 7-29
Quick Interaction Area - Query/Plot Editor - **At Spline Over Distance**

Note that the 1D part to use for the query must be selected from the Part's list.

Variable: 1 — A list of variables that can be chosen for the query. Choose one variable. If plotted, this variable will be plotted along the Y-Axis.

Variable: 2 — If you leave this as "None", DISTANCE will be the default X- Axis variable. If you choose a variable form the list, a "scatter plot" query will result, and the X-Axis will be the variable you have chosen.

Distance — A menu of choices that control the distance parameter.

Arc Length	The distance along the part from the first node to each subsequent node (i.e. the sum of the 1D element lengths).
X Arc Length	The X coordinate value of each node accumulated from the start.
Y Arc Length	The Y coordinate value of each node accumulated from the start
Z Arc Length	The Z coordinate value of each node accumulated from the start.
From Origin	The distance from the origin.
X from Origin	The X distance from the origin.
Y from Origin	The Y distance from the origin.
Z from Origin	The Z distance from the origin.

Spline — Pick a Spline. See Use the Spline Tool for more information on Spline Tools.

At Node Over Time

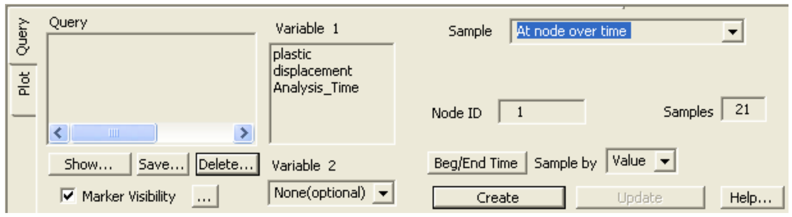


Figure 7-30
Quick Interaction Area - Query/Plot Editor - **At Node Over Time**

Variable: 1 A list of variables that can be chosen for the query. Pick one variable. If plotted, this variable will be plotted along the Y-Axis.

Variable: 2 If you leave this as “None”, TIME will be the default X- Axis variable. If you choose a variable form the list, a “scatter plot” query will result, and the X-Axis will be the variable you have chosen.

Samples Specifies how many evenly timed moments over the specified range of time steps at which to query (if left blank, you get a sample point at each time step). If you specify more or fewer sample points than the number of time steps, EnSight linearly interpolates between the adjoining time steps. If query is an FFT sampling, the number of frequencies output will be less than or equal to the number of sample points.

Node ID Specifies a node ID.

Beg/End Time... Opens up the Solution Time Editor in the Quick Interaction Area. Here you can specify the start and end times for queries Over Time.
(see Section 7.2, Flipbook Animation)

Sample By Opens a pop-up menu for specification of how to report values for Over Time queries. Options are:

Value reports values versus time.

FFT reports FFT spectral values versus FFT positive frequencies.

At Element Over Time

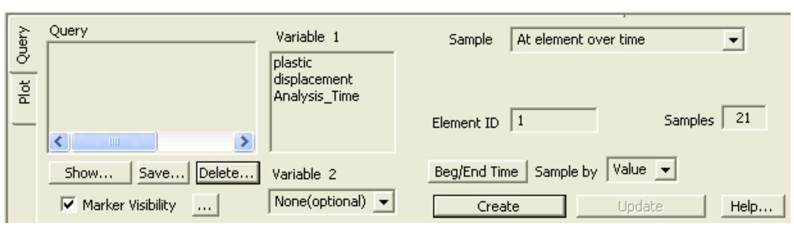


Figure 7-31
Quick Interaction Area - Query/Plot Editor - **At Element Over Time**

Variable: 1 A list of variables that can be chosen for the query. If plotted, this variable will be plotted along the Y-Axis. (Note: only per_element variables can be used for this query type.)

Variable: 2 If you leave this as “None”, TIME will be the default X- Axis variable. If you choose a variable form the list, a “scatter plot” query will result, and the X-Axis will be the variable you have chosen.

Samples Specifies how many evenly timed moments over the specified range of time steps at which to query (if left blank, you get a sample point at each time step). If you specify more or fewer sample points than the number of time steps, EnSight linearly interpolates between the adjoining time steps. If query is an FFT sampling, the number of frequencies output will be less than or equal to the number of sample points.

Element ID Specifies an element ID.

Beg/End Time... Opens up the Solution Time Editor in the Quick Interaction Area. Here you can specify the start and end times for queries Over Time.
(see Section 7.2, Flipbook Animation)

Sample By Opens a pop-up menu for specification of how to report values for Over Time queries. Options are:

Value reports values versus time.

FFT reports FFT spectral values versus FFT positive frequencies.

At IJK Over Time

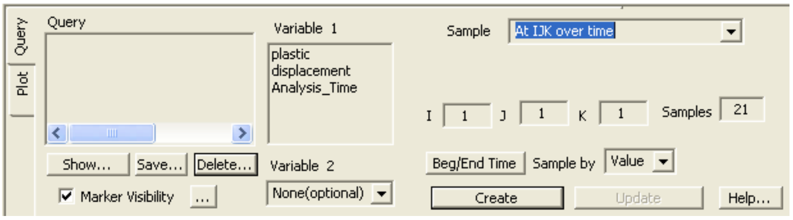


Figure 7-32
Quick Interaction Area - Query/Plot Editor - **At IJK Over Time**

Variable: 1 A list of variables that can be chosen for the query. If plotted, this variable will be plotted along the Y-Axis.

Variable: 2 If you leave this as “None”, TIME will be the default X- Axis variable. If you choose a variable form the list, a “scatter plot” query will result, and the X-Axis will be the variable you have chosen.

Samples Specifies how many evenly timed moments over the specified range of time steps at which to query (if left blank, you get a sample point at each time step). If you specify more or fewer sample points than the number of time steps, EnSight linearly interpolates between the adjoining time steps. If query is an FFT sampling, the number of frequencies output will be less than or equal to the number of sample points.

IJK Specifies the IJK planes of the desired location.

Beg/End Time... Opens up the Solution Time Editor in the Quick Interaction Area. Here you can specify the start and end times for queries Over Time.
(see Section 7.2, Flipbook Animation)

Sample By Opens a pop-up menu for specification of how to report values for Over Time queries. Options are:

Value reports values versus time.

FFT reports FFT spectral values versus FFT positive frequencies.

At XYZ Over Time

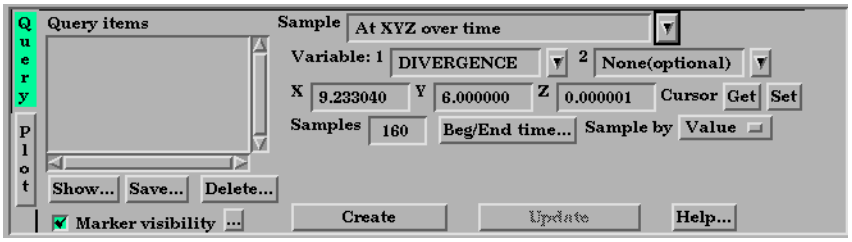


Figure 7-33
Quick Interaction Area - Query/Plot Editor - **At Cursor Over Time**

Variable: 1 A list of variables that can be chosen for the query. Choose one variable. If plotted, this variable will be plotted along the Y-Axis.

Variable: 2 If you leave this as "None", TIME will be the default X- Axis variable. If you choose a variable form the list, a "scatter plot" query will result, and the X-Axis will be the variable you have chosen.

Samples Specifies how many evenly timed moments over the specified range of time steps at which to query (if left blank, you get a sample point at each time step). If you specify more or fewer sample points than the number of time steps, EnSight linearly interpolates between the adjoining time steps. If query is an FFT sampling, the number of frequencies output will be less than or equal to the number of sample points.

Get Set Get will put the current coordinates of the cursor tool into the x, y and z fields. Set will open up the Transformation Editor (Cursor Tool) dialog for specificationof the cursor location. You can of course also set this location using interactive or picking methods.

Beg/End Time... Opens up the Solution Time Editor in the Quick Interaction Area. Here you can specify the start and end times for queries Over Time.
(see Section 7.2, Flipbook Animation)

Sample By Opens a pop-up menu for specification of how to report values for Over Time queries. Options are:

Value reports values versus time.

FFT reports FFT spectral values versus FFT positive frequencies.

At Minimum Over Time

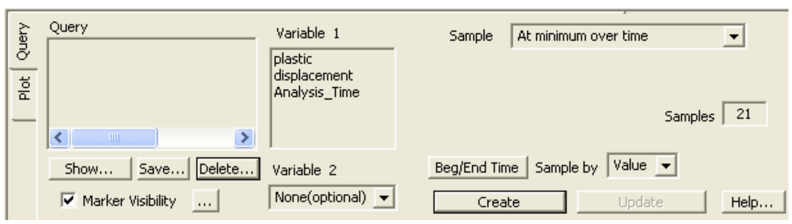


Figure 7-34
Quick Interaction Area - Query/Plot Editor - **At Minimum Over Time**

Variable: 1 A list of variables that can be chosen for the query. If plotted, this variable will be plotted along the Y-Axis. Note that you can choose more than one variable in the list which is a time-saving feature producing multiple queries at once as EnSight marches through time.

Variable: 2 If you leave this as "None", TIME will be the default X- Axis variable. If you choose a variable form the list, a "scatter plot" query will result, and the X-Axis will be the variable you have chosen.

Samples Specifies how many evenly timed moments over the specified range of time steps at which to query (if left blank, you get a sample point at each time step). If you specify more or fewer sample points than the number of time steps, EnSight linearly interpolates between the adjoining time steps. If query is an FFT sampling, the number of frequencies output will be less than or equal to the number of sample points.

Beg/End Time... Opens up the Solution Time Editor in the Quick Interaction Area. Here you can specify the start and end times for queries Over Time.
(see Section 7.2, Flipbook Animation)

Sample By Opens a pop-up menu for specification of how to report values for Over Time queries. Options are:

Value reports values versus time.

FFT reports FFT spectral values versus FFT positive frequencies.

At Maximum Over Time

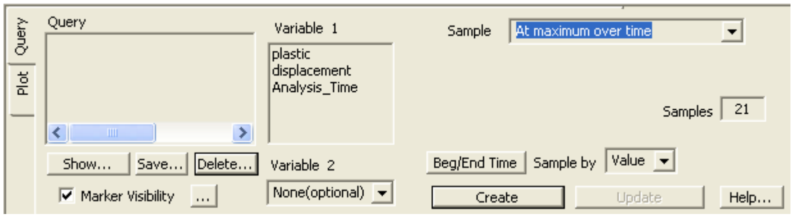


Figure 7-35
Quick Interaction Area - Query/Plot Editor - **At Maximum Over Time**

Variable: 1 A list of variables that can be chosen for the query. If plotted, this variable will be plotted along the Y-Axis. Note that you can choose more than one variable in the list which is a time-saving feature producing multiple queries at once as EnSight marches through time.

Variable: 2 If you leave this as “None”, TIME will be the default X- Axis variable. If you choose a variable form the list, a “scatter plot” query will result, and the X-Axis will be the variable you have chosen.

Samples Specifies how many evenly timed moments over the specified range of time steps at which to query (if left blank, you get a sample point at each time step). If you specify more or fewer sample points than the number of time steps, EnSight linearly interpolates between the adjoining time steps. If query is an FFT sampling, the number of frequencies output will be less than or equal to the number of sample points.

Beg/End Time... Opens up the Solution Time Editor in the Quick Interaction Area. Here you can specify the start and end times for queries Over Time.
(see Section 7.2, Flipbook Animation)

Sample By Opens a pop-up menu for specification of how to report values for Over Time queries. Options are:

Value reports values versus time.

FFT reports FFT spectral values versus FFT positive frequencies.

By scalar value

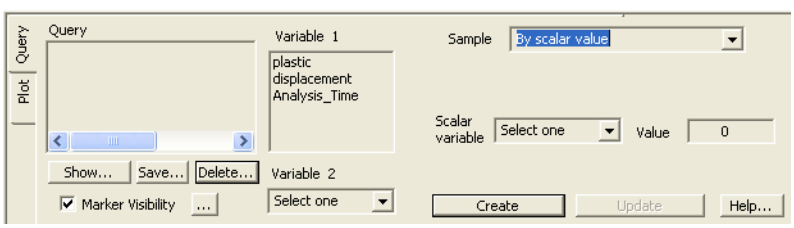


Figure 7-36
Quick Interaction Area - Query/Plot Editor - **By scalar value**

Variable: 1 A list of variables that can be chosen for the query. Choose one variable. If plotted, this variable will be plotted along the Y-Axis.

Variable: 2 A list of variables that can be chosen for the query. If plotted, this variable will be plotted along the X-Axis.

Scalar variable The scalar variable to use.

Value The value of the Scalar variable to use.

By Operating On Existing Queries

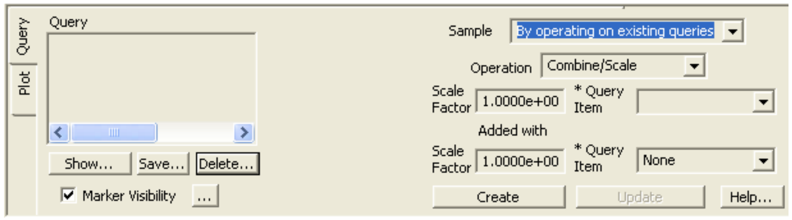


Figure 7-37
Quick Interaction Area - Query/Plot Editor - **By Operating On Existing Queries**

Scale Factor — Scale factor for the Query Item selected. The values of the selected query will be multiplied by this factor either before it is added to the second query or before the new query is created (if only operating on a single query).

Query Item — The existing query item(s) to operate on. A new query will be create consisting of scaled values one query, or the scaled, algebraic sum of two queries.

Read From An External File

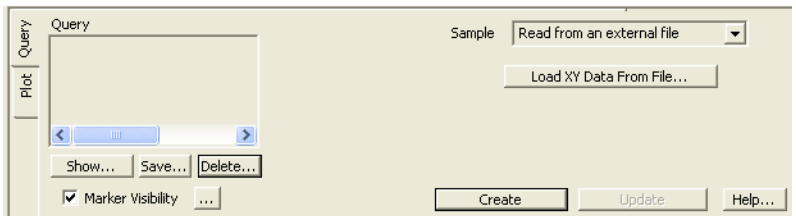


Figure 7-38
Quick Interaction Area - Query/Plot Editor - **Read From An External File**

Load XY Data From File — Opens the File Selection dialog from which a previously saved or externally generated query can be retrieved. EnSight's XY data format or MSC Dytran .ths files can be read.

Read from server file

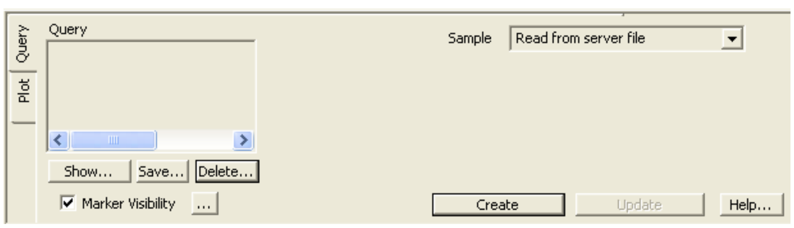


Figure 7-39
Quick Interaction Area - Query/Plot Editor - **Read from server file**

When create is hit, any avaiable queries that the server knows about will be placed in the Query items list.

(See also How To Query/Plot)

7.6 Interactive Probe Query

EnSight enables you to obtain scalar, vector, or coordinate information for the model at a point directly under the mouse pointer, at the location of the cursor tool, or at particular node, element, ijk, or xyz locations. The information is normally displayed in the Interactive Probe Query section of the Quick Interaction Area, but it can also be displayed in the Graphics Window. The performance of Interactive Query operations is dependent on the refresh time of the Graphics Window. Interactive query values are not echoed to EnSight command language files.

Clicking once on the Interactive Probe Query Icon opens the Interactive Probe Query Editor in the Quick Interaction Area which is used to specify parameters for querying interactively.

Figure 7-40
Interactive Probe Query Icon

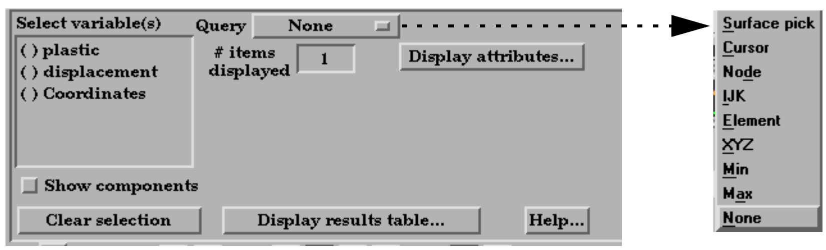


Figure 7-41
Quick Interaction Area - Interactive Probe Query Editor

Select Variable(s) — List of variables and their components (if vector and Show Components is toggles on).

Query — Selection of whether interactive query is on, or which method to use to indicate input.

Surface Pick	will query the location under the mouse in the Main View. The query will be performed when the “p” keyboard key is pressed (when “Pick Use ‘p’ ” is on) or whenever the mouse moves to a new location in the Main View (when “Continuous” is on).
Cursor	will query the location indicated by the Cursor Tool in the Main View. The query will be performed when the “p” keyboard key is pressed (when “Pick Use ‘p’ ” is on) or whenever the Cursor Tool moves to a new location in the Main View (when “Continuous” is on. You can do a quick query by right-clicking on the cursor and choose ‘Query Variable over Time’.
Node	will query the node as specified in the “Node ID” field.
IJK	will query the IJK node as specified in the “I J K” fields.
Element	will query the element as specified in the “Element ID” field.
XYZ	will query the x, y, z location as specified in the “x y z” fields.
Min	will query the min value.
Max	will query the max value.
None	indicates that interactive query is off.

Search — Selects the location for the query. (Only active for Surface Pick and Cursor queries.)

- *Exact* — indicates that the query will occur at the location of the mouse.
- *Closest Node* — indicates that the query will “snap” to the node closest to the mouse.

Pick (Use ‘p’) — When the Action is Surface Pick or Cursor, controls whether the query will occur on a keyboard ‘p’ key press (when on) or will occur continuously - tracking the mouse location.

Node ID — For Node Queries, specify the node id.

Element ID — For Element Queries, specify the element id.

 # Items Displayed — Sets the number of query locations that are kept in memory and displayed to the user.

Clear Selection — Clears all the selected variables.

Display Results Table... — Opens the Interactive Probe Query Results Table dialog which shows a table of all selected variables as well as the current query type, the latest xyz coordinates, the latest ijk values (if applicable), and the latest node or element id (if applicable). Note that the contents of this dialog can be saved to a file by using the Save... button.

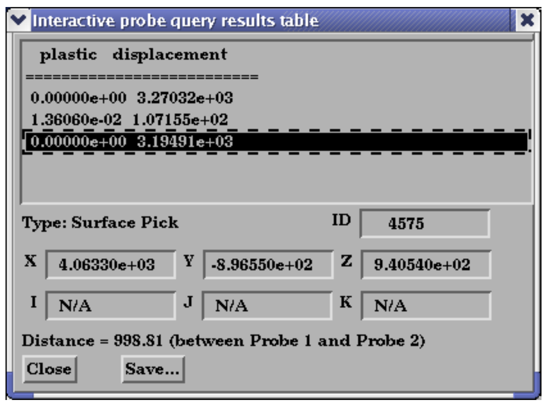


Figure 7-42
Interactive Probe Query Results Table

Display Attributes... — Opens the Interactive Probe Query Display Attributes dialog.

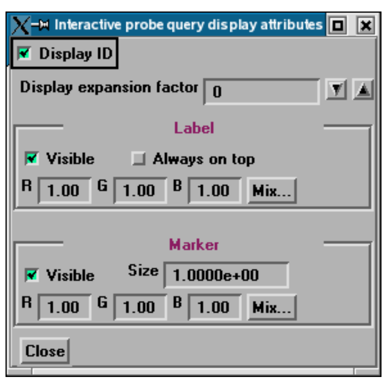


Figure 7-43
Interactive Probe Query Display Attributes dialog

Display ID Toggle — When toggled on, if an ID is appropriate for the type of search, will display the ID in the query table and in the label on the model.

Display expansion factor — **If the model has element labels**, this extracts the elements contained by the query locations and creates a subset part from these elements. The factor is how many layers of elements will be extracted, i.e., 1 is extract the element that contains the query while 2 means take the results from the elements extracted at level 1 and find all of the neighbors to these elements. If the interactive Query toggle is turned to None and the subset part has been created via the expansion factor mechanism you will be prompted if you wish to save the subset part.

Label

Visible Toggle — When toggled on, query information will be displayed in the Graphics Window

Always on Top	When on, query information in the Graphics Window will not be hidden from view behind other geometry.
RGB	These fields specify color values for the Labels.
Mix	Opens the Color Selector dialog (see Section 8.1, Color Selector)

Marker

Visible	When on, query location markers will be displayed in the Graphics Window.
Size	The size of the markers.
RGB	These fields specify color values for the markers.
Mix	Opens the Color Selector dialog (see Section 8.1, Color Selector)

7.7 Contour Create/Update

Contours are lines that trace out constant values of a variable across the surface(s) of selected Part(s), just like contour lines on a topographical map.

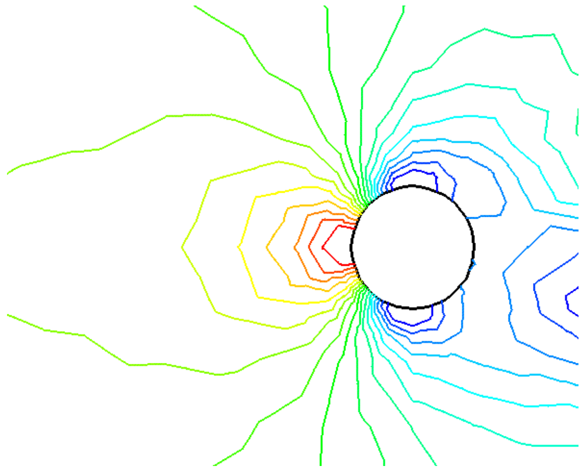


Figure 7-44
Pressure Contours in a Flow Field around a Circular Obstruction

The variable must be a scalar or vector variable. If it is a vector, the magnitude will be used by default, but one component can be used as well. A Contour Part can consist of one contour line, or a set of lines corresponding to the value-levels of the variable palette. A Contour Part has its own attributes independent of those Parts used to create it (the parent Part(s)).

Contours are drawn across the faces of parent Part elements (one-dimensional elements are ignored). At each node along the edges of any one element face, the contour's variable has a value. If the range of these values includes the contour's value-level, the contour line crosses the face. EnSight draws the contour by dividing the face into triangles each having the face's centroid as one vertex. For each triangle the contour crosses, it will cross only two sides. EnSight interpolates to find the point on each of those two sides where the variable value equals the contour value-level, then creates a bar element to connect the two points. Note that a contour line can bend while crossing an element face.

Because Contour Parts are created on the EnSight Client, the Representation attribute of the parent Part(s) greatly affects the result. Representations that reduce Part elements to one-dimensional representations (Border applied to two-dimensional Parts and Feature Angle), or do not download the Part (Not Loaded), will eliminate those Part elements from the Contour creation process. On the other hand, Full representation of three-dimensional elements will create contour lines across hidden surfaces. Usually, you will want the Representation selection to be 3D Border, 2D Full.

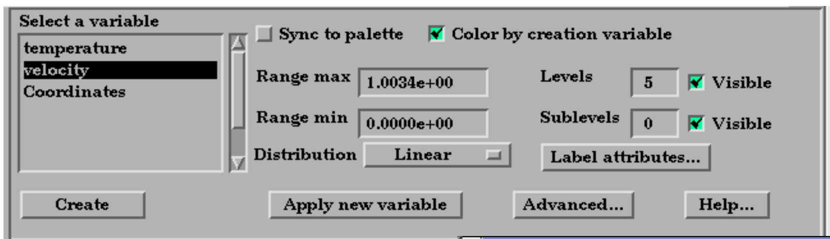
Contour Parts are created on the Client, and so cannot be queried or used in creating new variables. However, Contours can be used as parent Parts for Profiles and Vector Arrows.

If you change the value-levels in the Feature Detail Editor (Variables) Summary and Palette section, the Contour automatically regenerates using the new value-levels.

Use care when simultaneously displaying contours based on different function palettes so that you do not become confused as to which contours are which. Coloring them differently and adding an on-screen legend can help.

Left-clicking once on the Contour Create/Update Icon opens the Contour Editor in the Quick Interaction Area which is used to both create and update (make changes to) contour Parts. Left-clicking on the contour part in the graphics window will pop up a green handle in the shape of a cross. Drag this cross left and right to interactively change the contour density. Right-clicking on the contour part will give you a number of quick options.

Figure 7-45
Contour Create/Update Icon



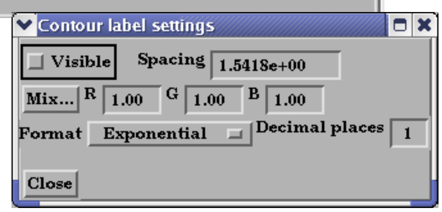


Figure 7-46
Quick Interaction Area - Contour Editor

Sync To Palette — Toggles on/off the contour line synchronization to the legend color palette.

Color by creation — If toggled on at Contour part creation, then the Contour Part is colored by the variable.

Range Min — This field is activated when Sync to Palette Toggle is Off.

Range Max — This field is activated when Sync to Palette Toggle is Off.

Distribution — This pop-up menu is activated when Sync to Palette Toggle is Off. Opens a pop-up menu for the selection of a distribution function for the contour lines. Choices include Linear, Logarithmic, and Quadratic.

Levels — This field is activated when Sync to Palette Toggle is Off. This field determines the number of contours between the Range Min and Range Max.

Visible — Toggles whether the main level contours are visible or not.

Sublevels — This field allows you to specify the number of sub-contours you wish to be drawn at evenly spaced value-levels between the value-levels defined in the Variable Feature Detail Editor Summary and Palette section. Leaving this field 0 will produce exactly the number of contour lines for which value levels are specified in the Feature Detail Editor (Variables) Summary and Palette.

Visible — Toggles whether the sublevel contours are visible or not.

Label Attributes... — Opens the Contour Label Settings dialog.

Visible Toggle — Toggles on/off the visibility of number labels for contour lines.

Spacing — Determines the spacing between number labels.

Mix... — Opens the Color Selector dialog for the assignment of a color to number labels.

R,G,B — Allows the specification of red, green, and blue values for the assignment of a color to number labels.

Format — This pop-up menu allows selection of format of number labels. Choices include Exponential, and Floating Point.

Decimal Places — This field allows the specification of the number of decimal places of the number labels.

Create — Creates a Contour Part using the selected Part(s) in the Parts List and the color palette associated with the Variable currently selected in the Main Variables List.

Apply New Variable — Will change the Contour Part to show contours based on the color palette associated with the Variable currently selected in the Variables List.

Advanced — Will open the Feature Detail Editor and select the Contours icon for more advance control of the Contour Part.

Feature Detail Editor (Contours) — Double clicking on the Contour Create/Update Icon opens the Feature Detail Editor (Contours), the Creation Attributes Section of which provides access to the same functions available in the Quick Interaction Area, as well as one more. For a detailed discussion of the remaining Feature Detail Editor turn-down sections (which are the same for all Part types):

Display offset — This field specifies the normal distance away from a surface to display the contours. A positive value moves the contours away from the surface in the direction of the surface normal. A negative value moves in the negative surface normal direction.

Please note that there is a hardware offset that will apply to contours, vector arrows, separation/attachment lines, and surface restricted particle traces that can be turned on or off in the View portion of Edit->Preferences. This preference ("Use graphics hardware to offset line objects...") is on by default and generally gives good images for everything except move/draw printing. This hardware offset differs from the display offset in that it is in the direction perpendicular to the computer screen monitor (Z-buffer).

Thus, for viewing, you may generally leave the display offset at zero. But for printing, a non-zero value may become necessary so the contours print cleanly.

(see Section 3.3, Part Editing and How to Create Contours)

Troubleshooting Contours

Problem	Probable Causes	Solutions
No contours created.	Variable values on element faces are outside range of palette function value-levels.	Adjust palette function using the Feature Detail Editor (Variables) Summary and Palette section.
	Parent Parts do not contain any 2D elements.	Re-specify Parent Part list.
	Parent Parts do not contain the specified Variable.	Recreate the Variable for the selected Parent Part(s).
Too many contours.	Palette has too many function levels.	Change the number of levels for the palette using the Feature Detail Editor (Variables) Summary and Palette.
	Specified too many sub-contours.	Lower the sub-contour attribute.
Too few contours.	The palette levels do not adequately cover the function value range for the Parent Parts.	Modify the palette using the Feature Detail Editor (Variables) Summary and Palette.
	Sub-contour attribute set to 0.	Modify the Sub-contour attribute.

Problem	Probable Causes	Solutions
Contour Part created but (empty)	Parent Part is in Feature Angle representation.	Change Parent Part to 3D border, 2D full representation.
Contours are fine at first, but later go away.	Parent Parts representation changed to Feature Angle, or Not Loaded.	The contours are created from the Part representation on the EnSight client. Modifying the representation affects the Contour Parts.
Contour parts don’t print well	See Display Offset above.	Enter a display offset (may need to be less than zero if viewed from “backside”).

7.8 Isosurface Create/Update

Isosurfaces are surfaces that follow a constant value of a variable through three-dimensional elements. Hence, isosurfaces are to three-dimensional elements what contour lines are to two-dimensional elements.

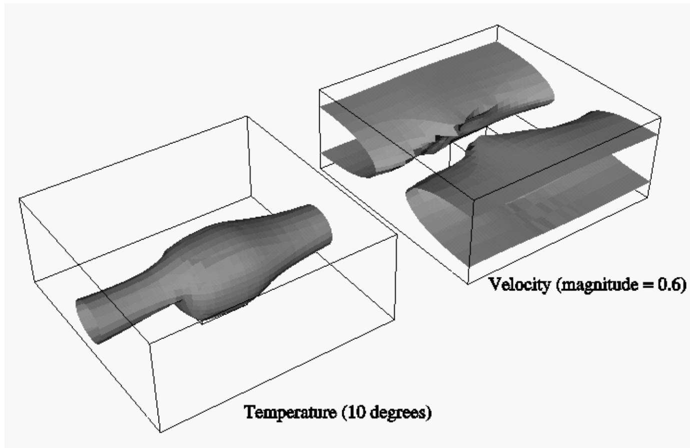


Figure 7-47
Isosurface Illustration

An isosurface may be based on a vector variable (magnitude or components), or a scalar variable.

At each node of a three-dimensional element, the isosurface's variable has a value. If the range of these values includes the isosurface's isovalue, the isosurface cuts through the element. EnSight draws the isosurface through that element by first determining which edges the isosurface crosses, and then interpolating to find the point on each of those edges corresponding to the isovalue. EnSight connects these points with triangle elements passing through the parent Part elements. If the Parent Part(s) contain two-dimensional elements, a line is created across the elements - just like a contour.

All the triangle elements created inside all the three-dimensional elements of all the parent Part(s) together with all the lines created across the two-dimensional elements of all the Parent Part(s) constitute the isosurface. One-dimensional elements of the parent Part(s) are ignored. Because isosurfaces are generated by the server, the Representation of the parent Part(s) is not important.

You can interactively manipulate the value of an isosurface with a slider allowing you to scan through the min/max range of a variable. This scanning can also be done automatically. The isosurface will change shape as the value is changed.

If you are using animation, you can specify an Animation Delta value by which the isovalue is incremented for each animation frame or page. The isosurface is automatically updated to appear as if it had been newly created at the new location and time.

Left-clicking once on the Isosurface Create/Update Icon opens the Isosurface Editor in the Quick Interaction Area which is used to both create and update (make changes to) isosurface Parts.

Left-clicking on the isosurface part in the graphics window will pop up a green handle in the shape of a cross. Drag this left and right to change the isosurface value. Right-clicking on the results in a pull-down menu of quick options for your isosurface.

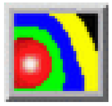


Figure 7-48
Isosurface Create/Update Icon

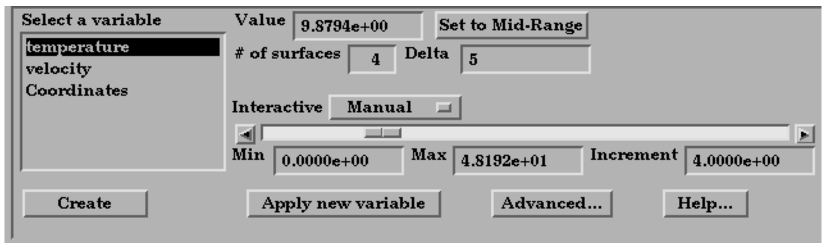


Figure 7-49
Quick Interaction Area - Isosurface Editor

Record Button (Enabled with active animation playing)

Value Specification of numerical isovalue of the isosurface. To avoid an empty Part, this value must be in the range of the Variable within the Parent Parts. You can find this range using the Variables dialog or by showing the Legend for the Variable. For vector-variable-based isosurfaces, the vector magnitude is used.

Set to Mid-Range Clicking this button will put the value that is halfway between the minimum and the maximum variable value.

# of surfaces If you want more than one isosurface calculated at a Delta offset from each other, enter the number of surfaces in this field. This number of isosurfaces a1s calculated then grouped together. This field is only available the first time the isosurface(s) are calculated. It is not possible to change this value and recalculate the isosurfaces. To change the number or the Delta, they must be deleted and recalculated.

Delta Offset value to use for creating a number of isosurfaces. The first isosurface is calculated at the number entered in Value, and the next one is Delta + Value, etc.

Interactive Opens pull-down menu for selection of type of interactive manipulation of the isosurface value. Options are:

- Off — Interactive isosurfaces are turned off.
- Manual — Value of the isosurface(s) selected are manipulated via the slider bar and the isosurface is interactively updated in the Graphics Window to the new value. For quick interactive control of the isosurface, simply left-click on the isosurface and grab the resulting green, cross-shaped click and go handle and drag left and right to see the isosurface value interactively decrease and increase respectively.
- Auto — Value of the isosurface is incremented by the Auto Delta value from the minimum range value to the maximum value when the cursor is moved into the Main View. When reaching the maximum it starts again from the minimum.
- Auto Cycle — Value of the isosurface is incremented by the Auto Increment value from the minimum range value to the maximum value. When reaching the maximum it decrements back to the minimum.

Record Button Opens a dialog for recording Interactive Auto or Interactive Auto Cycle animation to a file. This button is enabled when Auto or Auto cycle animation is actively playing.

Min Specification of the minimum isosurface value for the range used with the "Manual" slider bar and the "Auto" and "Auto Cycle" options.

Max Specification of the maximum isosurface value for the range used with the "Manual" slider and the "Auto" and "Auto Cycle" options.

Increment Specification of the increment/decrement the slider will move within the min and max, each time the stepper buttons are clicked.

Create Creates an isosurface Part at the value specified for the variable selected in the Variables List and from the Part(s) selected in the Parts List.

Apply New Variable Will recreate the isosurface Part at the value specified for the variable currently selected in the Variables List.

Advanced Will open the Feature Detail Editor and select the Isosurfaces icon for more advance control of the Isosurface Part.

Feature Detail Editor (Isosurfaces)

Double clicking on the Isosurface Create/Update Icon opens the Feature Detail Editor (Isosurfaces), the Creation Attributes Section of which provides access to additional features for isosurface creation and modification:

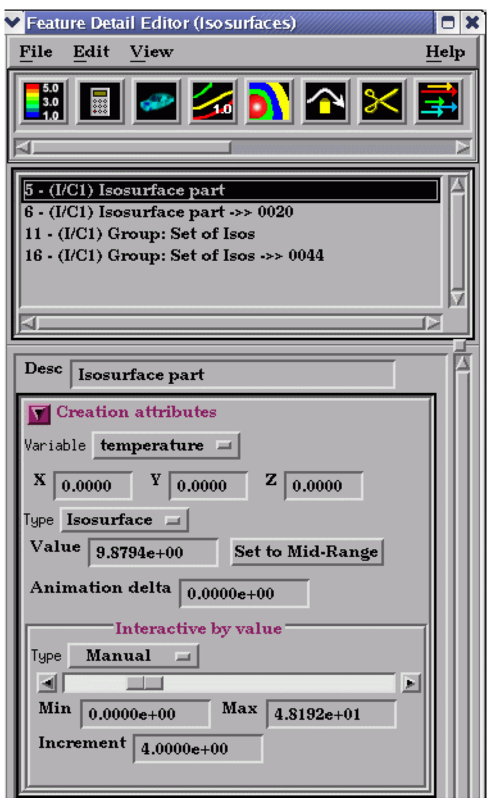


Figure 7-50
Feature Detail Editor (Isosurfaces) Creation Attributes Area

Variable Opens a pop-up menu for the selection of an active Variable to use to calculate the isosurface.

X Y Z These fields specify the vector- component coefficients. When the three fields are set to 0.0000, the vector magnitude is used for the isosurface calculation. Otherwise, the sum of:

(Vector$_x$ * X)+(Vector$_y$ *Y) + (Vector$_z$*Z) is used as the isosurface value.

Type

Isosurface — Specification that an Isosurface type part created from the specified Variable and selected parts will have the isovalue of Value for all its elements.

Value — Specification of the numerical isovalue of the Isosurface Part(s) selected in the Feature Detail Editor's Parts List (or if none is selected, of the isosurface you are about to Create)

Set to Mid-Range — Puts the value that is halfway between the minimum and the maximum variable value into the Value Field.

.

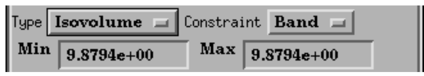


Figure 7-51
Feature Detail Editor (Isovolume) Creation Attributes Area

Isovolume — Specification that an Isovolume type part created from the specified Variable and selected parts will consist of elements with isovalues constrained to either below a Min, above a Max, or within the specified interval of Min and Max. The isosurface and isovolume algorithms are different. The isosurface algorithm defines the element interesection along the element surfaces. In contrast, the isovolume algorithm subdivides the 3D volume into tetrahedral elements and determines the intersections along the edges of each subdivided basis element resulting in more intersection points. For coarser meshes, the isosurface algorithm will be a smoother surface, but as the mesh gets finer the two algorithms should converge.

Constraint — Specification restricting the element isovalues of the Isovolume Part to an interval. The Constraint options are:

Low — all elements of Isovolume Part have isovalues below the specified Min value.

Band — all elements of Isovolume Part have isovalues within the specified Min and Max interval values.

High — all elements of Isovolume Part have isovalues above the specified Max value.

Min — Specification of the minimum isovalue limit for the Isovolume Part.

Max — Specification of the maximum isovalue limit for the Isovolume Part.

Animation Delta — This field specifies the incremental change in isovalue for each frame or page of animation. It can be negative.

(see Section 7.2, Flipbook Animation and Section 7.3, Keyframe Animation)

Create — (At the bottom of the Feature Detail Editor) Creates an Isosurface Part at the value specified for the variable selected in the Variable pop-up menu of the Creation Attributes section and from the Part(s) selected in the Main Parts List.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

For a detailed discussion of the remaining Feature Detail Editor turn-down sections (which are the same for all Parts):

(see Section 3.3, Part Editing and How To Create Isosurfaces

7.9 Clip Create/Update

A Clip is a straight line (a Clip Line), a plane (a Clip Plane), a quadric surface (cylinder, sphere, etc.), a constant x, y, or z plane, a box, or an i, j, or k plane that passes through selected model Parts (or already created Clips, Isosurface, or Developed Surface Parts). EnSight calculates the values of variables at the nodes of the Clip. Clips can be parent Parts. For example, you can create a Clip Line passing through a vector field, then create vector arrows originating from the nodes of the Clip Line. Clips are created on the server, and so are not affected by the selected Representation(s) of the parent Part(s). If you activate or create variables after creating a Clip, the Clip automatically updates to include them.

You specify the location, orientation, and size of the Clip numerically in the Transformations Editor dialog, or interactively using the Line, Plane, Box, or Quadric surface tool. If you wish, EnSight will automatically extend the size of a Clip Plane to include all the elements of the parent Part(s) that intersect the plane.

For a grid-type Clip Line, which is composed of bar elements, you specify how many evenly spaced nodes are along the line. For a grid-type Clip Plane, which is composed of rectangular elements, you specify the number of nodes in each dimension, resulting in an evenly spaced grid of nodes across the plane.

If you request a mesh-type Clip Line EnSight finds the intersection of the specified line with the selected parent Part(s) and creates bar elements that correspond to the mesh of the parent Part(s).

If you request a mesh-type Clip Plane, an xyz clip, or any of the quadric surfaces, EnSight finds the intersection of the specified plane or surface with the selected parent Part(s) and creates elements of various dimensions, sizes, and shapes that together form a cross-section of the parent Part(s). In this cross-section, three-dimensional parent Part elements result in two-dimensional Clip Plane elements, and two-dimensional parent Part elements result in one-dimensional Clip Plane elements. Note that two-dimensional parent Part elements that are coplanar with the cross-section are not included since they do not intersect the plane.

For line, XYZ, Plane, Quadric and Revolution Clips you can specify the resulting part to be all elements that intersect the specified value - resulting in a “crinkly” surface which can help analyze mesh quality.

For each Clip node on or inside an element of the selected parent Part(s), EnSight calculates the value of each variable by interpolating from the variable’s values at the surrounding nodes of the parent Part(s).

You can interactively manipulate the location of a clip Part by toggling on the Interactive Tool button. When this toggle is on, the tool used to create the clip Part will appear in the Graphics Window. Manipulation of this tool will cause the clip Part to be recreated at the new location. This feature allows you to interactively sweep a plane across your model or manipulate the size and location of the cylinder, sphere, or cone.

You can animate a Clip by specifying an Animation Delta vector that moves the Clip to a new location for each frame or page of the animation. The Clip updates to appear as if it had been newly created at the new location and time.

For structured Parts, you can sweep through the Part with any of the i, j, or k planes.

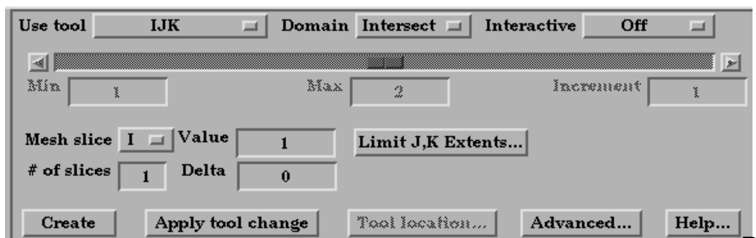
A Box Clip will create a part according to the Box Tool, and that can either be the intersection of the Box Tool walls with the selected model parts (intersect), the crinkly intersection of the Box Tool walls with the selected model parts (crinkly), the portion of the selected model parts that lie within the Box Tool (inside), or the portion of the selected model parts which lie outside the Box Tool (outside).

Clicking once on the Clip Create/Update Icon opens the Clip Editor in the Quick interaction Area which is used to both create and update clip Parts.

Figure 7-52
Clip Create/Update Icon

Use Tool

IJK The IJK clip tool is used with structured mesh results.



Record Button (Enabled with active animation playing)

Figure 7-53
Quick Interaction Area - Clip Editor - IJK tool

Domain — Specification to extract the intersection of the specified mesh slice values. For IJK clips, the only valid selection is "Intersect".

Interactive — Opens pull-down menu for selection of type of interactive manipulation of the IJK clip. Options are:

- Off — Interactive IJK clips are turned off.
- Manual — Value of the IJK clip selected are manipulated via the slider bar and the IJK clip is interactively updated in the Graphics Window to the new value.
- Auto — Value of the IJK clip is incremented by the Auto Delta value from the minimum range value to the maximum value. When reaching the maximum it starts again from the minimum.
- Auto Cycle — Value of the IJK clip is incremented by the Auto Increment value from the minimum range value to the maximum value. When reaching the maximum it decrements back to the minimum.

Slider Bar — For IJK clips, the slider bar is used to increment / decrement the Mesh Slice Value between its Minimum and Maximum value.

- Min — Specification of the minimum slice value for the range used with the "Manual" slider bar and the "Auto" and "Auto Cycle" options.
- Max — Specification of the maximum slice value for the range used with the "Manual" slider and the "Auto" and "Auto Cycle" options.
- Increment — Specification of the increment/decrement the slider will move within the min and max, each time the stepper buttons are clicked.

Mesh Slice — Opens a pull-down menu for selecting which of the IJK dimensions you wish to allow to change. You will then specify Min, Max and Step limits for the two remaining “fixed” dimensions.

Value — This field specifies the I, J, or K plane desired for the dimension selected in Mesh Slice

Limit IJK Extents — Opens the “Limits Extents of Current Slice By” dialog, in which the off dimension ranges can be limited.

IJK D(2)Min — This field specifies the minimum value for the second fixed dimension.

IJK D(2)Max — This field specifies the maximum value for the second fixed dimension.

IJK D(2) Step — This field specifies the step size through the second fixed dimension.

IJK D(3)Min — This field specifies the minimum value for the third fixed dimension.

IJK D(3)Max — This field specifies the maximum value for the third fixed dimension.

IJK D(3) Step — This field specifies the step size through the third fixed dimension.

Show Parent IJK Part Extents... — Will show the second and third dimension Min and Max extents as defined for the clip parent Part.

slices — If you want more than one clip calculated at a Delta offset from each other, enter the number of slices in this field. This number of clips is calculated then they are grouped together. This field is only available at the first time the clip(s) are calculated. It is not possible to change this value and recalculate the clips. To change the number or the Delta, they must be deleted and recalculated.

Delta — Offset value to use for creating a number of clips. The first clip is calculated at the number entered in Value, and the next one is Delta + Value, etc. and they are all grouped together in the Main Part List.

Create — Creates the Clip Part in the Graphics Window as specified.

Apply Tool Change — Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Advanced — Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.

Record Button — Opens a dialog for recording Interactive Auto or Interactive Auto Cycle animation to a file. This button is enabled when Auto or Auto cycle animation is actively playing.

Feature Detail Editor (Clips) - IJK — Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor, the Creation attributes section of which offers the same features for the IJK tool as the Quick Interaction Area Editor.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create IJK Clips)

Use Tool

XYZ — The XYZ tool is used to create a planar Part at a constant Cartesian component value that is referenced according to the local frame of the part.

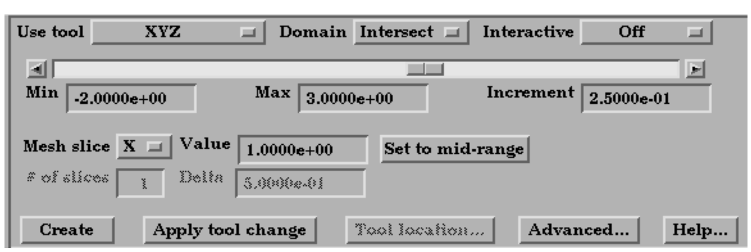


Figure 7-54
Quick Interaction Area - Clip Editor - XYZ Tool

Domain *I*

Intersect will create the cross section of the selected parts at the specified X, Y, or Z plane.

Crinkly will create a new part consisting of the parent part elements that intersect the X, Y, or Z plane

Interactive Opens pull-down menu for selection of type of interactive manipulation of the XYZ clip. Options are:

- Off — Interactive XYZ clips are turned off.
- Manual — Value of the XYZ clip selected are manipulated via the slider bar and the XYZ clip is interactively updated in the Graphics Window to the new value.

For quick interactive control of the isosurface, simply left-click on the isosurface and grab the resulting green, cross-shaped click and go handle and drag left and right to see the isosurface value interactively decrease and increase respectively.

- Auto — Value of the XYZ clip is incremented by the Auto Delta value from the minimum range value to the maximum value. When reaching the maximum it starts again from the minimum.
- Auto Cycle — Value of the XYZ clip is incremented by the Auto Increment value from the minimum range value to the maximum value. When reaching the maximum it decrements back to the minimum.

Slider Bar For XYZ clips, the slider bar is used to increment / decrement the Mesh Slice Value between its Minimum and Maximum value.

- Min — Specification of the minimum interval value of the interactive XYZ clip.
- Max — Specification of the maximum interval value of the interactive XYZ clip.
- Increment — Specification of the interval step of the interactive XYZ clip.

Mesh Slice Opens a pulldown menu for selecting which of the XYZ components you wish to clip, i.e. the X, the Y, or the Z component.

Value This field specifies the coordinate desired for the Mesh Slice component.

Set to Mid-Range Clicking this button will put the value that is halfway between the minimum and the maximum variable value.

 # slices If you want more than one clip calculated at a Delta offset from each other, enter the number of slices in this field. This number of clips is calculated then they are grouped together. This field is only available at the first time the clip(s) are calculated. It is not possible to change this value and recalculate the clips. To change the number or the Delta, they must be deleted and recalculated.

Delta Offset value to use for creating a number of clips. The first clip is calculated at the number entered in Value, and the next one is Delta + Value, etc. and they are all grouped together in the Main Part List.

Apply Tool Change Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Create Creates the Clip Part in the Graphics Window as specified.

A*dvanced* Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.

Feature Detail Editor (Clips) - XYZ

Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor (Clips), the Creation attributes section of which offers access to the same interactive clip parameters as found in the Quick Interaction Area Editor, along with additional animation delta control of clips using the XYZ tool.

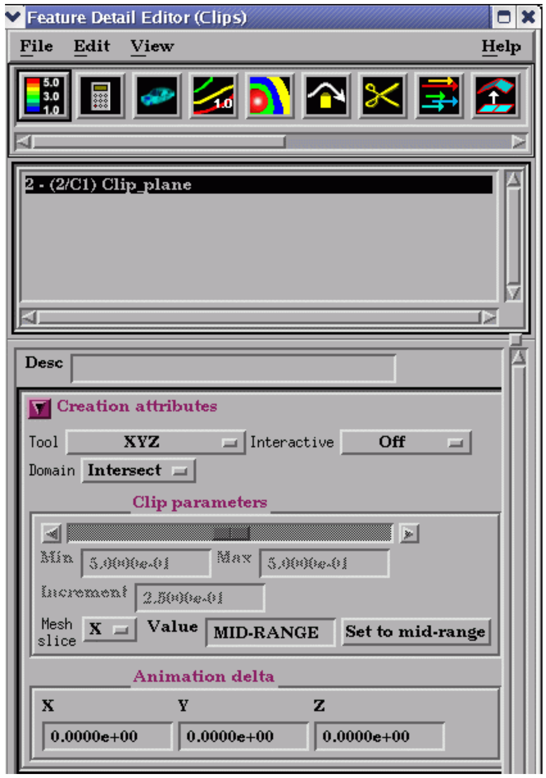


Figure 7-55
Quick Interaction Area - Clip Editor - XYZ Tool - Creation Attributes

Animation Delta

These X,Y,Z fields specify the incremental change in position of the clip for each page of Flipbook or frame of Keyframe animation.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create XYZ Clips)

Use Tool

RTZ

The RTZ tool is used to create a Part using cylindrical coordinates at a constant radius about an axis, angle around that axis or height along an axis.

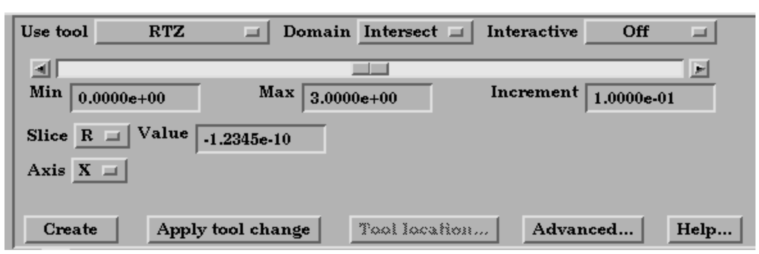


Figure 7-56
Quick Interaction Area - Clip Editor - RTZ tool

Domain

Intersect	Will create a cross section of the selected parts at the specified radius, angle, or distance along the axis.
Crinkly	Will create a new part consisting of the parent part elements that intersect the specified radius, angle or distance.

Interactive — Opens pull-down menu for selection of type of interactive manipulation of the RTZ clip. Options are:

Off	Interactive RTZ clips are turned off.
Manual	Value of the RTZ clip selected are manipulated via the slider bar and the RTZ clip is interactively updated in the Graphics Window to the new value.
Auto	Value of the RTZ clip is incremented by the Auto Delta value from the minimum range value to the maximum value. When reaching the maximum it starts again from the minimum.
Auto Cycle	Value of the RTZ clip is incremented by the Auto Increment value from the minimum range value to the maximum value. When reaching the maximum it decrements back to the minimum.

Slider Bar — For RTZ clips, the slider bar is used to increment / decrement the Slice Value between its Minimum and Maximum value.

Min	Specification of the minimum slice value for the range used with the “Manual” slider bar and the “Auto” and “Auto Cycle” options.
Max	Specification of the maximum slice value for the range used with the “Manual” slider and the “Auto” and “Auto Cycle” options.
Increment	Specification of the increment/decrement the slider will move within the min and max, each time the stepper buttons are clicked.

Slice	Opens a pull-down menu for selecting which of the RTZ components to clip, i.e. the radial (R), the angle theta (T) in degrees, or the distance along the longitudinal axis Z, (Z).
Value	This field specifies the magnitude desired for the Slice component, (theta in degrees).
Axis	The global axis with which to align the longitudinal (Z) RTZ axis.
Create	Creates the Clip Part in the Graphics Window as specified.
Advanced	Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.
Apply Tool Change	Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Feature Detail Editor (Clips) - RTZ

Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor, the Creation attributes section of which offers the same features for the RTZ tool as the Quick Interaction Area Editor.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create RTZ Clips)

Use Tool

Line The Line tool is used to create a clip line.

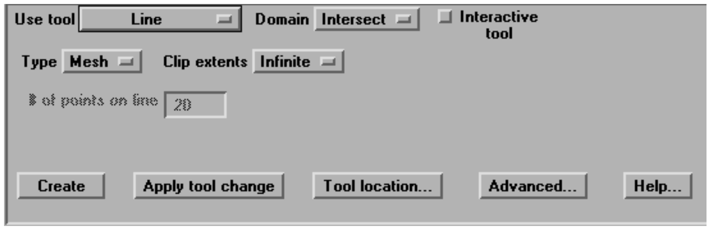


Figure 7-57
Quick Interaction Area - Clip Editor - Line Tool

Domain — Specification to extract the intersection of the line tool with the selected part(s). For Line clips, the only valid selections are "Intersect" and "Crinkly".

Interactive Tool — Toggles on/off interactive movement and updating of a clip Part. When toggled on, the line tool used to create the 2D clip line will appear in the Graphics Window. Movement of the tool will cause the Clip Part to be recreated at the new position. When manipulation of the tool stops, the clip Part and any Parts that are dependent on it will be updated. During movement, the Tool itself will not be visible, so as not to obscure the Line Clip Part. The Tool will reappear when the mouse button is released.

Type

Mesh — Will create a Line Clip showing the intersection of the line tool with the mesh elements of the parent Part.

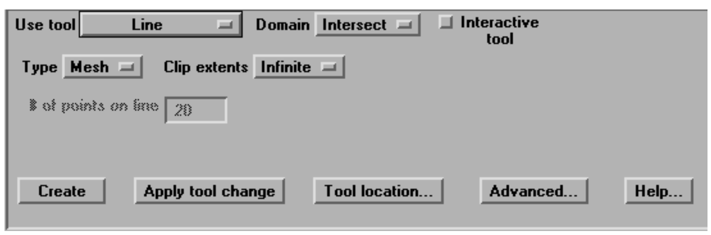


Figure 7-58
Quick Interaction Area - Clip Editor - Line Tool - Mesh Type

Clip Extents — Opens a pull down menu for selection of the extent of the Line Clip.
Finite limits the Line Clip to the length specified by the Line Tool endpoints.
Infinite Assumes the line tool defines an infinite line and uses this to intersect the elements of the selected model Parts.

Grid — Will create a Line Clip of evenly spaced bar elements along the line tool.

Figure 7-59
Quick Interaction Area - Clip Editor - Line Tool - Grid Type

# of Points on Line — Specification of number of evenly spaced points on the line at which to create a node.

Tool Location... — Opens the Transformation Editor dialog to permit precise positioning of the Line Tool within the Graphics Window. (see Tool Positions... Line Tool in Section 6.5, Tools Menu Functions and How To Use the Line Tool)

Apply Tool Change — Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Create — Creates the Clip Part in the Graphics Window as specified.

Advanced — Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.

Feature Detail Editor (Clips) - Line

Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor (Clips), the Creation attributes section of which offers access to additional features for the creation and modification of clips using the Line tool.

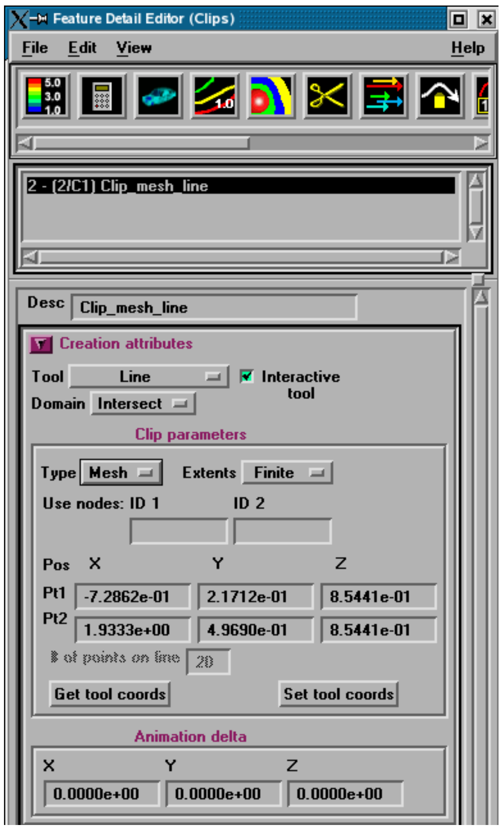


Figure 7-60
Feature Detail Editor (Clips) - Line Tool Creation Attributes

Domain — Specification to extract the intersection of the line tool with the selected part(s). For Line clips, the only valid selections are “Intersect” and “Crinkly”.

Clip Parameters

Type — Opens a pull down menu for selection of the line clip type.
Mesh - will create a Line Clip showing respecting the mesh elements of the parent Part.
Grid - will create a Line Clip of evenly spaced bar elements along the line tool.

Extents — Opens a pull down menu for selection of the extent of the Line Clip.
Finite limits the Line Clip to the length specified by the Line Tool endpoints.
Infinite extends the Line Clip to include the intersection of the line with all elements of the selected model Parts.

Use nodes — Allows for specification of the location of two nodes in the model from which to get the line clip endpoints. If this method is used, the line clip will remain tied to these nodes even if they move over time.

Pos of Pt1
Pos of Pt2 — Specification of XYZ endpoint-coordinates of Line Clip. The position of a Line Clip Part, if selected in the Feature Detail Editor’s Parts List, can be changed by manually entering values in the numeric fields and then pressing Return.

Set Tool Coords
: The position of the Line Clip tool can be changed by entering values in the numeric fields and then pressing Set Tool Coords.

Get Tool Coords
: The values in the numeric fields (and the position of a Line Clip Part, if selected in the Feature Detail Editor's Parts List) can be updated after moving the Line tool interactively in the Graphics Window by clicking Get Tool Coords. If a Line Clip Part is selected in the Feature Detail Editor Parts List, it will be repositioned to the new coordinates after clicking Get Tool Coords. Coordinates are always in the original model frame (Frame 0).

Animation Delta
: These X,Y,Z fields specify the incremental change in position of the clip for each page of Flipbook or frame of Keyframe animation.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create Line Clips)

Use Tool

Plane The Plane Tool is used to create a Plane Clip.

Domain

Intersect will create the cross section of the selected parts where they intersect the plane tool.

Crinkly will create a new part consisting of the parent part elements that intersect the plane tool.

Inside will cut the parent parts and create a new part consisting of the portion on the positive z side of the plane tool.

Outside will cut the parent parts and create a new part consisting of the portion on the negative z side of the plane tool.

In/Out will cut the parent parts and create two new parts - namely an *Inside* and *Outside* part.

Type

Mesh Will create a Plane Clip showing the cross section of the parent Part.

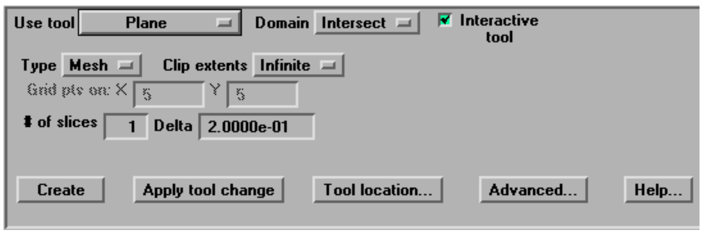


Figure 7-61
Quick Interaction Area - Clip Editor - Plane Tool - Mesh Type

Clip Extents Opens a pull down menu for selection of the extent of the Plane Clip.

Finite limits the Plane Clip to the area specified by the Plane Tool corner coordinates.

Infinite extends the Plane Clip to include the intersection of the plane with all elements of the selected model Parts.

Grid Will create a Plane Clip by discrete point sampling.

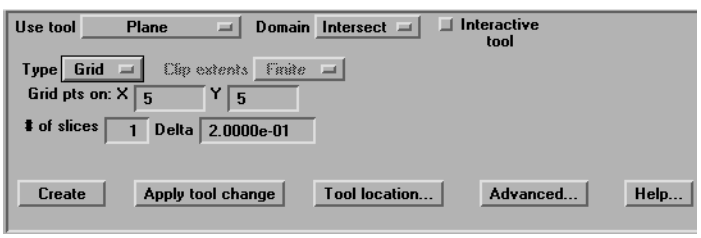


Figure 7-62
Quick Interaction Area - Clip Editor - Plane Tool - Grid Type

Grid Pts on:XY These fields specify the number of points on each edge of a Plane Clip at which to create nodes. Additional nodes are located in the interior of the plane to form an evenly spaced grid. The values must be positive integers. Applicable only to grid-type Plane Clips. Grid Pts in X correspond to the x-direction on the Plane tool, while the number of Grid Pts in Y correspond to the y-direction of the Plane tool.

slices If you want more than one clip calculated at a Delta offset from each other, enter the number of slices in this field. This number of clips is calculated then they are grouped together. This field is only available at the first time the clip(s) are calculated. It is not possible to change this value and recalculate the clips. To change the number or the Delta, they must be deleted and recalculated.

Delta Offset value to use for creating a number of clips. The first clip is calculated at the number entered in Value, and the next one is Delta + Value, etc. and they are all grouped together in the Main Part List.

Apply Tool Change — Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Interactive Tool — Toggles on/off interactive movement and updating of the clip Part. When toggled on, the Plane Tool used to create the clip Part will appear in the Graphics Window. Movement of the Plane Tool will cause the Plane Clip to be recreated at the new position. When manipulation of the tool stops, the clip Part and any Parts that are dependent on it will be updated. During movement, the Tool itself will not be visible, so as not to obscure the Line Clip Part. The Tool will reappear when the mouse button is released.

For quick interactive control of the clip plane, simply left-click on the plane tool origin and grab the resulting green, cross-shaped click and go handle and drag to see the clip location value interactively translate in the plane tool Z direction.

Tool Location... — Opens the Transformation Editor dialog to permit precise positioning of the Plane Tool. (see Section 6.5, Tools Menu Functions and How To Use the Plane Tool)

Create — Creates the Clip Part in the Graphics Window as specified.

Advanced — Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.

Feature Detail Editor (Clips) - Plane — Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor (Clips), the Creation attributes section of which offers access to additional features for the creation and modification of clips using the Plane tool.

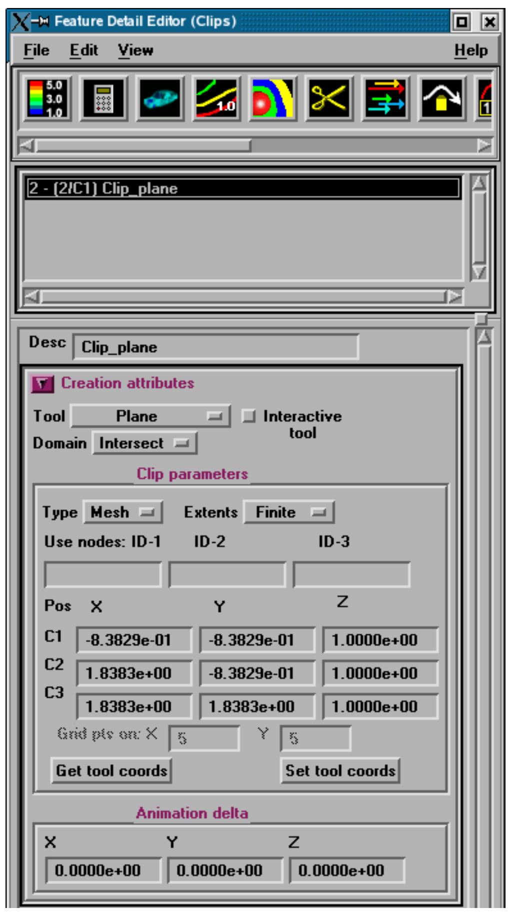


Figure 7-63
Feature Detail Editor (Clips) - Plane Tool Creation Attributes

Clip Parameters

Use Nodes:	Specification of three node ids which will be used to specify the plane of the clip. The clip plane will be tied to these three nodes, even if they move in time.
Pos of C1 Pos of C2 Pos of C3	Specification of the location, orientation, and size of the Plane Clip using the coordinates (in the Parts reference frame) of three corner points, as follows: Corner 1 is corner located in negative-X negative-Y quadrant Corner 2 is corner located in positive-X negative-Y quadrant Corner 3 is corner located in positive-X positive-Y quadrant
Set Tool Coords	Will reposition the Plane Tool to the position specified in C1, C2, and C3.
Get Tool Coords	Will update the C1, C2, and C3 fields to reflect the current position of the Plane Tool.
Animation Delta	These X,Y,Z fields specify the incremental change in position of the clip for each page of Flipbook or frame of Keyframe animation.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create Plane Clips)

Use Tool

Box — This Clipping Tool extracts portions of the model that are inside, outside, or that intersect a specified box.

Be aware that due to the algorithm used, this clip can (and most often does) have ***chamfered edges****, the size of which depends on the coarseness of the model elements.*

Domain

Intersect	will create a new part consisting of the intersection of the box tool sides and the selected parts.
Crinkly	will create a new part consisting of the parent part elements that intersect the box tool sides.
Inside	will extract the volume portion of the parent parts that lie within the box.
Outside	will extract the volume portion of the parent parts that do not lie within the box.
In/Out	will create two new parts - namely the *Inside* and *Outside* parts.

.

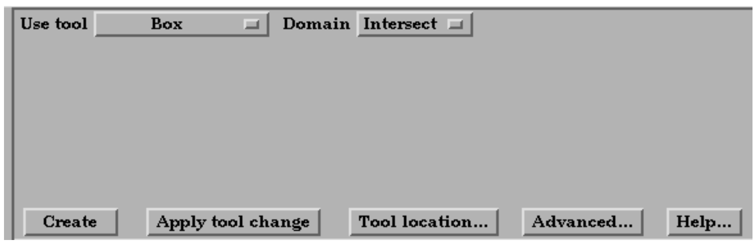


Figure 7-64
Quick Interaction Area - Clip Editor - Box Tool

Apply Tool Change — Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Tool Location... — Opens the Transformation Editor dialog to permit precise positioning of the Box Tool. (see Section 6.5, Tools Menu Functions and How To Use the Box Tool)

Advanced — Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.

Create — Creates the Clip Part in the Graphics Window as specified.

Feature Detail Editor (Clips) - Box

Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor, the Creation attributes section of which offers the same features for the Box tool as the Quick Interaction Area Editor.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create Box Clips)

Use Tool

Cylinder, Sphere, Cone These Tools are used to create a quadric clip surface.

Domain

Intersect will create the cross section of the selected parts where they intersect the quadric tool.

Crinkly will create a new part consisting of the parent part elements that intersect the quadric tool.

Inside will cut the parent parts and create a new part consisting of the portion on the inside of the quadric tool.

Outside will cut the parent parts and create a new part consisting of the portion on the outside of the quadric tool.

In/Out will cut the parent parts and create two new parts - namely an *Inside* and *Outside* part.

.

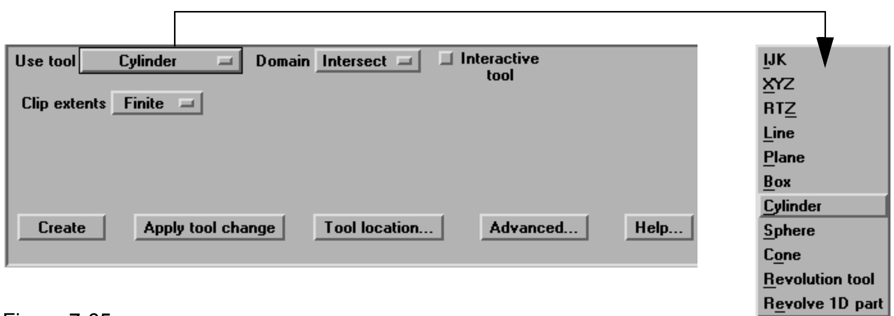


Figure 7-65
Quick Interaction Area - Clip Editor - Cylinder, Sphere, & Cone Tools

Interactive Tool — Toggles on/off interactive movement and updating of a clip Part. When toggled on, the Quadric Tool used to create the Clip Part will appear in the Graphics Window at the location of the Clip Part. Movement of the Quadric Tool will cause the Clip Part to be recreated at the new position. When manipulation of the tool stops, the Clip Part and any Parts that are dependent on it will be updated. During movement, the Tool itself will not be visible, so as not to obscure the Line Clip Part. The Tool will reappear when the mouse button is released.

Clip extents — Opens a pulldown menu that allows for the selection of Finite or Infinite extents. It is only present for cylinder and cone clips.

Advanced — Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.

Tool Location... — Opens the Transformation Editor dialog to permit precise positioning of Quadric Tools.(see Section 6.5, Tools Menu Functions and How To Use the Cylinder Tool, How To Use the Sphere Tool, and How To Use the Cone Tool)

Apply Tool Change — Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Create — Creates the Clip Part in the Graphics Window as specified.

Feature Detail Editor (Clips) Quadric Tool

Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor (Clips), the Creation attributes section of which offers access to additional features for the creation and modification of clips using the Quadric tools.

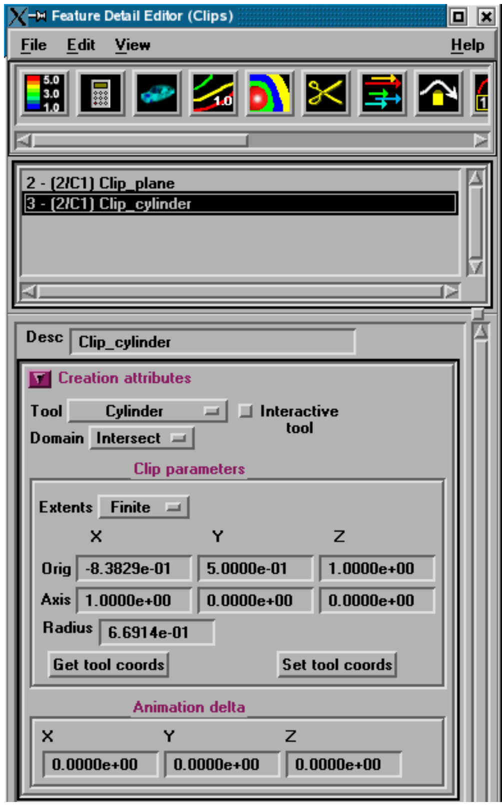


Figure 7-66
Feature Detail Editor (Clips) - Quadric Tool Creation Attributes

Clip Parameters

Cylinder

Orig XYZ — Specification of the origin (the center point) of the Cylindrical Clip.

Axis — Specification of the longitudinal axis direction of the Cylindrical Clip.

Radius — Specification of the radius of the Cylindrical Clip.

Sphere

Orig — Specification of the origin (the center point) of the Spherical Clip.

Axis — Specification of the axis direction of the Spherical Clip. *(Note: Axis is important if Developed Surface is created from the spherical clip.)*

Radius — Specification of the radius of the Spherical Clip.

Cone

Orig — Specification of the origin (the tip of the cone) of the Conical Clip.

Axis — Specification of the axis direction of the Conical Clip. Axis direction goes from tip to base.

Angle — Specification of the conical half angle (in degrees) of the Conical Clip.

Set Tool Coords — Will reposition the Quadric Tool to the position specified in the Clip Parameter fields.

Get Tool Coords — Will update the Clip Parameter fields to reflect the current position of the Quadric Tool.

Animation Delta These X,Y,Z fields specify the incremental change in position of the clip for each page of Flipbook or frame of Keyframe animation.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create Quadric Clips)

Use Tool

Revolution Tool This clipping Tool is used to create custom clip surfaces which are defined by revolving a set of lines about a defined axis.

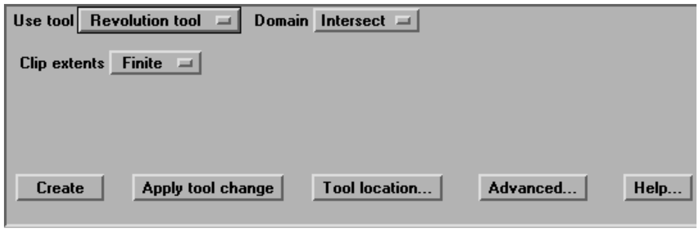


Figure 7-67
Quick Interaction Area - Clip Editor - Revolution Tool

Domain

Intersect will create the cross section of the selected parts where they intersect the revolved surface.

Crinkly will create a new part consisting of the parent part elements that intersect the revolved surface.

Inside will cut the parent parts and create a new part consisting of the portion on the inside of the revolved surface.

Outside will cut the parent parts and create a new part consisting of the portion on the outside of the revolved surface.

In/Out will cut the parent parts and create two new parts - namely an *Inside* and *Outside* part.

Tool Location... Opens the Transformation Editor dialog to permit precise location of the revolution tool within the Graphics Window. It is here where you also can control the number and positioning of the set of lines which make up the tool.
(see Section 6.5, Tools Menu Functions and How To Use the Surface of Revolution Tool)

Clip extents Opens a pulldown menu that allows for the selection of Finite or Infinite extents. It is only present for cylinder and cone clips.

Advanced Will open the Feature Detail Editor and select the Clip icon for more advance control of the Clip Part.

Apply Tool Change Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Create Creates the Clip Part in the Graphics Window as specified.

Feature Detail Editor (Clips) - Revolution Tool

Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor (Clips), the Creation attributes section of which offers access to additional features for the creation and modification of clips using the Revolution tool.

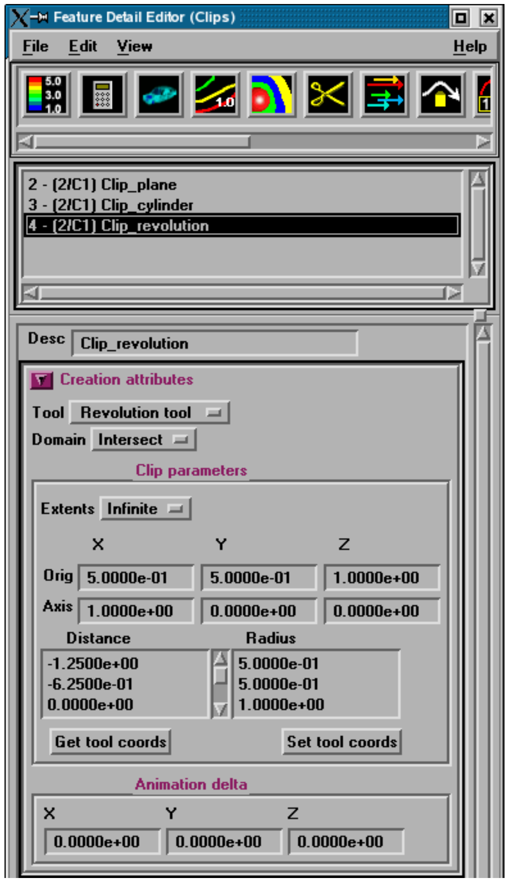


Figure 7-68
Feature Detail Editor (Clips) - Revolution Tool Creation Attributes

Revolution Tool Clip Parameters

Orig — Specify the XYZ coordinates of the origin (center point) of the Revolution Clip.

Axis — These fields specify the XYZ coordinates of the axis direction of the Revolution Clip.

Distance/Radius — These lists specify the distance (from the origin) and radius for each point that defines the Revolution Clip. The points can Not be edited within this dialog. You must edit the Revolution Tool in the Transformations dialog.

Set Tool Coords — Will reposition the Revolution Tool to the position specified in the Clip Parameter fields.

Get Tool Coords — Will update the Clip Parameter fields to reflect the current position of the Revolution Tool.

Animation Delta — These X,Y,Z fields specify the incremental change in position of the clip for each page of Flipbook or frame of Keyframe animation.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see Section 6.5, Tools Menu Functions and How To Use the Surface of Revolution Tool)

Use Tool

Revolve 1D Part This option will create a clip surface by revolving a line, defined by a Part, about an axis.

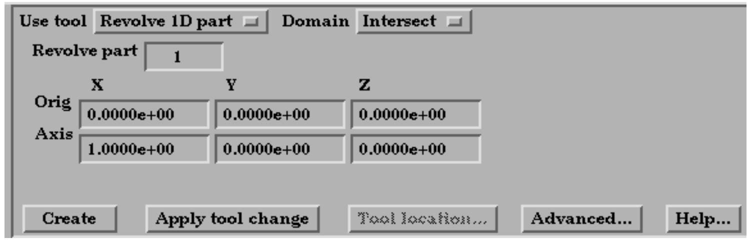


Figure 7-69
Quick Interaction Area - Revolve 1D Part Clip Editor

Domain

Intersect will create the cross section of the selected parts where they intersect the revolved surface.

Crinkly will create a new part consisting of the parent part elements that intersect the revolved surface.

Inside will cut the parent parts and create a new part consisting of the portion on the inside of the revolved surface.

Outside will cut the parent parts and create a new part consisting of the portion on the outside of the revolved surface.

In/Out will cut the parent parts and create two new parts - namely an *Inside* and *Outside* part.

Revolve Part This field specifies the Part number which will be revolved. The 1D Part must contain only bar elements and must have only two free ends (i.e., there must be only one "logical" line contained in the Part).

Orig These fields specify the XYZ coordinates of the axis line origin point.

Axis These fields specify the direction vector of the axis line. The "line" contained in the Part specified by number in Revolve Part will be revolved about this axis to create the clip surface Part.

Apply Tool Change Recreates the Clip Part selected in the Main Parts List at the current position of and of the type specified by Use Tool.

Create Creates the Clip Part in the Graphics Window as specified.

Use Tool

Spline This option will create a clip along an existing spline using evenly spaced nodes along the spline.

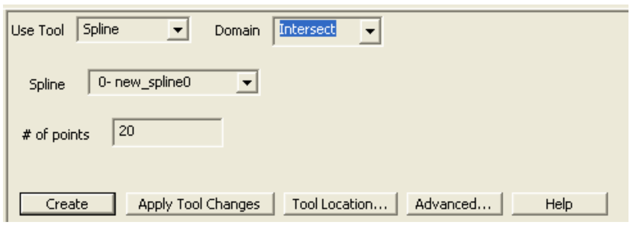


Figure 7-70
Quick Interaction Area - Revolve 1D Part Clip Editor

Domain *Intersect* will create the 1D part composed of bar elements using the selected parts where they intersect the spline at an evenly spaced number of points.

Spline This pulldown allows you to choose which spline to use as the clipping tool.

# of points Enter the number of evenly spaced points to use over the spline for the 1D clip creation.

General Quadric

Feature Detail Editor (Clips)

Double Clicking on the Clip Create/Update Icon brings up the Feature Detail Editor (Clips), the Creation attributes section of which offers access to one type of clip creation which is not available in the Quick Interaction area. It is possible to create a 3D Quadric clip using the General Quadric option by directly specifying the coefficients of a general quadric equation.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not affect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts)

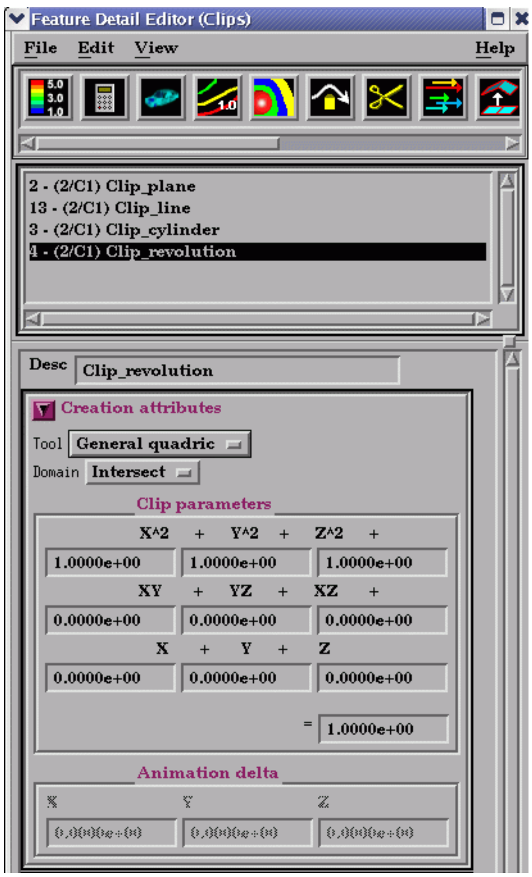


Figure 7-71
Feature Detail Editor (Clips) - Revolve 1D Part Creation Attributes

10 coefficient values

These coefficient values represent the general equation of a Quadric surface. They can be changed by modifying the values. No tool exists corresponding to this equation.

$$AX^2+BY^2+CZ^2+DXY+EYZ+FXZ+GX+HY+IZ=J$$

Animation Delta

Not available for General Quadric Clips.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

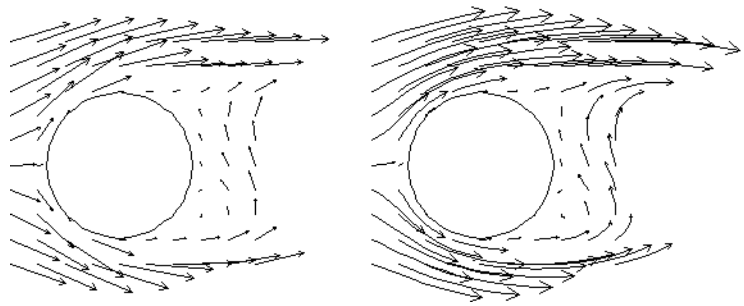
Troubleshooting Clips

Problem	Probable Causes	Solutions
Clip does not move during animation	Animation deltas are not set, or are too small.	Change the animation delta values.
Clip results in an empty Part.	Clip was taken outside of the model.	Change the clip Tool location.

7.10 Vector Arrow Create/Update

Vector Arrows visualize the magnitude and direction of a vector variable at discrete points (at nodes, element vertices, or at the center of elements).

Other features can visualize magnitude, but Vector Arrows also show direction.



Vector arrow Parts are dependent Parts known only to the client. They cannot be used as a parent Part for other Part types and cannot be used in queries. As dependent Parts, they are updated anytime the parent Part and/or the creation vector variable changes (unless the general attribute Active flag is off).

Vector arrows can be filtered according to low and/or high threshold values.

Vector arrows can emanate from the available nodes of the parent Part(s), the available element vertex nodes of the parent Part(s), or the available element centers of the parent Part(s) which pass through the filter successfully. The nodes and elements available in the parent Part are based on the visual Representation of the Part. Thus, for a border Representation of a Part, only the border elements and associated nodes are candidates.

Vector arrows can have straight shafts representing the vector at the originating location, or be the segment of a streamline emanating from the originating location (curved). Straight vector arrows are displayed relatively quickly, while curved vector arrows can be time consuming.

Different tip styles, sizes, and colors can be used to enhance vector arrow display.

To quickly create vector parts, right click in the graphics window on a surface and drag down to vector arrows, and vector arrows will automatically appear (if there's only one vector variable) or you will be prompted for which vector variable to use to create vector variables and they will automatically appear. Left click on the arrows and when a green, cross-shaped handle (click and go handle) appears, drag it left and right to scale the arrows. Left-clicking once on the Vector Arrow Create/Update Icon opens the Vector Arrow Editor section of the Quick interaction Area which is used to both create and update (make changes to) vector arrow Parts.

Figure 7-72
Vector Arrow Create/Update Icon

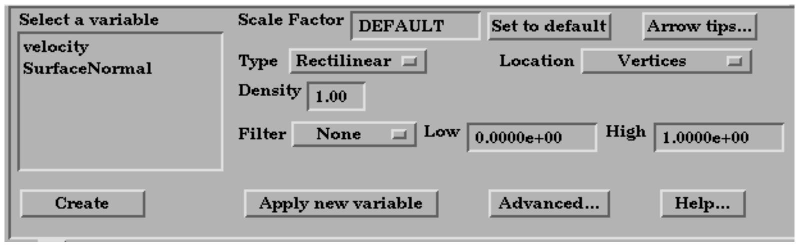


Figure 7-73
Quick Interaction Area - Vector Arrow Editor

Scale Factor / Time When Type is “Rectilinear”, this field specifies a scale factor to apply to the vector values before displaying them. Scaling is usually necessary to control the visual length of the vector arrows since the vector values may not relate well to the geometric dimensions. Can be negative, causing the vector arrows to reverse direction.

When Type is “Rect. Fixed”, this field specifies the length of the arrows in units of the model coordinate system. Can be negative, causing the vector arrows to reverse direction.

When Type is “Curved”, this field specifies the duration time for streamlines forming the shaft of curved vector arrows. Is an indication of the length of the curved vector arrow.

Set to Default Sets Scale Factor value to a computed reasonable value based on the vector variable values and the geometry.

Arrow Tips... Opens the Vector Arrow Tip Settings dialog.

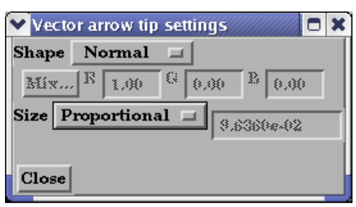


Figure 7-74
Vector Arrow Tip Settings dialog

Shape Opens a pop-up menu to select tip shape.

None option displays arrows as lines without tips.

Cone arrows have a tip composed of a 3D cone. Good for both 2D and 3D fields.

Normal arrows have two short line tips, similar to the way many people draw arrows by hand. The tip will lie in the X–Y, X–Z, or Y–Z plane depending on the relative magnitudes of the X, Y, and Z components of each individual vector. Suggested for 2D problems.

Triangles arrows have a tip composed of two intersecting triangles in the two dominant planes. Good for both 2D and 3D fields.

Tipped arrows display the tip of the arrow in any user specified color. Good for both 2D and 3D fields. The color may be specified in the RGB fields or chosen from the Color Selector dialog which is opened by pressing the Mix... button

Size

Opens a pop-up menu for selecting tip size.

Fixed sized arrows have tips for which the length is specified in the data entry field to the right of the pop-up menu button. Units are in the model coordinate system.

Proportional sized arrow tips change proportionally to the change in the magnitude of the vector arrows.

Type

Opens a pop-up menu for selection of shaft-type of vector arrows. Options are:

Rectilinear arrows have straight shafts. The arrow points in the direction of the vector at the originating location. The length of the arrow shaft is determined by multiplying the vector magnitude by the scale factor.

Rect. Fixed arrows have straight shafts. The arrow points in the direction of the vector at the originating location. The length of the arrow shaft is determined by the scale factor. It is independent of the vector variable.

Curved arrows have curved shafts. The arrow is actually a streamline emanating from the originating location. It represents the path that a massless Particle would follow if the flow field was steady state. For this option, the “Scale Factor” changes to “Time”. Time is the amount of time the streamline is allowed to take and is an indication of how long the arrow will be. Hint: Since curved arrows can take a significant amount of time (depending on the number of originating locations), the setting of a proper “Time” value is critical. The best way to do this is to first do a single Particle trace at a representative location with the estimated “Time” value as the Max Time. A quick iteration or two on the value here could save considerable time for the curved vector arrow computation.

Location

Opens a pop-up dialog for the selection of root-location of arrow shafts. The options are:

Node arrows originate from each node of the parent Part(s). Note: Discrete Particles Parts must use Node option.

Vertices arrows originate only from those nodes at the vertices of each element of the parent Part(s) (i.e., arrows are not displayed at free nodes or mid-side nodes).

Element Center arrows originate from the geometric center of each element of the parent Part(s).

Density

The fraction of the parent’s nodes/elements which will show a vector arrow. A value of 1.0 will result in a vector arrow at each node/element, while a value of 0.0 will result in no arrows. If between these two values, the arrows will be distributed randomly at the specified density. There is no check for duplicates in the random distribution of arrows. It is entirely possible that when you specify a density of 0.25 in a model containing 100 nodes you only get 15 unique locations with 10 duplicates. It will appear that only 15 arrows show up, but there are actually 25 with 10 duplicates.

Filter

Selection of pattern for filtering Vector Arrows according to magnitude. Options are:

None displays all the vector arrows. No filtering done.

Low displays only those arrows with magnitude above that specified in the Low field. Filters low values out.

Band displays only those arrows with magnitude below that specified in the Low field and above that specified in the High field. Filters the band out.

High displays only those arrows with magnitude below that specified in the High field. Filters the high values out.

Low_High displays only those arrows with magnitude between that specified in the Low field and that specified in the High field. Filters out low and high values.

Apply New Variable Changes the vector Variable used to create the Vector Arrows to that currently selected in the Variables List.

Advanced Will open the Feature Detail Editor and select the Vector Arrow icon for more advance control of the Vector Arrow Part.

Feature Detail Editor (Vector Arrows)

Double clicking on the Vector Arrow Create/Update Icon opens the Feature Detail Editor for Vector Arrows, the Creation Attributes Section of which provides access to the functions available in the Quick Interaction Area plus three more:

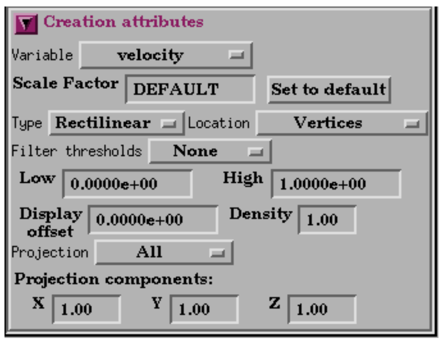


Figure 7-75
Feature Detail Editor (Vector Arrows)

Display offset This field specifies the normal distance away from a surface to display the vector arrows. A positive value moves the vector arrows away from the surface in the direction of the surface normal.

Please note that there is a hardware offset that will apply to contours, vector arrows, separation/attachment lines, and surface restricted particle traces that can be turned on or off in the View portion of Edit->Preferences. This preference ("Use graphics hardware to offset line objects...") is on by default and generally gives good images for everything except move/draw printing. This hardware offset differs from the display offset in that it is in the direction perpendicular to the computer screen monitor (Z-buffer).

Thus, for viewing, you may generally leave the display offset at zero. But for printing, a non-zero value may become necessary so the arrows print cleanly.

Projection Opens a pop-up menu to allow selection of which vector components to include when calculating both the direction and magnitude of the vector arrows. The vector components at the originating point are always first multiplied by the Projection Components (see below). Then one of the following options is applied:

All to display a vector arrow composed of the Projection-Component-modified X, Y, and Z components.

Normal to display a vector which is the projection of the All vector in the direction of the normal at the originating location.

Tangential to display a vector which is the projection of the All vector into the tangential plane at the originating location.

Component to display both the Normal and the Tangential vectors.

The *All, Normal*, and *Tangential* options produce a single vector per location, while the *Component* option produces two vectors per location. If selection is not applicable to a Particular element, that element's vector arrow uses the *All* projection.

Projection Components X Y Z These fields specify a scaling factor for each coordinate component of each vector arrow used in calculating both the magnitude and direction of the vector arrow. Specify 1 to use the full value of a component. Specify 0 to ignore the corresponding vector component (and thus confine all the vector arrows to planes perpendicular to that axis). Values between 0 and 1 diminish the contribution of the corresponding component, while values greater than 1 exaggerate them. Negative values reverse the direction of the component. Always applied before the Projection options above.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How to Create Vector Arrows)

Troubleshooting Vector Arrows

Problem	Probable Causes	Solutions
Vector arrows do not match up with their originating locations on one or more of the parent Parts.	Displacements are on for some of the parent Parts, but not others. Or the parent Parts have been assigned to different coordinate frames	Create separate vector arrow Parts for the parents that will be displaced (or assigned to different frame) and the ones that will not be displaced (or assigned to different frames).
You are displaying several different vector arrow Parts at once and can't tell which is which.	Just too much similar information in the same area.	Use different attributes for the different vector arrow Parts, or better yet, display the conflicting vector arrow Parts on separate Part copies which have been moved apart.
You are trying to display vector arrows on a Discrete Particle Part, but can't get them to show up	Arrow Location set to Vertices (the default).	Set the Arrow Location to Nodes.
	No vector data provided for the Discrete Particle dataset, thus values all set to zero when read into EnSight.	Provide vector data for the particles. Specify in the Measured results file. See Section 3.7.
Vector arrows do not print well	See Display Offset discussion above.	Enter a non-zero Display Offset.

7.11 Particle Trace Create/Update

A streamline or pathline *Particle trace* visualizes a vector field by displaying the path that a particle would follow if placed in that field. At each point on the Particle trace, the direction of the trace is parallel to the vector field at that point and time.

A *streamline* is a Particle trace in a steady-state vector field, while a *pathline* is a Particle trace in a time-varying vector field. Particle traces can be lines or "ribbons" (that additionally visualize the rotation of the vector field around the path of the trace).

EnSight is capable of computing a pathline through a model with changing coordinates and/or changing connectivity. The variable values are assumed to behave linearly between the known timesteps.

Node Tracks

Another form of trace that is available is entitled node tracking. This trace is constructed by connecting the locations of nodes through time. It is useful for changing geometry or transient displacement models (including measured particles) which have node ids.

Min/Max Variable Tracks

A further type of trace that is available is a min or max variable track. This trace is constructed by connecting the min or max of a chosen variable (for the selected parts) though time. Thus, on transient models one can follow where the min or max variable location occurs.

Particle Trace Parts have their own attributes, so you can, for example, trace a flow field using the velocity variable, and then color the resulting trace using the temperature variable.

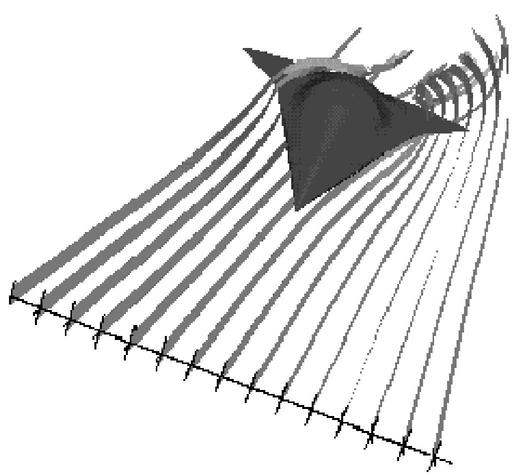


Figure 7-76
Particle Trace Illustration

Emitters

A streamline or pathline Particle Trace Part consists of one or more Particle traces originating from points on one or more *emitters*. Each emitter is capable of emitting a Particle starting at a specified time and continuing to emit Particles at given intervals. When pathlines are generated with emitters emitting at multiple time intervals and these traces are then animated, *streaklines* are displayed.

Emitters consist of single points, points along a line, points forming a grid in a plane, or points corresponding to the nodes of a Part. You can define emitters using the Cursor tool, the Line tool, the Plane tool, or a Part. In fact, if you have a cursor, line or plane tool visible in a flow field, you can right click on the tool and choose ‘Particle Trace’ to immediately create a particle trace in the flow field using all the default settings.

Emitters can be created using the cursor, line, and plane tools, using existing Part nodes, or can be created in a surface restricted mode where the mouse can be used to project points, rakes or nets directly onto the displayed surfaces of the model.

Pathlines, of course, must be drawn forward in time, but streamlines can be drawn forward in time, backward in time, or both. Each Particle trace terminates when either (1) the Particle trace moves outside the space in which the vector field is defined, (2) a user-specified time limit is reached, (3) the massless Particle becomes stationary in a place where the vector field is zero, or (4) the last transient-data time step is reached. (4 applies only to pathlines)

A Particle trace can pass through any point inside an element of the parent Part(s). The vector field at any point is calculated from the shape function of the containing element. Emitter points located outside the elements are ignored when creating Particle traces.

Surface-Restricted Traces

A surface-restricted Particle trace is constrained to the surface of the selected Part(s) by using only the tangential component of the velocity. The velocity values for this type of trace can be the velocity at the surface (if nonzero) or at some user specified offset into the velocity field.

Interactive Traces

A streamline Particle trace can be updated interactively by entering interactive mode and moving the tool used to create the emitter. When a trace is selected and interactive emitter is turned on, the tool will appear at the location of the emitter. The user then manipulates the tool interactively in the Graphics Window or using the transformations dialog. (This option is not available for surface-restricted Particle traces, traces emitted from a Part nor in Server of Server mode).

Integration Method

EnSight creates streamline and pathline Particle traces by integrating the vector field variable over time using a Fourth Order Runge-Kutta method and utilizing a time varying integration step. The integration step is lengthened or shortened depending on the flow field, but you can control the minimum number of integration steps performed in any element as well as other time step controls.

Normally, EnSight will perform the integration using all of the components of the vector. However, it is possible to restrict the integration to a plane by specifying which components of the vector to use. Typical uses of this feature would be to restrict the Particle traces to a clip plane. Surface-restricted Particle traces provide even greater flexibility in restricting a trace to planes or other surfaces.

Max # of Segments

A trace will continue until it attempts to leave the flow field. However sometimes a trace will continue indefinitely, for example when the trace is caught in a vortex or recirculation area. For this situation EnSight has established a maximum number of segments in each trace. The maximum number of segments in each massless trace is by default 5000 for surface-restricted traces, and 6000 for all other massless trace types. This default can be changed by entering a command line entry (File>Command) into the command dialog.

Surface restrcted: `test: max_skin_segments <value>`

All other trace types: `test: max_ptrace_segments <value>`

A value of 0 disables the maximum segment check, and for massed particles the value entered is automatically doubled.

Lines or Ribbons

Line-type Particle traces consist of bar elements. Ribbons consist of 4-noded quad elements and originate with their end-edge parallel to the Z-axis of the global frame. Then, at each integration step, the leading edge is rotated around the current direction of the path by the same amount the vector field has rotated around the path since the previous time step. Ribbons are not available for surface-restricted Particle traces.

Particle Trace Parts are created on the server, so the Representation-type of the parent Parts has no effect. The algorithm that creates Particle traces initially sets up a cross-referencing map of adjoining elements. Hence, the first Particle trace takes longer to generate than subsequent traces.

If you calculate pathlines, consider calculating as many as possible at a time, since the process can be very time consuming (most of the time is taken in reading time step information). However, the data for the Trace Part is sent to and stored on the Client. Thus, you cannot label nodes or elements, and some query options may be limited.. In such cases, it may be helpful to perform the labeling or query option on the Particle Trace's parent Part(s) instead. Line-type Particle Traces can be parent Parts for Profiles. You can animate the motion of the massless Particles along their Particle traces.

Transient Data

By default the emission point is always set to emit the Particles at the current time step. This can be a problem if you have a transient dataset with the current time set at the last time step available. If you compute pathlines from this location, the default emission time will be at the current time (last time step), thus no pathlines will be generated. To solve this problem you will need to either change the current time, or change the Start Time of the emitter.

The process of creating a streamline or pathline Particle trace is always to specify an emission point (location and time), specify the Part(s) to trace the Particle through and specify which vector variable to integrate. There are quick ways of doing this process which assume that the correct defaults are set, or there are more deliberate ways which give you more control. Particle trace Parts carry only one set of attributes for all of the traces in the Part, thus it is not possible, for example, to trace some of the emission points forward in time and others backward in time.

Particle trace Parts are different from all other created Parts in that when the parent Parts change (such as at a time step change), the Particle trace Part does not change. This is due to the fact that the Particle trace has been created at a specified time (the emission time), making the Part independent of time (after the trace has been created).

Regular streamline or pathline Particle traces can only be computed through a set of parent Parts consisting of model Parts. Surface-restricted Particle traces can be created on model Parts, clip Parts, elevated surface Parts, and developed surface Parts.

If your dataset contains 3D elements, the Particles for regular traces will be traced through 3D element fields only. Surface-restricted traces would have to be used to trace along 2D elements of such a data set.

Massed-Particle Traces

A particle trace can be created or updated from a massless-particle trace to a massed-particle trace, or visa-versa. Massed-particle traces are specified via their appended section in the Feature Detailed Editor (Traces) dialog. Massed particle traces use an RK4(5) (Fehlberg) integration algorithm.

Definitions

Motion of a particle as a function of its velocity is defined as

$$d/dt(X_p) = V_p$$

with initial conditions $V_p(t_0)=V_{p,0}$ and initial particle position $X_p(t_0)=X_{p,0}$ (capital letters denote vectors unless otherwise indicated).

For massless particles, the particle velocity is always identical to the local fluid velocity, $V_p=V_f$. For massed particles, additional forces acting on them result in a different velocity for the V_p than for the fluid, V_p not equal to V_f. This particle velocity is determined from a momentum balance for the particle by

$$m_p A_p = F_p,$$

or

$$m_p \; d/dt(V_p) = F_g + F_b + F_{\cancel{p}} + F_d + F_e,$$

where

A_p = particle acceleration vector,

F_p = total (particle) force vector,

F_g = gravitational (body) force vector = $m_p G$,

F_b = buoyancy (body) force vector = $- m_p G(\rho_f/\rho_p)$,

$F_{\cancel{p}}$ = pressure (surface) force vector = $- v_p \nabla \cancel{p}_f$,

F_d = drag (surface) force(s) vector = $- \frac{1}{2} \rho_f \, a_p \, c_d \, |V_r| \, V_r$,

F_e = additional forces vector, here = 0,

given the following definitions (Note: the underlined definitions are user specified)

m_p = spherical particle mass = $\rho_p \, d_p^3 \, \pi/6$,

ρ_p = particle density,

d_p = particle diameter,

ρ_f = fluid density (scalar or constant),

G = gravitational acceleration vector,

$\nabla \cancel{p}_f$ = fluid pressure gradient vector, (computed from $\cancel{p}_f$ = fluid pressure scalar variable)

a_p = particle reference area = $d_p^2 \, \pi/4$,

V_p = particle velocity vector,

V_f = fluid velocity vector,

V_r = reference velocity vector = $V_p - V_f$,

c_d = drag coefficient, typically given as a function of the local relative Re, i.e. $c_d = c_d(Re)$,

Re = Reynolds number = $\rho_f \, |V_r| \, d_p \, / \, \mu_f$,

μ_f = fluid dynamic viscosity (scalar or constant).

Thus, the total mass balance equation for massed particles may be defined by:

$$m_p \; d/dt(V_p) = m_p G - m_p G(\rho_f/\rho_p) - (v_p \nabla p_f) - (\tfrac{1}{2}\rho_f a_p c_d |V_r| V_r)$$

Dividing through by the particle mass m_p yields the following acceleration terms:

$$d/dt(V_p) = G - G(\rho_f/\rho_p) - \nabla p_f/\rho_p - \tfrac{1}{2}\rho_f |V_r| V_r (a_p c_d/m_p)$$

Note the following relation in the drag acceleration term:

$$(a_p c_d/m_p) = 1/c_b$$

where: c_b = <u>ballistic parameter</u> or <u>coefficient</u> = $m_p/(a_p c_d) = 2\rho_p d_p/(3c_d)$

Drag Coefficient

Currently, the following Drag Coefficient (C_d) table is provided as the default.

$Re \ll 1$	$C_d = 24/Re$
$1 < Re \ll 500$	$C_d = 24/Re^{0.646}$
$500 < Re \ll 3e5$	$C_d = 0.43$
$3e5 < Re \ll 2e6$	$C_d = 3.66E\text{-}4\ Re^{.4275}$
$2e6 < Re$	$C_d = 0.18$

This table is also coded as an example for your reference and access via the User-Defined Math Function `DragCoefTable1(Re)` which is found in

$CEI_HOME/ensight92/src/math_functions/drag_coef_table1/libudmf-drag_coef_table1.c

In addition, two other drag coefficient functions are provided for your selection via the User-Defined Math Function facility.

$$\text{DragCoefPoly(Re)} = (a + b\,Re + c\,Re^2 + d/Re)$$

Where: {a,b,c,d} are polynomial coefficients with default values of {1.,0.,0.,0}, respectively.

$$\text{DragCoefPower(Re)} = (1 + .15\,Re^{0.687})\ 24\ /\ Re$$

Both of these functions are located respectively in

$CEI_HOME/ensight92/src/math_functions/drag_coef_poly/libudmf-drag_coef_poly.cf

$CEI_HOME/ensight92/src/math_functions/drag_coef_power/libudmf-drag_coef_power.c

You may also code your own. (See UDMF in EnSight User Manual.)

Rebound Massed Traces (Off a Boundary Wall)

Massed-particle traces can be toggled to rebound off boundary walls. The rebounding massed traces are based on the following derivation. The derivation assumes both the massed particle and boundary wall (or boundary) are both rigid so that there is no deformation of the massed particle or boundary. Also rotational considerations are ignored.

Starting with the initial impact particle velocity

$$V_p = V_p(V_{Ni}, V_{Ti}), \qquad (R0)$$

the tangential friction force opposing the massed particle is given by

$$F_T = -m_p dV_T/dt = -m_p(V_{Tr}-V_{Ti})/dt, \qquad (R1)$$

and the normal reaction force is given by

$$F_N = m_p dV_N/dt = m_p(V_{Nr}-V_{Ni})/dt, \qquad (R2)$$

The tangential friction force is proportional to the normal reaction force by the *coefficient of friction* μ given by

$$F_T = \mu F_N. \qquad (R3)$$

Equating R1 to R2, canceling out m_p/dt, and taking into account R3 we have

$$-(V_{Tr}-V_{Ti}) = \mu(V_{Nr}-V_{Ni}) \qquad (R4)$$

Solving for V_{Tr} we have

$$V_{Tr} = V_{Ti}-\mu(V_{Nr}-V_{Ni}). \qquad (R5)$$

Now given that in the normal direction the final (reflected) velocity of the massed particle is proportional to the initial (incident) velocity of the massed particle by the *coefficient of restitution* ε, we have

$$V_{Nr} = \varepsilon V_{Ni} \qquad (R6)$$

which is the final normal component of the rebounding massed-particle velocity. Subbing R6 into R5 and simplifying we have

$$V_{Tr} = V_{Ti}+\mu(1+\varepsilon)V_{Ni} \qquad (R7)$$

which yields the final tangential component of the rebounding massed-particle velocity. Combining these two components (R6 and R7) yield the final rebounding particle velocity

$$V_p = V_p(V_{Nr}, V_{Tr}). \qquad (R8)$$

Where:

m_p	=	the mass of the particle as defined above.
F_T	=	the friction force tangent to the boundary "opposing" the particle (thus the "-" sign on the right hand side of the equation assuming the particle is traveling in the positive direction).
F_N	=	the normal force on the boundary "opposing" the particle (assuming the normal direction back into the field as positive).
V_T	=	the tangential component of the particle velocity V_p.
V_{Ti}	=	The tangential component of the incident V_p impacting onto the boundary.
V_{Tr}	=	The tangential component of the reflected V_p bouncing off the boundary.
V_N	=	The normal component of the particle velocity V_p.

V_{Ni} = The normal component of the incident V_p impacting onto the boundary.

V_{Nr} = The normal component of the reflected V_p bouncing off the boundary.

ε = The coefficient of restitution.

μ = The coefficient of friction.

Particle-Mass Scalar on Boundaries

Information to compute a particle-mass scalar on boundaries ($m_P = \sum m_{Pi}$) is provided each time massed-particle traces are created. This scalar is found and computed via the New Computed Variables (NCV) functionality.

Massed Particle Scalar(massed-particle traced part(s))

This scalar creates a massed-particle per element scalar variable for each of the parent parts of the massed-particle traces. This per element variable is the mass of the particle times the sum of the number of times each element is exited by a mass-particle trace.

References

The following references have contributed in part toward the development of the massed-particle algorithm.

Donley, H. Edward
“The Drag Force on a Sphere”,
http:\\www.ma.iup.edu/projects/CalcDEMma/drag/drag.html

Lund, Christoph
“Vorgaben für die Berechnung und Visualisierung der Bahnlinien massebehafteter Partikel im Postprozessor EnSight”, Volkswagen AG, 27.07.2001. English translation by Kent Misegades.

Fluid Dynamics International, Inc.
FIDAP 7.0 Theory Manual”, April 1993, pp12-3+

Danby, J.M.A.
“Conputing Applications to Differential Equations”,
Restin Pub. Co., Inc. Restin, VA; 1985

Howard Brady, Rod Cross, Crawford Lindsey
“The Physics and Technology of Tennis”,
Raquet Tech Pub., Solana Beach, CA, 2002

Richard Burden, J. Douglas Faires, Albert C. Reynolds
“Numerical Analysis, 2nd Ed.”,
Prindle, Weber, & Schmidt, Boston, 1978

Paul Tipler
“Physics”,
Worth Pub. Inc.; NY, 1976

Clicking once on the Particle Trace Create/Update Icon opens the Particle Trace Editor in the Quick Interaction Area which is used to both create and update (make changes to) Particle trace Parts.

Figure 7-77
Particle Trace Create/Update Icon

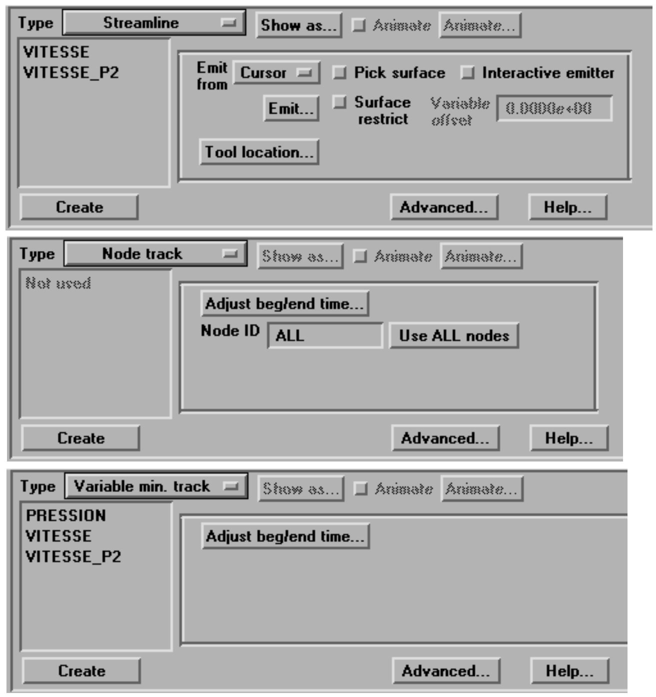


Figure 7-78
Quick Interaction Area - Particle Trace Editor

Record Button
(Enabled with active trace animation)

Type — Opens a pull-down menu for specification of whether Particle trace calculation uses steady-state data (streamlines), transient data (pathlines), or one of the tracking options.

Node Track — Tracks one (or all) nodes of the selected part(s) through time. Requires node ids and transient geometry or transient displacements.

Pathline — Traces a massless Particle through a time-varying vector field *and so is only available with transient results data*. On certain systems, this selection can consume significant quantities of CPU time to calculate the resulting Particle trace.

Streamline — Traces a massless Particle in a steady-sate vector field (for steady-state data or the current time-step of transient data).

Variable min track — Tracks the location of the minimum value of the chosen variable through time.

Variable max track — Tracks the location of the maximum value of the chosen variable through time.

Show As — Opens a dialog for specification of trace representation for streamline and pathline traces.

Line — Depicts the trace as a line.

Ribbon — Depicts trace as if it were a ribbon. The ribbon width is a specified fixed value, while the twisting is determined by the rotation of the flow about the path of the trace at any particular point on the trace.

Square Tubes — Depicts trace as if it were a square tube. The tube width is a specified fixed value, while the twisting is determined by the rotation of the flow about the path of the trace at any particular point on the trace.

Animate Toggle

Toggles on/off the animation of the motion of the Particles along the traces. In addition to creating Particle traces based on vector variables, EnSight can also animate the motion of the Particles along the Particle traces. To distinguish them from discrete Particles, we call Particles moving along Particle traces “tracers.”

At any instant, each tracer consists of a portion of a Particle trace displayed with attributes you specify separately from the attributes of the Particle trace. EnSight animates each tracer by updating which portion of the Particle trace is currently displayed. You specify the length of each tracer as a time value, so the tracer’s length varies dynamically as it moves down the Particle trace (faster moving tracers are longer). This option can add tremendously to the understanding of the flow field since relative speed can be determined.

EnSight provides control over how the tracer looks and acts. You can animate one, some, or all of the Particle traces you have created, but they are all animated in the one way you specify. To help you get started, at the click of a button EnSight will suggest time-specification values based on the Particle traces you have selected to animate. You can specify the line width of the tracer, and choose to color it with a constant color or the same calculated color used to color the Particle trace. If you wish to change the opacity of the trace in the Color, Lighting and Transparency dialog, you cannot do it with the opacity slider, but must instead use the “screen door” transparency in the Fill Pattern pulldown. You can also display a spherical “head” on the leading-end of the tracer, and dynamically size the head according to any active variable.

You control the speed of the motion and have the option to display multiple tracers on the same Particle trace separated by a time interval. Hence, you can choose to view rapid-fire pulses, slow moving “noodles,” or something in between. For steady-state Particle traces (streamlines), “time” is the integration time with the emitters located at time zero. For transient Particle traces (pathlines), you have the option to synchronize the animation time to the solution time. The choice of whether a Particle trace is a streamline or a pathline is made when you create the Particle trace.

You do not have to animate the entire Particle trace. You can specify where you want the animation to start with a time value corresponding to a distance down the Particle trace from the emitter, and where you want the animation to stop with a time value corresponding to a distance farther down the Particle trace.

Tracers on all animated Particle traces are synchronized. If you combine Particle trace animation with flipbook animation or keyframe animation, the animation time values are automatically synchronized if you toggle-on Sync To Transient in the Trace Animation Settings dialog.

Animate... Opens the Trace Animation Settings dialog

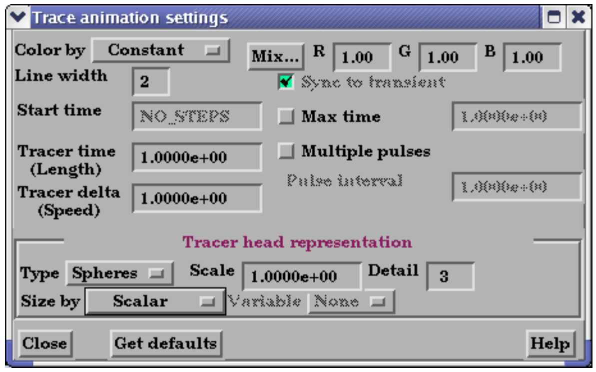


Figure 7-79
Trace Animation Settings dialog

Color By Opens a pull-down menu for selection of method by which to color the tracers.

Constant — Displays tracers in the constant color specified in this dialog.
Mix… — Opens the Color Selector dialog (see Section 8.1, Color Selector).
R,G,B — Fields allow specification of constant color.

Trace Color — Displays tracers in the same color as the Particle Trace Part from which they originate.

Line Width Specification of displayed width (in pixels) of tracers. Note: Line Width specification may not be available on some workstation platforms.

Start Time Specification of how far down each Particle trace to begin displaying tracers. A Particle trace is made up of line segments. Each segment that makes up a Particle trace has an associated time value. The start time indicates where on the Particle trace the tracers will begin animation.

Tracer Time (Length) Specification of length of tracers which varies as the tracer speed varies along the Particle trace. The Particle Time Length parameter scales the length of all tracers at all times.

Tracer Delta (Speed) Specification of how fast tracers move. Longer times result in faster moving tracers. This parameter is not applicable when using Sync To Transient and displaying transient data through flipbook or keyframe animation.

Sync to Transient Toggle Toggles on/off synchronization of tracer position to solution time of transient data. When toggled-on, and transient data is in use (i.e. solution time, flipbook or keyframe animations), each tracer is displayed with its leading-end at the correct location along the Particle trace for the current solution time. Traces only move forward in time so cycling through transient data is not applicable here.

Max Time Toggle Toggles on/off maximum lifetime for all tracers. If toggled-off, tracers continue to end of Particle trace. If toggled-on, each tracer stops after moving down the Particle trace for a distance corresponding to the specified Max Time (or until one of the other conditions that stop a tracer occurs).

Max Time Field specifies lifetime of all tracers when Set Max Time is toggled-on.

Multiple Pulses Toggle Toggles on/off multiple emission of tracers. When toggled-off, a single tracer for each Particle trace appears at the specified Start Time. When toggled-on, additional tracers appear after each specified Pulse Interval. Not applicable to pathlines.

Pulse Interval Field specifies time delay between tracers. Not applicable when Multiple Pulses is toggled-off.

Tracer Head Representation

Type — Opens a pull-down menu for selection of type of head for each tracer.

None	Specifies that no head will appear.
Spheres	Specifies that a sphere will appear on the leading end of the tracer.
Scale	Specification of scaling factor for head size. Values between 0 and 1 reduce the size, factors greater than one enlarge the size. Not applicable when Head Type is None.
Detail	Specification of detail-level of head in range from 2 to 10, with 10 being the most detailed (e.g., rounder spheres because more polygons are used to create spheres). Higher values take longer to draw, slowing performance. Not applicable when Head Type is None.
Size By	Opens a pull-down menu for the selection of variable-type to use to size each tracer's head. If you select a variable, the head size is determined by multiplying the Scale factor times the variable value, which will vary depending on the location of the tracer. Not applicable when Head Type is None.
Constant	Sizes head using just the Scale factor value.
Scalar	Sizes head using a scalar variable.
Vector Mag	Sizes head using magnitude a vector variable.
Vector X	Sizes head using X-component of a vector variable.
Vector Y	Sizes head using Y-component of a vector variable.
Vector Z	Sizes head using Z-component of a vector variable.
Variable	Selection of variable to use to size the tracer heads. Not applicable when Type is None or Size By is Constant.
Get Defaults	Click to set time-value specifications in this dialog to values suggested by EnSight based upon the characteristics of the selected Particle traces.

See Also: How To Animate Particle Traces

Troubleshooting Animated Particle Traces

Problem	Probable Causes	Solutions
No motion. Can't see any tracers.	No Particle traces selected to animate.	Select the traces you wish to animate in the list at the top of the Animated Trace Setup dialog.
	Tracers colored same as Particle traces and have same line width.	Change Color By or Line Width.
	Animate Traces not toggled-on.	Toggle Animate on in the Quick Interaction Area.
	Start Time > maximum Particle trace time for all traces selected.	Change settings in the Trace Animation Settings dialog.
	Delta Time (Speed) set too high.	Change settings in the Trace Animation Settings dialog.
	Particle Time (Length) set too small.	Change settings in the Trace Animation Settings dialog.
Motion too fast.	Delta Time (Speed) set too high.	Change settings in the Trace Animation Settings dialog.
Can't get multiple pulses at same time.	Pulse interval too high.	Decrease to have pulses start closer together.

Problem	Probable Causes	Solutions
Have one big tracer, no pulses.	Pulse interval too small, pulses start right after each other with no separation.	Increase the interval.

Quick Interaction Area Particle Trace Editor, continued,

Emit From — Opens a pull-down menu for the specification of the emitter type.

Cursor — Creates Particle trace beginning from the position of the Cursor tool.

Line — Creates Particle traces beginning from the position of the Line tool.

Points — This field specifies the number of evenly spaced traces you want to emit from the Line tool.

Plane — Create Particle traces beginning from the position of the Plane tool.

Points — These fields specify the number of traces you want to emit from the Plane tool in the X and Y axes of the tool.

Part — Creates particle traces beginning from nodes of the Part specified by the Part ID Number field.

Part ID — This field specifies the Part you wish to use as an emitter for the creation of a particle trace. The Part ID number for a Part is found in the Main Parts List.

Number of Emitters — This field specifies the number of emitters desired. They will be randomly selected from the nodes of the part. (see Section 3.1, Part Overview)

File — Creates particle traces from the locations specified in an external file.
(see Section 11.12, EnSight Particle Emitter File Format)

Emit... — Opens the Emission Detail Attributes dialog.

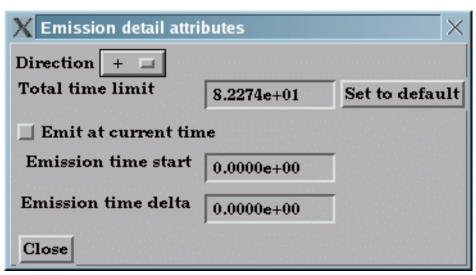


Figure 7-80
Emission Detail Attributes dialog

Direction — Trace the Particle in positive time, meaning to trace with the vector field, or trace the Particle in negative time, meaning to trace the Particle upstream. Option only applies to streamlines. Pathlines must be traced in + time.

(+) — Positive time option traces Particle(s) forward in time. (This is the only option for pathlines.)

(–) — Negative time option traces Particle(s) backward in time.

(+/–) — Positive/Negative time option traces Particle(s) both forward and backward in time.

Total Time Limit — This field specifies the maximum length of time the Particle trace may continue (it may terminate sooner for other reasons). For vector fields with recirculation zones, this can be important to keep from integrating a trace indefinitely.

Set to default — This button sets the Total Time Limit field to a reasonable default value using the vector value and the geometry size.

Emission Time Start — This field specifies the simulation time at which to begin Particle emission. Enter value

	between beginning and ending time available.
Time Delta	This field specifies the time interval between emissions of Particles from the emitters. If “0”, only one set of emissions will occur at start time
Pick Surface Toggle	Toggles on/off the feature which allows you to place the trace emitter at a point on a surface directly below the mouse pointer by clicking the left mouse button.
Surface-Restrict Toggle	Toggles on/off surface restricted feature for streamlines. The streamline will be constrained to stay on the surface of the selected Part(s) by using only the tangential component of velocity. Be sure to use the Pick Surface feature in locating the emitter for a surface restricted particle trace to ensure that the emitter is located on the surface of a Part.
Variable Offset	If Surface-Restrict toggled on, this field specifies the distance into the flow field at which velocity (and other variables) are to be sampled for the surface restricted trace(s). If velocity values are present at the surface, this offset can be set to zero.
Interactive Emitter	Toggles on/off interactive Particle tracing. Manipulation of the Cursor, Line or Plane tool will cause the Particle trace to be recreated at the new location and updated in the Graphics Window. When manipulation of the tool stops, the Particle trace and any Parts that are dependent on it will be updated. (Only available for non-surface-restricted streamlines).
Tool Location...	Opens Transformations Editor dialog which allows you to precisely position the Cursor, Line or Plane tool.
Adjust beg/end time...	Opens the Solution Time area so the beginning and ending times to be used for Node Tracks or Min/Max Variable tracks can be modified.
Node ID	Field for specifying the desired node id of a Node Track. Note that a single node id can be specified to track a single node, or ALL can be specified to track all nodes of the selected part(s).
Use ALL nodes	Clicking this button sets the value in the Node ID field to ALL.
Create	Creates a Particle trace Part using the selected Part(s) in the Main Parts List and the vector Variable selected in the Main Variables List.
Advanced	Will open the Feature Detail Editor and select the Particle Trace icon for more advance control of the Particle Trace Part.
Record Button	Opens a dialog for recording trace animation to a file. This button is only enabled when an active trace animation is playing.

Feature Detail Editor (Traces)

Double clicking on the Particle Trace Create/Update Icon opens the Feature Detail Editor for Particle Traces, the Creation Attributes Section of which provides access to additional functions for trace creation and modification:

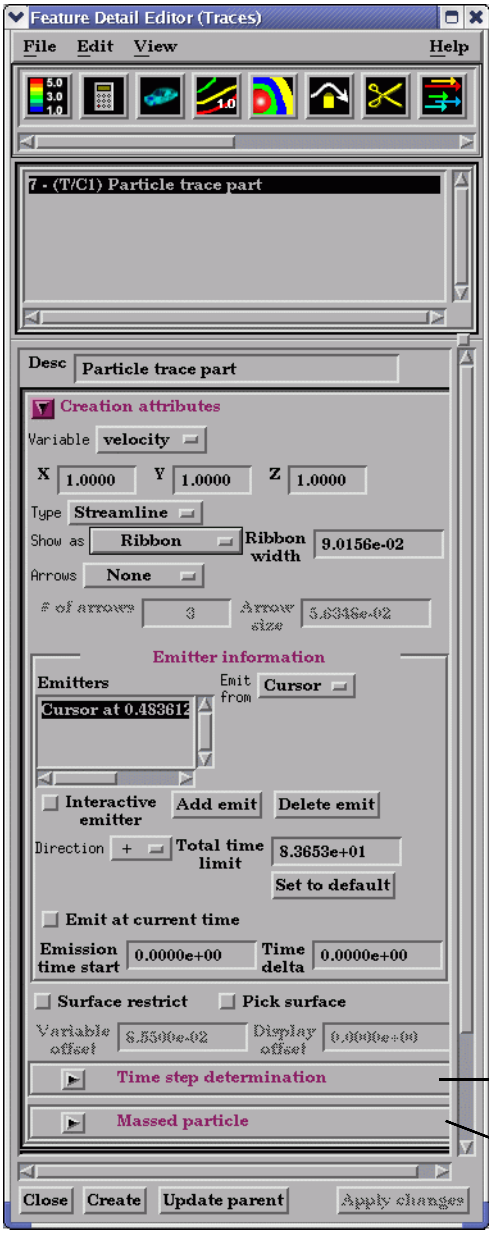
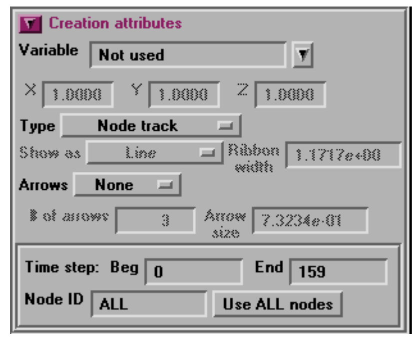
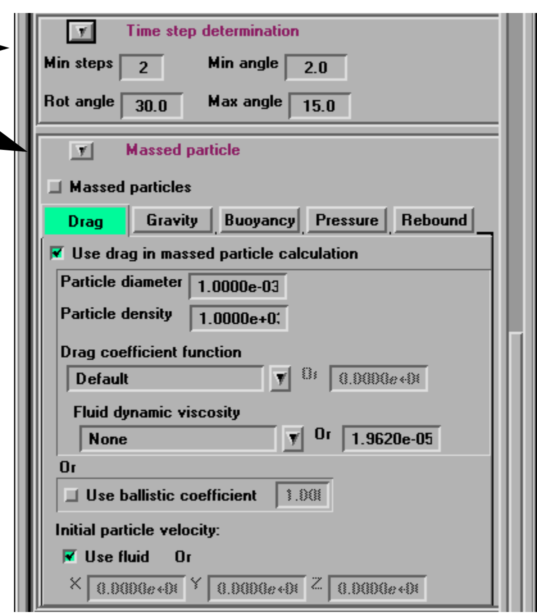


Figure 7-81
Feature Detail Editor (Traces)



Creation Attributes for Node Tracks. For Min/Max Variable tracks, it is similar but without the Node ID field.



Contents of Time Step Determination and Massed Particle turn-downs. (Drag tab)

Variable	Opens a pop-up menu for the selection of an active variable to use to calculate the trace. This is not used for Node Tracks.
X Y Z	These fields specify the fraction of each vector component to be used in the calculation. Specify 1 to use the full value of the vector component. Specify 0 to ignore the corresponding vector component (and thus confine the motion of the Particle to a plane perpendicular to that axis). Values between 0 and 1 diminish the contribution of the corresponding component, while values greater than 1 exaggerate them.
Type	Opens a pull-down menu for specification of whether Particle trace calculation uses steady-state data (streamlines), transient data (pathlines), or one of the tracking options.
Node Track	Tracks one (or all) nodes of the selected part(s) through time. Requires node ids and transient geometry or transient displacements.
Pathline	Traces a massless Particle through a time-varying vector field *and so is only available with transient results data*. On certain systems, this selection can consume significant quantities of CPU time to calculate the resulting Particle trace.
Streamline	Traces a massless Particle in a steady-sate vector field (for steady-state data or the current time-step of transient data).
Variable min track	Tracks the location of the minimum value of the chosen variable through time.
Variable max track	Tracks the location of the maximum value of the chosen variable through time.
Show As	Opens a dialog for specification of trace representation. Only available for streamlines and pathlines
Line	Depicts the trace as a line.
Ribbon	Depicts trace as if it were a ribbon. The ribbon width is a specified fixed value, while the twisting is determined by the rotation of the flow about the path of the trace at any particular point on the trace.
Ribbon Width	This field only applies when Ribbon representation is chosen. Larger values in this field produce wider ribbons. Only available for streamlines and pathlines.
Arrows	Controls whether the flow direction is indicated with arrows. *None* option displays arrows as lines without tips. *Cone* arrows have a tip composed of a 3D cone. Good for both 2D and 3D fields. *Normal* arrows have two short line tips, similar to the way many people draw arrows by hand. The tip will lie in the X–Y, X–Z, or Y–Z plane depending on the relative magnitudes of the X, Y, and Z components of each individual vector. Suggested for 2D problems. *Triangles* arrows have a tip composed of two intersecting triangles in the two dominant planes. Good for both 2D and 3D fields.
# of Arrows	Controls density of arrows. The trace with the longest dwell time will have this number of arrowheads, and the other traces will get a number that corresponds to their dwell time.
Size	Scaling size of Arrowheads.
Time step Beg/End	Allows for specification of beginning and ending time steps for Node Tracks and Min/Max Variable Tracks. It is not available for streamline and pathline traces, which are controlled by emitter information.
Node ID	Field for specifying the desired node id of a Node Track. Note that a single node id can be specified to track a single node, or ALL can be specified to track all nodes of the selected part(s).
Use ALL nodes	Clicking this button sets the value in the Node ID field to ALL.

Emitter Information

Emitters List	This section shows a list of all emitters created for the currently selected Particle Trace Part.
Emit From	Opens a pull-down menu for the specification of the emitter type.
Cursor	Creates Particle trace beginning from the position of the Cursor tool.
Line	Creates Particle traces beginning from the position of the Line tool.
# Points	This field specifies the number of traces you want to emit from the Line tool.
Plane	Create Particle traces beginning from the position of the Plane tool.
# Points	These fields specify the number of traces you want to emit from the Plane tool in the X and Y axes of the tool.
Part	Creates particle traces beginning from each node of the Part specified by the Part ID Number field.
Part ID Number	This field specifies the Part you wish to use as an emitter for the creation of a particle trace. The Part ID Number for a Part is found in the Main Parts List. (see Section 3.1, Part Overview)
Density	If 1.0, will emit from each node of the part. Lass than 1.0 values indicate a subset of nodes to be used, randomly placed, as emitters.
Interactive Emitter	Toggles on/off interactive Particle tracing. Manipulation of the emitter currently selected in the Emitters List will cause the Particle trace to be recreated at the new location and updated in the Graphics Window. When manipulation of the tool stops, the Particle trace and any Parts that are dependent on it will be updated. (Only available for non-surface-restricted streamlines) (Emitters created by picking a surface or from a Part can not be made interactive).
Add Emit	Adds an emitter of the type specified by Emit From to the currently selected Particle Trace Part.
Delete Emit	Deletes the emitter selected in the Emitters List from the selected Particle Trace Part.
Direction	Trace the Particle in positive time, meaning to trace with the vector field, or trace the Particle in negative time, meaning to trace the Particle upstream. Option only applies to streamlines. Pathlines must be traced in + time.
(+)	Positive time option traces Particle(s) forward in time. (This is the only option for time-dependent datasets.)
(–)	Negative time option traces Particle(s) backward in time.
(+/–)	Positive/Negative time option traces Particle(s) both forward and backward in time.
Total Time Limit	This field specifies the maximum length of time the Particle trace may continue (it may terminate sooner for other reasons).
Set to default	This button sets the Total Time Limit field to a reasonable default value using the vector value and the geometry size.
Emission Time Start	This field specifies the solution time at which to begin Particle emission. Enter value between beginning and ending time available.
Time Delta	This field specifies the time interval between emissions of Particles from the emitters. If “0”, only one set of emissions will occur at start time
Surface-Restrict Toggle	Toggles on/off surface restricted feature for streamlines. The streamline will be constrained to stay on the surface of the selected Part(s) by using only the tangential component of velocity. Be sure to use the Pick Surface feature in locating the emitter for a surface restricted particle trace to ensure that the emitter is located on the surface of a Part.
Pick Surface Toggle	Toggles on/off the feature which allows you to place the trace emitter at a point on a surface directly below the mouse pointer by clicking the left mouse button. This option is forced on if the Surface Restricted Toggle is on.

Variable Offset — This field specifies the distance into the flow field at which velocity (and other variables) are to be sampled for the surface restricted trace(s). If velocity values are present at the surface, this offset can be set to zero.

Display offset — This field specifies the normal distance away from a surface to display the surface restricted traces. A positive value moves the traces away from the surface in the direction of the surface normal.

Please note that there is a hardware offset that will apply to contours, vector arrows, separation/attachment lines, and surface restricted particle traces that can be turned on or off in the View portion of Edit->Preferences. This preference ("Use graphics hardware to offset line objects...") is on by default and generally gives good images for everything except move/draw printing. This hardware offset differs from the display offset in that it is in the direction perpendicular to the computer screen monitor (Z-buffer).

Thus, for viewing, you may generally leave the display offset at zero. But for printing, a non-zero value may become necessary so the traces print cleanly.

Time Step Determination — Opens a turn-down area for the specification of time-step parameters.

Min Steps — This field is used to specify the minimum number of integration steps to perform in each element.

Min Angle — If angle between two successive line segments of the Particle trace is less than this value EnSight will double the integration step.

Max Angle — If angle between two successive line segments of the Particle trace is greater than this value EnSight will half the integration step.

Rot Angle — If the change in rotation angle at two successive points of the Particle trace is greater than this value, the integration step is halved.

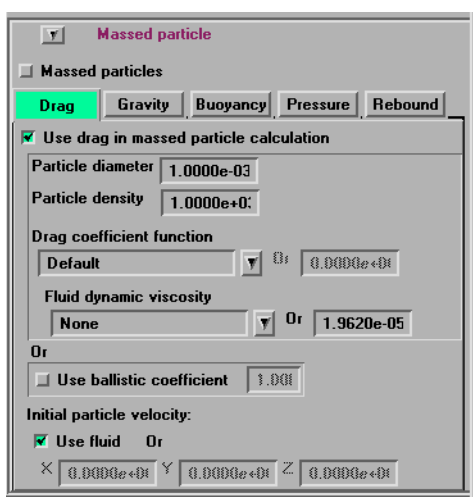
Massed particle — Opens a turn-down area for the specification of massed-particle parameters.

Massed particles — Toggles on/off the massed-particle traces feature. The default is OFF

Note: Some dependent parameters are duplicated under multiple tabs for reference convenience, i.e. Gravity under Gravity and Buoyancy tabs. Note that these parameters are updated under all applicable tabs when changed under a particular tab.

.

Drag term tab — Showing dependent parameters for drag acceleration term (default selection)

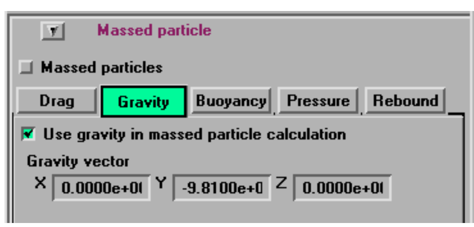


Use drag in massed particle calcuation — Toggles on/off the inclusion of the drag term in the massed-particle computation. The default is ON

Particle diameter	This field specifies the diameter of all particles. The default is 1.e-3.
Particle density	This field specifies the density value of all particles. The default is 1.e+3.
Drag coefficient function	This field specifies the drag coefficient function to be called each time the drag coefficient is calculated. This function defaults to "Default" which essentially defaults to the table described above. Other functions may be accessed via the User-Defined Math Function facility, i.e. DragCoefTable1(Re) (same as default), DragCoefPoly(Re), DragCoefPower(Re). All functions must take the Reynolds Number as their only argument. This parameter only works with the drag term.
Or	This field specifies the drag coefficient value to be used in the computation if "None" is specified as the variable name. The default value is 0. This parameter only works with the drag term.
Fluid dynamic viscosity	This field specifies the fluid dynamic viscosity variable to be used in the massed-particle computation. The default is "None".
Or	This field specifies the fluid dynamic viscosity value to be used in the computation if "None" is specified as the variable name. The default value is 1.9620e-5. This parameter only works with the drag term.
Use ballistic coefficient	Toggles on/off the use of the ballistic coefficient value $(m_p/(a_p c_d))$ in place of the above drag parameters which are greyed-out, i.e. particle diameter and density (used in m_p and a_p), and drag coefficient and fluid dynamic viscosity (used in c_d). The default toggle is OFF. When toggled ON, the default value is 1.
Initial particle velocity	Determines what initial velocity to use for all the particle emitters. The default is Use fluid ON. This parameter only works with the drag term.
Use fluid	Toggles on/off whether all particle emitters should use the fluid velocity at their corresponding locations. The default is ON.
Or X, Y, Z	These fields specify the initial velocity components of all particle emitters. Their default is <0., 0., 0.>.

.

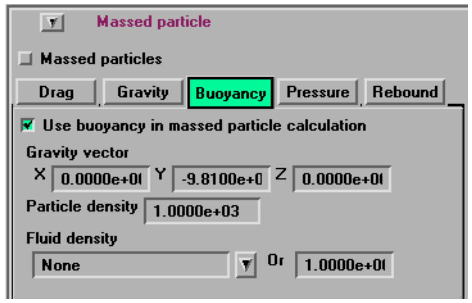
Gravity term tab Showing dependent parameters for gravity acceleration term



Use gravity in massed particle calcuation	Toggles on/off the inclusion of the gravity term in the massed-particle computation. The default is ON
Gravity vector	These fields specify the gravity vector to be applied in the massed-particle computation. The default gravity components are <0., -9.81, 0.>. This parameter only works with the gravity and buoyancy terms.

Buoyancy term tab

Showing dependent parameters for buoyancy acceleration term



Use buoyancy in massed particle calcuation
: Toggles on/off the inclusion of the buoyancy term in the massed-particle computation. The default is ON

Gravity vector
: These fields specify the gravity vector to be applied in the massed-particle computation. The default gravity components are <0., -9.81, 0.>. This parameter only works with the gravity and buoyancy terms.

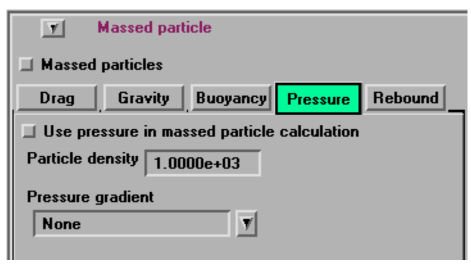
Particle density
: This field specifies the density value of all particles. The default is 1.e+3.

Fluid density
: This field specifies the fluid density variable to be used in the massed-particle computation. The default is “None”. This parameter only works with the buoyancy term.

Or
: This field specifies the fluid density value to be used in the computation if “None” is specified as the variable name. The default value is 1.

Pressure gradient term tab

Showing dependent parameters for pressure-gradient acceleration term



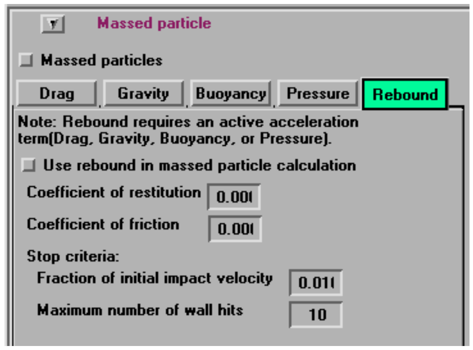
Use pressure in massed particle calcuation
: Toggles on/off the inclusion of the pressure-gradient term in the massed-particle computation. The default is OFF

Particle density
: This field specifies the density value of all particles. The default is 1.e+3.

Pressure gradient
: This field specifies the fluid pressure gradient variable to be used in the massed-particle computation. The default is “None”. This parameter only works with the pressure force term.

.

Rebound term tab Showing dependent parameters for allowing rebound of massed traces



Use rebound in massed particle calcuation — Toggles on/off the inclusion of rebound parameters to account for coefficient of restitution and friction effects when massed traces contact boundaries. The default is OFF

Note: Rebound requires an active acceleration term, i.e. Drag, Gravity, Buoyancy, and/or Pressure.

Coefficient of restitution — This field specifies the coefficient of restitiution value to be used for all massed trace computations. The default value is 0. (no restitution - no rebound). The typical range for this value is 0. to 1. (The value of 1. being full restitution, or a perfect elastic bounce off the wall where the angle of reflection off the boundary into the field is equal to the angle of incidence into the wall from the field.) This value is combined with the coefficient of friction to determine the final rebound of the massed particle off the boundary wall.

Coefficient of friction — This field specifies the coefficient of friction value of the boundary to be used for all massed trace computations. The default value is 0. (no friction). The typical range for this is 0. to <1. This value is combined with the coefficient of restitution to determine the final rebound of the massed particle off the boundary wall.

Fraction of initial impact velocity — This field specifies a fraction of the initial impact velocity (magnitude) value to be used as a stopping criteria for a rebounding massed particle. The default is .01.

Maximum number of wall hits — This field specifies the maximum number of wall hits (per massed particle) value to be used as a stopping criteria for a rebounding massed particle. The default value is 10 hits.

Create — (At the bottom of the Feature Detail Editor) Creates the Particle trace Part in the Graphics Window as specified.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How to Create Particle Traces)

Troubleshooting Particle Traces

Problem	Probable Causes	Solutions
Particle Trace Part is empty.	Velocity is zero.	Change time steps or change location of emitters.

Problem	Probable Causes	Solutions
	Emitter points are outside of flow field.	Change location for emitter points.
	Dataset is 3D and parent Parts are 2D, or dataset is 2D and parent Parts are not planar.	Change parent Parts.
	The created variable selected does not exist for the parent Part(s)	Recreate the variable for the parent Part(s) selected
Streamline is OK, but pathline is empty.	Creating pathline with the emitter emitting at the last time step.	Modify emitter time for the emitter groups.
Particle trace terminates prematurely	Velocity has gone to zero.	None
	Particle has been traced out of the flow field.	None
	Stopping point is at the boundary between two Parts.	Change the parent Parts for the Particle trace to include neighbor Part.
	Particle getting lost and EnSight's search algorithm failing.	Call CEI hotline support.
	Total Time Limit reached.	Change Total Time Limit.
Particle trace exists, then is removed after deleting Parts.	The parent Part for the Particle trace was deleted.	None
Particle trace creation requested, but Particles don't come back.	Requested a large number of Particle traces and/or doing pathlines in large transient dataset.	Be patient.
	Particles are stuck in a recirculation area.	Process will finish when Total Time Limit is reached. Consider terminating job and starting over with a smaller Total Time Limit.
Interactive tracing is slow.	The size of the model and density of the mesh will affect the performance of an interactive trace.	If you can, run on a faster, larger memory workstation. Also, limit if possible the area of interest by cutting the mesh into pieces with the Cut & Split Part editing operation.
Interactive trace does not enter the next Part	Interactive tracing is only done through the Part the emitter resides in.	When you let go of the emitter the full trace will be shown
Surface restricted Particle trace does not appear	Zero velocity at chosen variable offset	Select a Variable offset distance that will give nonzero velocity
	Display offset causing trace to be on opposite side of a surface (hidden surface on)	Change sign of the Display offset
	Emitter does not lie on the surface of selected Parts	Create emitters that lie on the surface
Surface Restricted particle trace does not print well	See Display Offset discussion above	Enter a non-zero display offset.

7.12 Subset Parts Create/Update

EnSight enables you to create and modify Subset Parts from ranges of node and/or element labels of model parts. The Subset Parts feature allows you to isolate contiguous and/or non-contiguous regions of large data sets, and apply the full-range of feature applications and inspection provided by EnSight.

Subset Parts can only be created from parts that have node and/or element labels. Therefore, Subset Parts can not be created from any Created Parts, because the only parts that can have node and element labels are Model Parts such as parts built from file data, Merged Model Parts, or Computational Mesh Model Parts (parts created via the periodic computational symmetry Frame attribute). Model Parts that do not have given or assigned node and/or element labels can not be used to create Subset Parts.

Subset Parts are created and reside on the server. They are Created Parts that provide proper updating of all dependent parts and variables.

Subset Parts are created and modified by specifying parent parts, as well as their node and/or element labels. Node and/or element labels can be displayed and filtered interactively according to global View Mode and local Part Mode attributes.

Clicking once on the Subset Parts Create/Update Icon open the Subset Parts Editor in the Quick Interaction Area which is used to both create and update Subset Parts.

Figure 7-82
Subset Parts Create/Update Icon

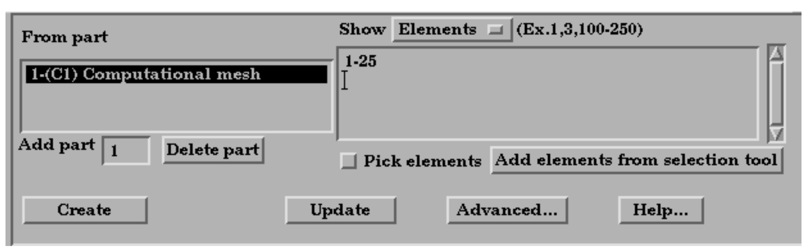


Figure 7-83
Quick Interaction Area - Subset Parts Editor

From Part — List reflecting the parent parts that have been added to the list. Selecting a part in this list displays any corresponding element or node range specifications in the Show List.

Show — Opens a pull-down menu for selecting which type of part entity you wish to include (or have included) in your Subset Part. The Show options are:
Elements show any specified element label ranges
Nodes show any specified node label ranges

Show List — This field specifies the label ranges of Elements and/or Nodes wanted for the Subset Part that correspond to the selected part in the From Part list. The Elements or Nodes are specified as a range as the example indicates, i.e. (Ex. 1,3,8,9,100-250).

Add — This field specifies the GUI part number you wish to add to the From Part list.

Delete — This button removes any selected entries in the From Part list along with any corresponding element or node range specifications in the Show List.

Pick elements — This toggle enables element picking from the graphics window. Elements will be picked using the pick selection which by default is the ‘p’ key.

Add Elements from Selection Tool — Activate the selection tool, and adjust your selection window then click this button to select all the elements within the window.

Update Part — Recreates the Subset Parts selected in the Main Parts List according to the selections in the From Part List and the Show List.

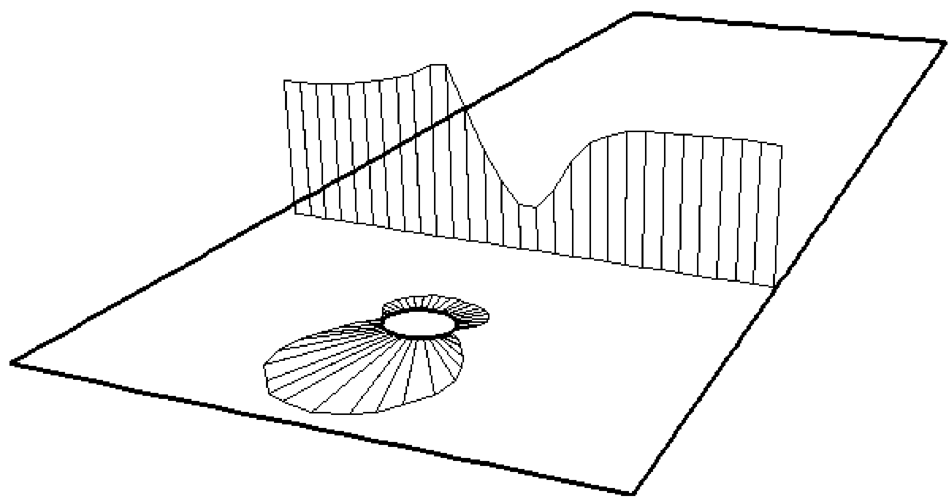
Advanced — Click on the Subset Parts Create/Update Icon to bring up the Feature Detail Editor (Subset Parts), which offers the same features under its the Creation Attributes as offered the Subset Parts as the Quick Interaction Area Editor.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

7.13 Profile Create/Update

Profiles visualize values of a variable along a line with a plot projecting away from the line. Projectors are parallel to a plane, but not necessarily in a plane. Hence, Profile can follow the line.

You can scale and offset projectors. The positive direction is set with the center point of the Plane Tool (away from center point is positive). Consider a base-line (not necessarily straight) along which the value of a variable is known. Moving along this base-line, you can “plot” the value of the variable on an “axis” whose origin moves along the base-line and whose orientation varies so that it is always both perpendicular to the base-line and parallel to a specified *plane* (but not necessarily parallel to a *line*, enabling the plot-line to follow the curve of the base-line in one dimension). A surface joining the base-line to the plot-line is called a *profile.*



The parent Part of a Profile-Part can be a 2D-Clip Line, a Contour, a Particle Trace, or a model Part consisting of a chain of bar elements. From each node of the parent Part, EnSight draws a “projector” whose length is proportional to the value of the variable at the node, and whose orientation makes it (1) parallel to a specified plane, (2) pointing in a direction corresponding to the sign of the variable’s value at the node (with the negative-direction determined by the location of a specified point), and (3) perpendicular to the base-line elements adjoining the node, or, if the base-line bends at the node, oriented so that its projection into the plane defined by the base-line elements will bisect the angle formed by the base-line elements. The outer-end of each projector is connected to those of its neighbors, forming a series of four-sided polygons and hence a surface.

The appearance of the profile depends greatly on the position of the specified sign-direction point (From Point) and the orientation of the specified plane, which you can specify numerically or with the Plane tool. EnSight calculates the projectors using the vector cross-product of the specified-plane’s normal (the Z-axis) and each parent Part element, thus you should orient the plane so that its normal is not parallel to the parent Part elements.

The projector length is calculated by adding to the variable's value an Offset value, then multiplying the sum by a Scaling value. Adding the Offset enables you to shift the zero location of the projectors, which might be useful if you wanted to make all the projectors have the same sign. An offset performs a "shift", but does not change the "shape" of the resulting profile. The Scaling factor changes the displayed size of the profile, a "stretching" type of action. EnSight will provide default values for both factors based on the variable's values at the parent Part's nodes.

Clicking once on the Profile Create/Update Icon opens the Profile Editor the Quick Interaction Area which is used to both create and update (make changes to) profile Parts.

Figure 7-84
Profile Create/Update Icon

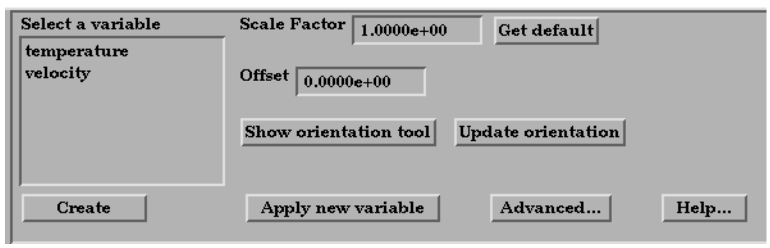


Figure 7-85
Quick Interaction Area - Profile Editor

Scale Factor This field specifies the scaling for magnitude of the projector. The Scale Factor is multiplied times the value of the variable. Values larger than one increase the size and values smaller than one decrease the size.

Offset The value specified in this field is added to the variable values before the Scale Factor is applied to change the magnitude of projectors. Default offset is magnitude of most-negative projector (making them all positive). Has the effect of shifting the plot, but does not change the plot shape.

Get Default Click to set Scale Factor and Offset values to the calculated defaults based on the variable values for the parent Part.

Show Orientation Tool Causes the Plane Tool to become visible in the Graphics Window at the location specified

Update Orientation Recreates the Profile Part at the current location and orientation of the Plane Tool.

Apply New Variable Changes the variable the Profile Part is based on to that currently selected in the Variables List.

Advanced Will open the Feature Detail Editor and select the Profile icon for more advance control of the Profile Part.

Feature Detail Editor (Profiles)

Double clicking on the Profile Create/Update Icon opens the Feature Detail Editor for Profiles, the Creation Attributes Section of which provides access to additional functions for the creation and modification of Profiles:

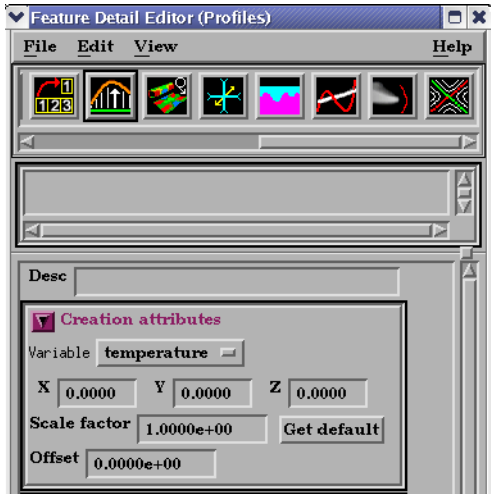


Figure 7-86
Feature Detail Editor (Profiles)

X Y Z

These fields specify the vector components used in creating the Part for vector based or coordinate-based Profiles. These fields are not applicable to Scalar-based Profiles. When all fields are zero, the magnitude of the Variable value is used. If a value other than zero is entered into a field, the sum of $(Vector_X{*}X)+(Vector_Y{*}Y)+(Vector_Z{*}Z)$ is used as the variable value.

Orientation Plane
Pos of C1
Pos of C2
Pos of C3

Specification of the location, orientation, and size of the Plane Clip using the coordinates (in the Parts reference frame) of three corner points, as follows:

Corner 1 is corner located in negative-X negative-Y quadrant
Corner 2 is corner located in positive-X negative-Y quadrant
Corner 3 is corner located in positive-X positive-Y quadrant

Set Tool Coords

Will reposition the Plane Tool to the position specified in C1, C2, and C3.

Get Tool Coords

Will update the C1, C2, and C3 fields to reflect the current position of the Plane Tool.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create Profile Plots)

Troubleshooting Profiles

Problem	Probable Causes	Solutions
The entire profile is not projected the direction you want.	The Plane is not oriented correctly.	Turn on the Plane tool so you can see its orientation. The projectors will be parallel to this plane, so adjust its orientation.
	The From Point is not in a good location	Turn on the Plane tool so you can see the location of the center of the plane. Positive projectors will go away from this point, negative towards.
Portions of the profile appear to be projected in the wrong direction.	The From Point is not in a good location.	Turn on the Plane tool so you can see the location of the center of the plane. Positive projectors will go away from this point, negative towards.
	The normal to the Plane is parallel to some of the elements of the parent Part.	Turn on the Plane tool so you can see its orientation. Try to make sure the Z axis of the Plane tool does not lie parallel to any portions of the parent Part.
	The Parent Part does not contain elements which are consistently ordered	None

7.14 Elevated/Offset Surface Create/Update

Elevated Surfaces visualize the value of a variable by creating a surface projected away from the 2D elements of the parent Part. An Elevated Surface might be used to show the pressure on a 2D surface representing the pressure as height above the 2D surface.

An Offset Surface projects a an origin part into a 3D fluid domain using a single, fixed, translation vector and then interpolates a variables from the 3D domain onto the offset part. For example, an Offset Surface might be used to slightly offset the roof of a car in the vertical direction into the flow field to visualize the flow field velocity just outside of the boundary layer of the curved roof surface. Or, an Offset Surface might be used to translate an origin part into a 3D parent domain and 'clip' the 3D domain using the origin part.

Elevated Surface

First let's look at the Elevated Surface. It is easiest to describe this feature if you think of a planar Part as the parent Part. Now warp this surface up out of plane proportionally to the value of a variable. The resultant surface is an Elevated Surface. Elevated surfaces are to surfaces what Profiles are to lines. While planar surfaces are perhaps the most useful parent Parts to use, parents do not have to be planar. Model Parts containing 2D elements, Clip Planes, Isosurfaces, and even other elevated surfaces are all valid parent Parts.

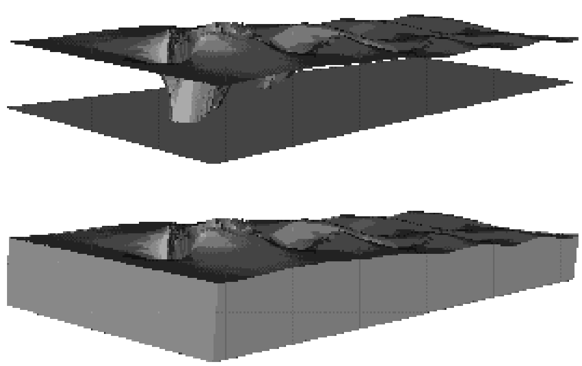


Figure 7-87
Elevated Surface example, with and without Sidewalls

The parent Part is not actually changed, a new surface is created. As this new surface is "raised", projection (Sidewall) elements can be created stretching from the parent to the elevated surface around the boundary of the surfaces if desired. Just the surface, just the sidewalls, or both can be created.

The projection from a node on the parent Part will be in the direction of the normal at the node. If the node is shared by multiple elements, the average normal is used.

The projected distance from a parent Part's node to the corresponding elevated surface node is calculated by adding to the variable's value an Offset value, then multiplying the sum by a Scaling value. Adding the Offset enables you to shift the zero location of the plane. An Offset performs a "shift", but does not change the "shape" of the resulting elevated surface. The Scaling factor changes the distance between parent and elevated surface, a "stretching" effect. EnSight will provide default values for both factors based on the variable's values at the parent Part's nodes.

Offset Surface

An offset surface requires two parts: an origin part and a 3D parent part. The origin part is offset by a single scaled vector into the 3D part and the offset part inherits the variable values of the 3D part at the intersection with the offset part. An example will help. Imagine you have the upper surface of an aircraft composed of 2D elements. The aircraft surface is enclosed within a 3D volume. The origin surface of the aircraft is shifted by the value of a user-supplied constant vector (and scale factor) into the 3D flowfield volume and becomes a new, Offset Part. The new Offset Part now inherits the 3D flowfield volume's variables at the new location of the surface. Offset functionality is effectively clipping a 3D volume using an origin part offset into the volume by a scaled, constant XYZ vector. You cannot scale, warp, or rotate the origin part. You can only translate it.

To use this function you must change the Elevated Surface default pulldown to Offset Surface. Then you must make sure you have selected the 3D volume parent part in the part list, and then enter the origin part in the Quick Interaction Area just above the graphics window.

.

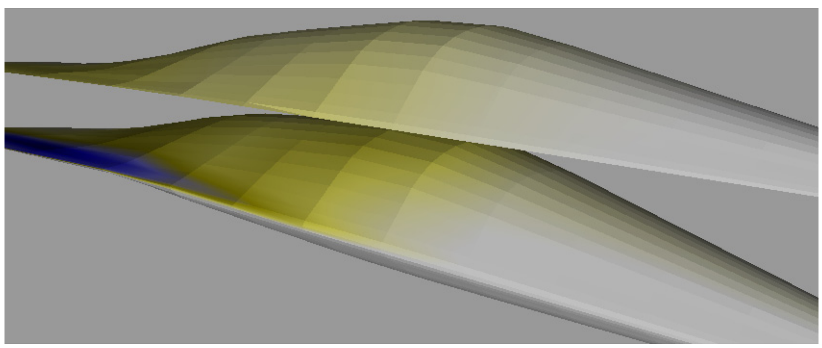


Figure 7-88
Offset Surface example above main model

Clicking once on the Elevated/Offset Surface Create/Update Icon opens the Elevated/Offset Surface Editor in the Quick Interaction Area which is used to both create and update (make changes to) elevated surface Parts.

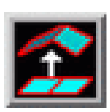


Figure 7-89
Elevated Surface Create/Update Icon

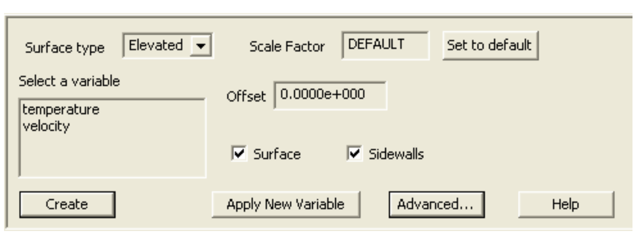


Figure 7-90
Quick Interaction Area - Elevated Surface Editor

Surface Type: Elevated This pulldown chooses between Elevated Surface and Offset surface. Shown below are descriptions of the Elevated Surface options.

Scale Factor This field specifies the scaling for magnitude of distance between the parent Part node and the corresponding elevated surface node. The Factor is multiplied times the value of the variable. Values larger than one increase the size and values smaller than one decease the size. A negative value will have the effect of switching the direction of the projected surface

.

Set to Default Click to set Scale Factor and Offset values to the calculated defaults based on the variable values for the parent Part.

Offset Value specified is added to the variable values before the Scale Factor is applied to change the magnitude of projected distance. Default offset is magnitude of most-negative projection distance (will cause the surface to be projected positively). Has the effect of shifting the surface plot, but does not change the surface plot shape.

Surface Toggle Toggles on/off the creation of the actual elevated surface. The sidewalls alone will be created if this toggle is off.

Sidewalls Toggle Toggles on/off the creation of the sidewalls of the Elevated Surface. Elements will stretch from the parent Part to the Elevated surface around the boundary of the surfaces. The Elevated Surface alone will be created if this toggle is off.

Apply New Variable Changes the variable the Elevated Surface Part is based on to that currently selected in the Variables List.

Advanced Will open the Feature Detail Editor and select the Elevated Surface icon for more advance control of the Elevated Surface Part.

Feature Detail Editor (Elevated Surfaces) Double clicking on the Elevated Surfaces Create/Update Icon opens the Feature Detail Editor for Elevated Surfaces, the Creation Attributes Section of which provides access to all of the functions available in the Quick Interaction Area plus one more:

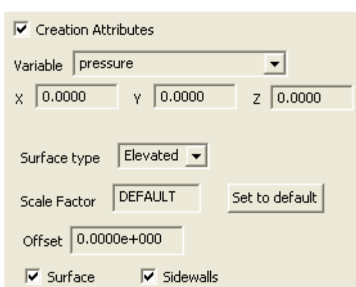


Figure 7-91
Feature Detail Editor (Elevated Surfaces)

X Y Z For vector-based or coordinate-based elevated surfaces, specify vector components used in creating the elevated surface. Not applicable to scalar-type elevated surfaces. Are according to the reference frame of the Elevated Surface-Part. Letters labeling dialog data entry fields depend on type of the reference frame (Rectangular, Spherical, or Cylindrical). If all components are 0.0, the vector or coordinate magnitude will be used.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

Surface Type Offset

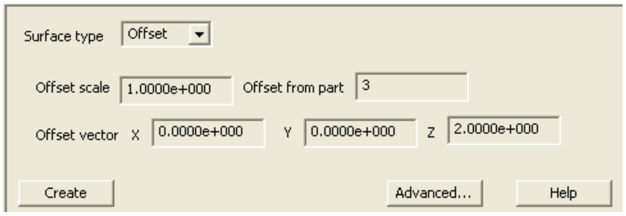


Figure 7-92
Quick Interaction Area - Offset Surface Editor

Alternatively, to use the Offset Surface option, select the 3D volume part(s) in the main part window and choose the pulldown surface type Offset, then set the options as described below.

Surface Type: Offset — This pulldown chooses between Elevated Surface and Offset surface. Shown below are descriptions of the Offset Surface options.

Offset Scale — Scales the offset vector.

Offset Part — This field picks the origin part number that will be used to clip the volume part selected in the main part list.

Offset Vector — These fields are the rigid body translation vector for the entire offset origin part. The origin part cannot be scaled, warped, or rotated. Note: Letters labeling dialog data entry fields depend on type of the reference frame (Rectangular, Spherical, or Cylindrical).

Feature Detail Editor (Offset Surfaces) — Double clicking on the Elevated Surfaces Create/Update Icon opens the Feature Detail Editor for Elevated Surfaces, the Creation Attributes Section of which provides access to all of the functions available in the Quick Interaction Area plus one more:

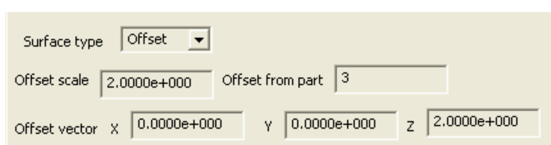


Figure 7-93
Feature Detail Editor (Elevated Surfaces)

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

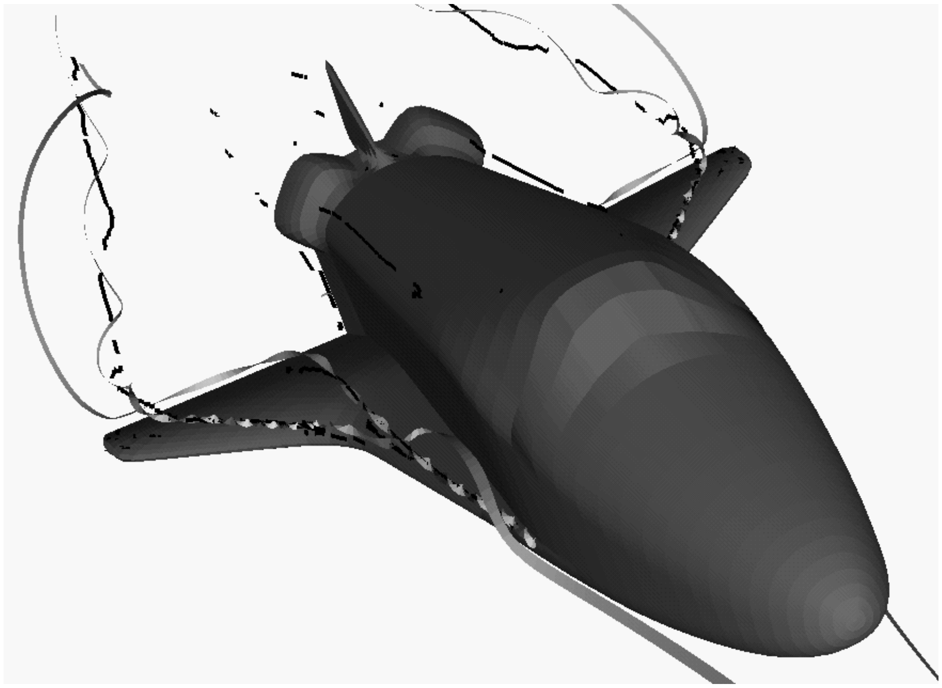
(see How to Create Elevated Surfaces)

Troubleshooting Elevated Surfaces

Problem	Probable Causes	Solutions
The entire Elevated Surface is not projected in the direction you want.		Change the sign of the scale factor.
You have a non-planar parent Part and the elevated surface seems to have strange intersecting elements.	Sidewall elements are not appropriate	Turn off sidewall toggle.
	Scale factor too large.	Lower the Scale Factor.
The Elevated Surface projection appears to be “confused” at various locations.	Inconsistently ordered elements, such that the normals are not “consistent”	Modify element ordering to be consistent, if possible.

7.15 Vortex Core Create/Update

Vortex cores help visualize the centers of swirling flow in a flow field. EnSight creates vortex core segments from the velocity gradient tensor of 3D flow field part(s). Core segments can then be used as emitters for ribbon traces to help visualize the strength and nature of the vortices.



Velocity Gradient Tensor

EnSight automatically pre-computes the velocity gradient tensor for all 3D model parts prior to creating the vortex cores. Since this variable is automatically created, all subsequent 3D model parts created will also have this tensor computed.

Note: The velocity gradient tensor variable will continue to be created and updated for all 3D model parts until it is deactivated.

This tensor variable behaves like any other created tensor variable, and may be deactivated via the Feature Detail Editor (Variables) dialog.

Thresholding

Core segments may be filtered out according to the settings of a threshold variable, value, and relational operator (see Access Clicking once on the Vortex Core Create/Update Icon opens the Vortex Core Editor in the Quick Interaction Area which is used to both create and update (make changes to) the vortex core parts. below for details). Most active variables can be used as threshold variables. Thresholding was implemented to help the user filter-out, or view portions of the core segments according to variable values.

When vortex core parts are Created/Updated, the vorticity magnitude scalar variable "`fx_vortcore_streng`" is created to help you threshold unwanted core segments according to these scalar values. (This is the magnitude (RMS) of the vorticity as defined in chapter 4.)

Due to the difference in algorithms, some segments produced may not be vortex cores (see Caveats). Thus, the need for a filtering mechanism that filters out segments according to different variables arose and has been provided via thresholding options.

Algorithms

Currently, vortex cores are calculated according to two algorithms based on techniques outlined by Sujudi, Haimes, and Kenwright (see References below). Both techniques are linear and nodal. That is, they are based on decomposing finite elements into tetrahedrons and then solving closed-form equations to determine the velocity gradient tensor values at the nodes. Also, any variable with values at element centers are first averaged to element nodes before processing.

The eigen-analysis algorithm uses classification of eigen-values and vectors to determine whether the vortex core intersects any faces of the decomposed tetrahedron. The vorticity based algorithm utilizes the fact of alignment of the vorticity and velocity vectors to determine core intersection points.

References

Please refer to the following references for more detailed explanations of pertinent concepts and algorithms.

D. Banks, B. Singer, “Vortex Tubes in Turbulent Flows: Identification, Representation, Reconstruction”, IEEE Visualization ‘94, 1994

D. Sujudi, R. Haimes, “Identification of Swirling Flow in 3-D Vector Fields”, AIAA-95-1715, Jun. 1995

D. Kenwright, R. Haimes, “Vortex Identification - Applications in Aerodynamics”, IEEE Visualization ‘97, 1997

M. Roth, R. Peikert, “A Higher-Order Method For Finding Vortex Core Lines”, IEEE Visualization ‘98, 1998

R. Haimes and D. Kenwright, On the Velocity Gradient Tensor and Fluid Feature Extraction”, AIAA-99-3288, Jan. 1999

R. Peikert, M. Roth, “The ‘Parallel Vectors’ Operator - a vector field visualization primitive”, IEEE Visualization ‘99, 1999

D. Kenwright, T. Sandstrom, GEL, NASA Ames Research Center, 1999

R. Haimes, D. Kenwright, The Fluid Extraction Toll Kit, Massachusetts Institute of Technology, 2000

R. Haimes, K. Jordan, “A Tractable Approach to Understanding the Results from Large-Scale 3D Transient Simulations”, AIAA-2000-0918, Jan. 2001

Caveats

Due to the linear implementation of both the eigen-analysis and vorticity algorithms, they both have problems finding cores of curved vortices. In addition, testing has shown that both algorithms usually fail to predict vortex core segments in regions of weak vortices. Note: for regions of weak vorticies consider using the Lambda2 or Q-criteria calculator functions (see Section 4.3, Variable Creation).

Since the eigen-analysis method finds patterns of swirling flow, it can also locate swirling flow features that are not vortices (especially in the formation of boundary layers). These non-vortex core type segments can be filtered out via thresholding (see Thresholding). In addition, the eigen-analysis algorithm may produce incorrect results when the flow is under more than one vortex, and has a tendency to produce core locations displaced from the actual vortex core.

The vorticity based method does not seem to exhibit the problem of producing core segments due to boundary layer formations, because the stress components of the velocity gradient tensor have been removed in the formation of the vorticity vector. Thus, the vorticity method seems to produce longer and more contiguous cores - in most cases; and therefore, the reason for including both algorithms.

Access

Clicking once on the Vortex Core Create/Update Icon opens the Vortex Core Editor in the Quick Interaction Area which is used to both create and update (make changes to) the vortex core parts.

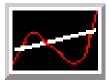


Figure 7-94
Vortex Core Create/Update Icon

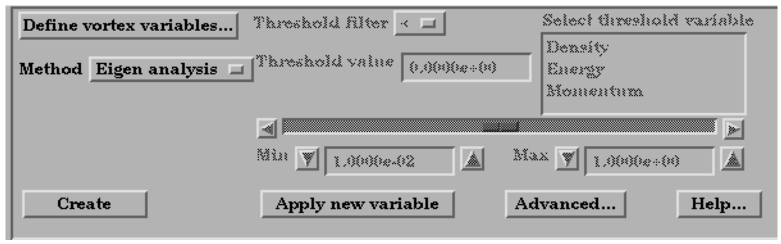


Figure 7-95
Quick Interaction Area - Vortex Core Editor (before Create)

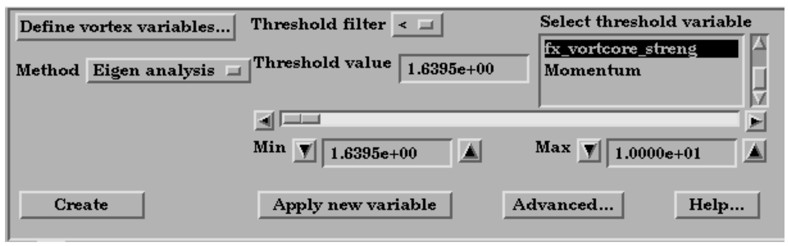
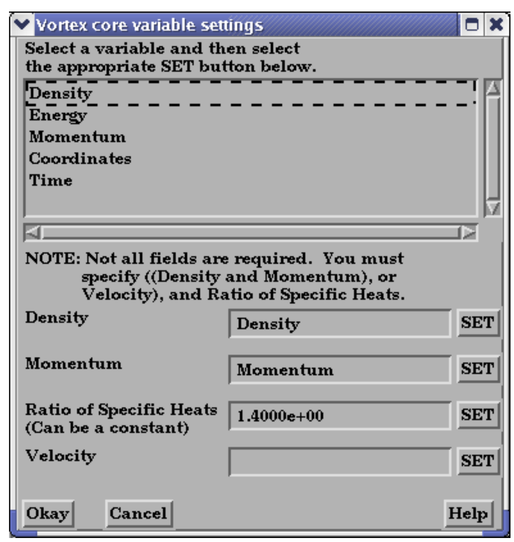


Figure 7-95
Quick Interaction Area - Vortex Core Editor (after Create)

Define Vortex Variables

Opens the Vortex Core Variable Settings dialog which allows the user to identify and set the dependent variables used in computing the vortex cores. This dialog has a list of current accessible variables from which to choose. Immediately below is a list of dependent variables with corresponding text field and SET button. The variable name in the list is tied to a dependent variable below by first highlighting a listed variable, and then clicking the corresponding dependent variables's SET button, which inserts the listed variable into its corresponding text field.



All text fields are required, except you may specify either Density and Momentum (which permits velocity to be computed on the fly), or just Velocity. A default constant value is supplied for the Ratio of Specific Heats which may be changed or specified by a scalar variable name.

Clicking Okay activates all specified dependent variables and closes the dialog.

Method	Opens a pop-up dialog for the specification of which type of method to use to compute the vortex cores in the 3D field. These options are: *Eigen Analysis* — Scheme that uses eigen-analysis on the Velocity gradient tensor to compute the vortex core segments. (See Algorithms above). *Vorticity* — Scheme that uses the vorticity vector from the anti-symmetric portions of the velocity gradient tensor to compute the vortex core segments. (See Algorithms above).
Threshold Filter	Relational operators used to filter out vortex core segments. < Filter out any core segments less than the Threshold Value (default). > Filter out any core segments greater than the Threshold Value.
Threshold Value	The value at which to filter the vortex core segments.
Select Threshold Variable List	A list of possible variables that you may use to help filter out vortex core segments. This list includes the vorticity magnitude scalar variable (named `fx_vortcore_streng`) which gets created when you Create/Update a vortex core part.
Threshold Slider Bar	Used to change the Threshold Value in increments dependent on the Min and Max settings. The stepper button on the left (and right) of the slide bar is used to decrement (and increment) the Threshold value. *Min* — The minimum value of the Threshold Variable. The stepper button on the left (and right) side of the Min text filed is used to decrease (and increase) the order of magnitude, or the exponent, of the min value. *Max* — The maximum value of the Threshold Variable. The stepper button on the left (and right) side of the Max text field is used to decrease (and increase) the order of magnitude, or the exponent, of the Max value.
Create	Creates vortex cores that correspond to the selected 3D field in the part list, based on the respective settings.
Apply New Variable	Applies the threshold settings to the vortex core segments based on the threshold variable that is highlighted in the Select Threshold Variable list.
Advanced	Will open the Feature Detail Editor and select the Vortex Core icon for more advance control of the Vortex Core Part.

Note: Vortex Core feature extraction does not work with multiple cases.

Troubleshooting Vortex Cores

Problem	Probable Causes	Solutions
Error creating vortex cores	Non-3D part selected in part list	Highlight 3D part
Undefined (colored by part color) regions on vortex cores	Vortex core line segment node was not mapped within a corresponding 3D field element	Make sure corresponding 3D field part is defined.

7.16 Shock Surface/Region Create/Update

The Shock Surface/Region feature helps visualize shock waves in a 3D flow field. Shock waves are characterized by an abrupt increase in density, energy, and pressure gradients, as well as a simultaneous sudden decrease in the velocity gradient.

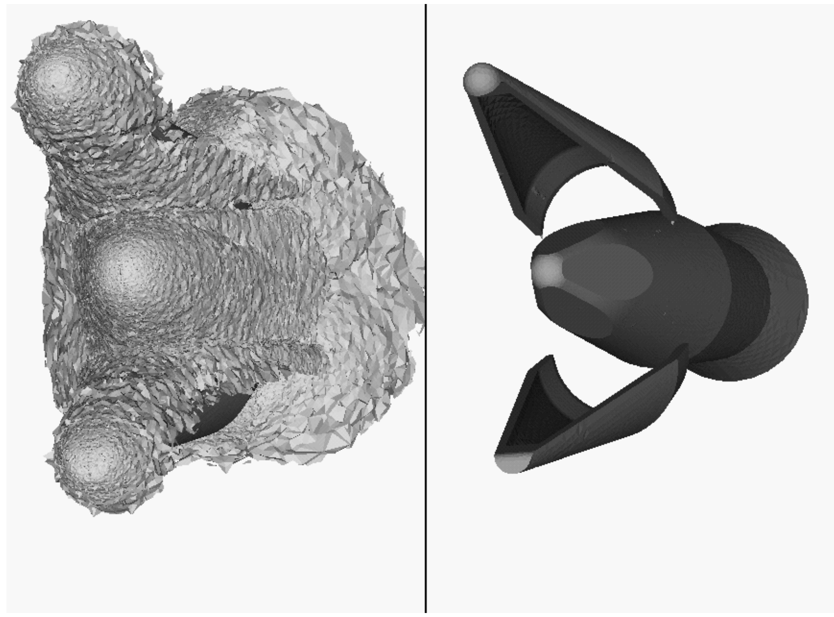


Figure 7-96
Shock Surface (Data Courtesy of Craft Technology)

EnSight creates candidate shock surfaces in 3D (trans/super-sonic) flow fields using a creation scalar variable (i.e. density, pressure) along with the velocity vector (see Algorithms below).

Thresholding

Due to the nature of the shock algorithms, other surfaces with similar characteristics may be produced besides shock surfaces, i.e. expansion regions, etc. Therefore, a filtering mechanism is necessary to help filter out these non-shock regions.

Shock surfaces may be filtered out according to the settings of a threshold variable, value, and relational operator (see Access Clicking once on the Vortex Core Create/Update Icon opens the Vortex Core Editor in the Quick Interaction Area which is used to both create and update (make changes to) the vortex core parts. below for details). Most active variables can be used as threshold variables, but gradients of the density and energy related scalar variables in the streamwise direction seem to work best.

When Shock parts are created via the Surface method, the scalar "`SHK_*`" variable (where * is the appended name of the variable, i.e. `SHK_Density`) is created to help threshold unwanted areas according to these scalar values. When Shock parts are created via the Region method, the scalar "`SHK_Threshold`" variable is created to help threshold respective unwanted areas.

Currently, these `SHK_*` variables consist of the gradient of an appropriate creation variable (i.e. `SHK_Density`, `SHK_Pressure`, etc.) in the streamwise direction. For the Region method, the creation variable is always pressure.

EnSight tries to compute a reasonable default threshold value each time one of these threshold variables is applied. By default this value is half of one exponential order less than the maximum value of the threshold variable on the shock part. This seems to produce a reasonable starting surface for the user to threshold. By default, the smaller the threshold value, the larger the part.

The default threshold variable for non "SHK_" variables is the minimum of the specified variable on the shock part.

The default Min/Max slider values try to bound the default threshold value by appropriate orders of magnitude. Min/Max slider values floor/ceil the min/max values of the threshold variable of the shock part when these ranges are exceeded (see Threshold Slider Bar below).

Algorithms

Shock parts are calculated according to two algorithms, or methods. The first algorithm (referred to by EnSight as the Surface method) is based on the work of Pagendarm et. al., and the second algorithm (referred to by EnSight as the Region method) is based on the work by Haimes et. al. (See References below.)

The Surface method utilizes the maximal gradient of a quantity like density or pressure in the streamwise direction. This yields a surface that requires thresholding to distinguish significant portions of the shock patterns from weak numerical artefacts.

The Region method utilizes flow physics to define shocks in steady state and transient solutions. The steady state equation is based on developing a scalar field based on combining the mach vector with the normalized pressure gradient field. The transient solution combines this term with appropriate correction terms. The Region method produces iso-shock surfaces that form regions that bound the shock wave.

Note: Both methods use dependent variables (See Define Shock Variables below). If some of the dependent variables do not exist and are required, they will be temporarily calculated based on other defined dependent variables (as defined in Section 4.3, Variable Creation). The user has the responsibility to ensure these variables have consistent units.

Both techniques have been implemented in a linear and nodal fashion. That is, their gradient calculations are based on decomposing finite elements into tetrahedrons to approximate the gradient values at the nodes. Also, any variables with values at element centers are averaged to element nodes before processing.

Other Notes

Pre-filtering flow field elements by Mach Number.

The Surface Method allows the user to filter-out any flow field elements less than a specified mach number, by issuing the following command via the command line processor (See Section 2.5, Command Files):

```
test: shock_mach_prefilter #
```

Where # is the corresponding mach-number value (>=0.0) by which to filter. (Zero is the default value - which means this option is turned-off until activated by a value > 0.0.) Ideally this mach-number value would be 1; and thus, would eliminate any subsonic regions from being processed via the Surface method's algorithm. In some transonic cases, this has doubled the efficiency of the algorithm by eliminating the calculation of the second derivative on many elements. Unfortunately, other cases have been observed (especially noticed in regions with normal shack waves) where this option (due to the grid resolution and/or the numerical dissipation inherent in the shock algorithm - see 1999 reference by D. Lovely and R. Haimes) has eliminated some valid shock regions. Although care is taken to provide an appropriate stencil of elements for the gradient calculations of values adjacent to these areas, it appears this value may need to be < 1 to prevent these shock regions from being eliminated. This option is therefore provided at the discretion and expertise of the user. This option only takes effect when issued prior to a create or an update in shock method.

Post-filtering shock part elements by Mach Number.

Both methods allow the user to filter-out (prior to thresholding) any shock part elements less than a specified mach number, by issuing the following command via the command line processor (see Section 2.5, Command Files):

```
test: shock_mach_postfilter #
```

Where # is the corresponding mach-number value (>=0.0) by which to filter. (Zero is the default value - which means this option is turned-off until activated by a value > 0.0.) Ideally this mach number value would be 1; and thus, would eliminate any subsonic regions from being displayed as part of the shock surface. Unfortunately, some cases have been observed (especially noticed in regions with normal shock waves) where this options (due to the grid resolution and/or the numerical dissipation inherent in the shock algorithm - see 1999 reference by D. Lovely and R. Haimes) has eliminated some valid shock regions. This option is therefore provided at the discretion and expertise of the user. This option only takes effect when issued prior to a create or an update in shock method.

Moving Shock.

Both methods compute the stationary shock based on the user specified parameters. The Region Method has the capability of applying a correction term to represent moving shocks in transient cases. This capability is toggled ON/OFF by issuing the following command via the command line processor (see Section 2.5, Command Files).

```
test: toggle_moving_shock
```

Issuing the command a second time will toggle this option off. This option is provided at the discretion and expertise of the user. This option only takes effect when issued prior to a create or an update in shock method.

References

Please refer to the following references for more detailed explanations of pertinent concepts and algorithms.

H.G. Pagendarm, B. Seitz, S.I. Choudhry, “Visualization of Shock Waves in Hypersonic CFD Solutions”, DLR, 1996

D. Lovely, R. Haimes, “Shock Detection from Computational Fluid Dynamics Results”, AIAA-99-3285, 1999, 14th AIAA Computational Fluid Dynamics Conference, Vol 1 technical papers.

R. Haimes and D. Kenwright, “On the Velocity Gradient Tensor and Fluid Feature Extraction”, AIAA-99-3288, Jan. 1999, 14th AIAA Computational Fluid Dynamics Conference, Vol 1 technical papers.

D. Kenwright, T. Sandstrom, GEL, NASA Ames Research Center, 1999

R. Haimes, D. Kenwright, The Fluid Extraction Tool Kit,
Massachusetts Institute of Technology, 2000, 39th Aerospace Sciences Meeting and Exhibit, Reno.

R. Haimes, K. Jordan, “A Tractable Approach to Understanding the Results from Large-Scale 3D Transient Simulations”, AIAA-2000-0918, Jan. 2001

Access

Clicking once on the Shock Surface/Region Create/Update Icon opens the Shock Editor in the Quick Interaction Area which is used to both create and update (make changes to) the shock part.

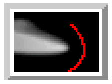
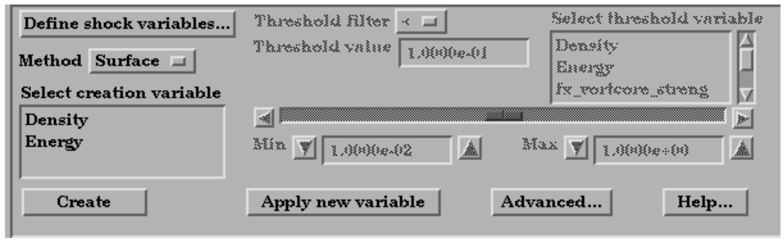


Figure 7-97
Shock Surfaces/Regions Create/Update Icon



Quick Interaction Area - Shock Surfaces/Regions Editor (before Create)

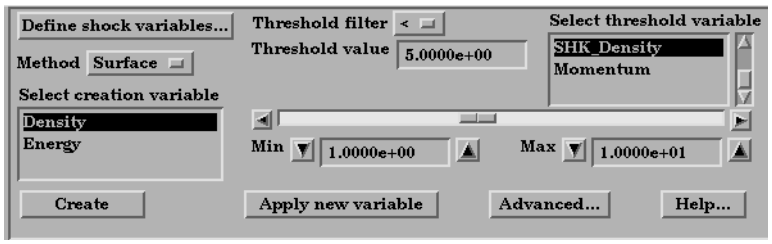
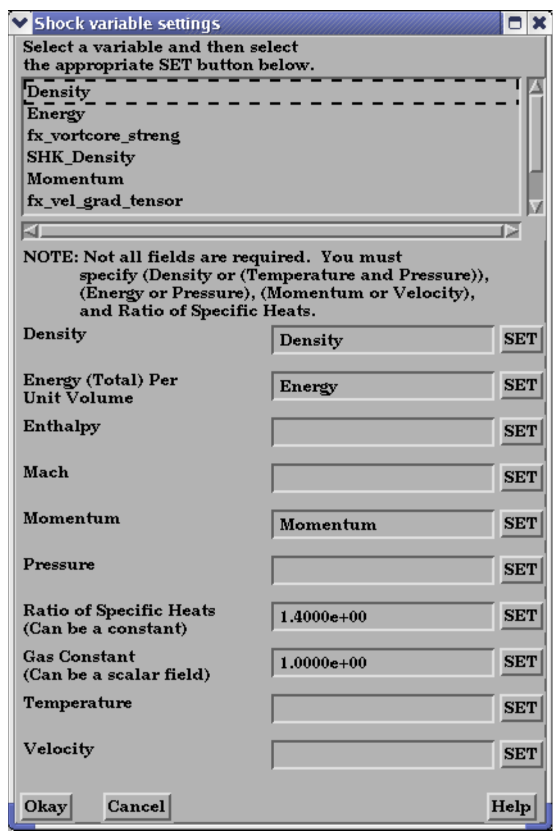


Figure 7-98
Quick Interaction Area - Shock Surfaces/Regions Editor (after Create)

Define Shock Variables...

Opens the Shock Variable Settings dialog which allows the user to identify and set the dependent variables used in computing the shock parts. This dialog has a list of current accessible variables from which to choose. Immediately below is a list of dependent variables with corresponding text field and SET button. The variable name in the list is tied to a dependent variable below by first highlighting the listed variable, and then clicking the corresponding dependent variable's SET button, which inserts the listed variable into its corresponding text field.



Not all text fields are required. Although you must specify either Density or Pressure, Temperature, and Gas Constant; either Energy or Pressure; either Velocity or Momentum; and the Ratio of Specific Heats. A default constant value is supplied for the Ratio of Specific Heats and the Gas Constant which may be changed or specified by a scalar variable name.

Clicking Okay activates all specified dependent variables and closes the dialog.

Method	Opens a pop-up dialog for the specification of which type of method, to use to compute the vortex cores in the 3D field. These options are: *Surface* Scheme that uses maximal density or pressure gradients in the streamwise direction to locate candidate shock surfaces. (See Algorithms above). *Region* Scheme that uses flow physics based on the mach vector coupled with pressure gradient to locate candidate shock regions. (See Algorithms above.)
Select Creation Variable	A list of variables used to create the shock surface via Surface method. These variable are specified via those SET in the Define Shock Variables list above. *Note: This list is not used for the Region method. The Region method only uses pressure as the creation variable.*
Threshold Filter	Relational operators used to filter out shock areas. < Filter out any areas less than the Threshold Value (default). > Filter out any areas greater than the Threshold Value.
Threshold Value	The value at which to filter the shock areas.
Select Threshold Variable List	A list of possible variables that you may use to help filter out unwanted areas. This list includes the shock threshold variables "`SHK_*`" which gets created when you Create/ Update a shock part.
Threshold Slider Bar	Used to change the Threshold Value in increments dependent on the Min and Max settings. The stepper button on the left (and right) of the slide bar is used to decrement (and increment) the Threshold value. *Min* The minimum value of the Threshold Variable. The stepper button on the left (and right) side of the Min text field is used to decrease (and increase) the order of magnitude, or the exponent, of the Min value. *Max* The maximum value of the Threshold Variable. The stepper button on the left (and right) side of the Max text field is used to decrease (and increase) the order of magnitude, or the exponent, of the Max value.
Create	Creates shock parts that correspond to the selected 3D field in the part list, based on the respective settings.
Apply New Variable	Applies the threshold settings to shock surfaces based on the threshold variable that is highlighted in the Select Threshold Variable list.
Advanced	Will open the Feature Detail Editor and select the Shock Surface icon for more advance control of the Shock Surface Part.

Note: Shock Surface feature extraction does not work with multiple cases.

Troubleshooting Shock Surfaces/Regions

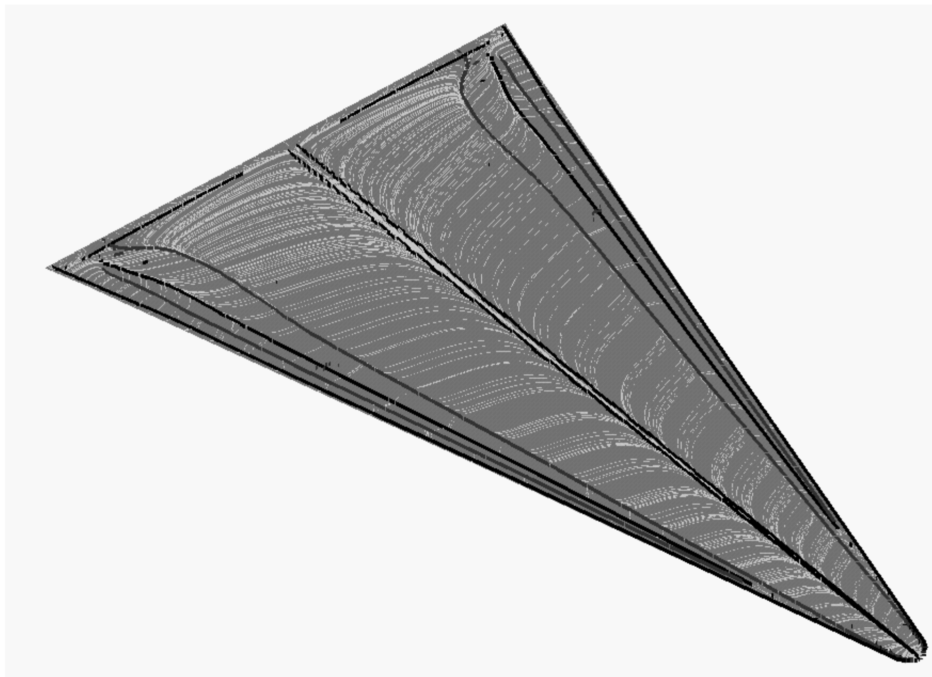
Problem	Probable Causes	Solutions
Error creating shock part	Non-3D part selected in part list	Highlight 3D flow field part
No shock part created	Flow field part subsonic	No shock in subsonic regions

Problem	Probable Causes	Solutions
	Shock dependent variables defined with incorrect units, i.e. since Region method uses density and mach, if file variables are pressure, temperature, and velocity, then density (and thus mach) is dependent on gas constant. By default this value is 287 (Nm/KgK)	Make sure dependent variables have correct units. i.e. gas constant may need to be 1716(ft-lb/slugDegR), or some other value rather than the default
No to little shock part created	Threshold value too large for < operation	Decrease threshold value

7.17 Separation/Attachment Lines Create/Update

Separation and Attachment Lines exist on 2D surfaces and help visualize areas where flow abruptly leaves or returns to the 2D surface in 3D flow fields. These lines are topologically significant curves on the 2D surface where flow converges and then separates (separation lines) from the surface into the 3D flow field, and where flow attaches and then diverges (attachment lines) to the surface from the 3D flow field.

These line segments can be used as emitters for ribbon traces to help visualize flow interaction from the 2D surface into the 3D field, or displayed along with surface-restricted traces to help visualize the topology of the 2D surface.



EnSight creates separation and attachment lines as two distinct parts so that each may be assigned their own attributes. Although both are updated computationally when changes are made to either one via the quick interaction area.

Separation/Attachment lines can be created on any 2D part, whether it is a boundary surface or internal surface to a 3D flow field. These lines can also be created on 3D flow field parts. However, computation of the separation/attachment lines is restricted to only the boundary surfaces of the 3D flow field.

Velocity Gradient Tensor

Ensight creates separation and attachment lines from the velocity gradient tensor of the 3D flow field part. EnSight automatically pre-computes the velocity gradient tensor for all 3D model parts prior to creating the separation and attachment lines. These values are then mapped to any corresponding 2D model part, or inherited by any created part.

Since this variable is automatically created, all subsequent 3D model parts created will also have this tensor variable computed.

Note: The velocity gradient tensor variable will continue to be created and updated for all 3D model parts until it is deactivated.

This tensor variable behaves like any other created tensor variable, and may be deactivated via the Feature Detail Editor (Variables) dialog.

Thresholding

Separation/Attachment lines may be filtered out according to the settings of a threshold variable, value, and relational operator (see Access Clicking once on the Vortex Core Create/Update Icon opens the Vortex Core Editor in the Quick Interaction Area which is used to both create and update (make changes to) the vortex core parts. below for details). Most active variables can be used as threshold variables. Thresholding was implemented to help the user to filter-out, or view portions of the line segments according to variable values.

When separation and attachment line parts are Created/Updated, the scalar variable “`fx_sep_att_strength`” is created to help you threshold unwanted core segments according to these scalar values.

Note: This scalar variable is currently set to the vorticity magnitude scalar, until a better thresholding variable can be identified.

Since it has been observed that the current implementation of this algorithm may produce additional lines that are not separation or attachment lines, the need for a filtering mechanism that filters out segments according to different variables arose and had been provided via thresholding options.

Algorithms

Currently, separation and attachment lines are calculated according to the phase-plane algorithm presented by Kenwright (see References below). This algorithm detects both closed and open separation. Closed separation lines originate and terminate at critical points. Whereas open separation lines do not need to start or end at critical points.

This technique is linear and nodal. That is, 2D elements are decomposed into triangles, and then closed-form equations are solved to determine the velocity gradient tensor values for eigen-analysis at the nodes. Also, any variables with values at element centers are averaged to element nodes before processing.

References

Please refer to the following references for more detailed explanations of pertinent concepts and algorithms.

J. Helman, L. Hesselink
“Visualizing Vector Field Topology in Fluid Flows”,
IEEE CG&A, May 1991

D. Kenwright, “Automatic Detection of Open and Closed Separation and Attachment Lines”, IEEE Visualization ‘98, 1998, pp. 151-158

R. Haimes and D. Kenwright, “On the Velocity Gradient Tensor and Fluid Feature Extraction”, AIAA-99-3288, Jan. 1999, pp. 315-324

S. Kenwright, C. Henze, C. Levit, “Feature Extraction of Separations and Attachment Liens”, IEEE TVCG, Apr.-Jun. 1999, pp. 135-144

R. Peikert, M. Roth, “The ‘Parallel Vectors’ Operator - a vector field visualization primitive”, IEEE Visualization ‘ 99, 1999

D. Kenwright, T. Sandstrom, GEL, NASA Ames Research Center, 1999

R. Haimes, D. Kenwright, The Fluid Extraction Tool Kit,
Massachusetts Institute of Technology, 2000

Access

Clicking once on the Separation and Attachment Lines Create/Update Icon opens the Separation and Attachment Lines Editor in the Quick Interaction Area which is used to both create and update (make changes to) the separation and attachment line parts.

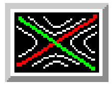


Figure 7-99
Separation/Attachment Lines Create/Update Icon

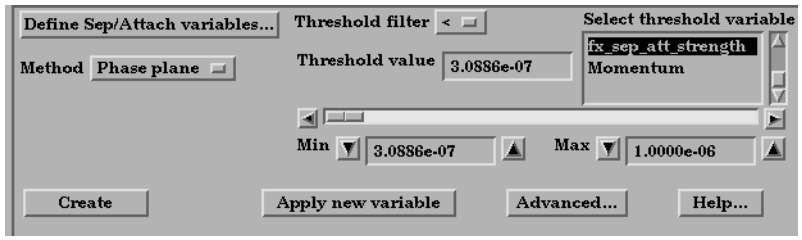
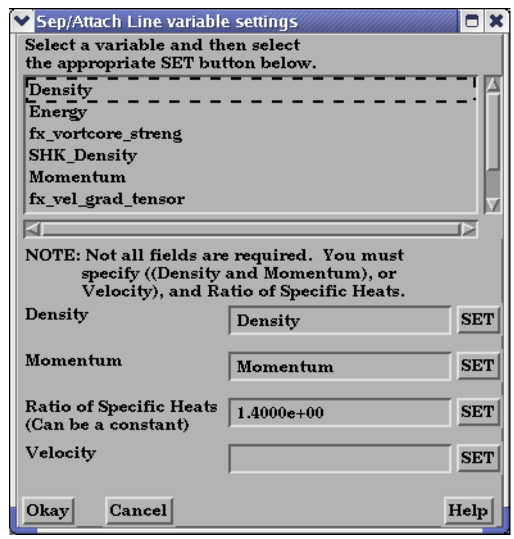


Figure 7-100
Quick Interaction Area - Separation/Attachment Lines Editor

Define Sep/Attach Variables...

Opens the Sep/Attach Line Variable Settings dialog which allows the user to identify and set the dependent variables used in computing separation and attachment lines. This dialog has a list of current accessible variables from which to choose. Immediately below is a list of dependent variables with corresponding text field and SET button. The variable name in the list is tied to a dependent variable below by first highlighting a listed variable, and then clicking the corresponding dependent variable's SET button, which inserts the listed variable into it's corresponding text field.



All text fields are required, except you may specify either Density and Momentum (which permits velocity to be computed on the fly), or just Velocity. A default constant value is supplied for the Ratio of Specific Heats which can be changed or specified by a scalar variable name.

Clicking Okay activates all specified dependent variables and closes the dialog.

Method

Opens a pop-up dialog for the specification of which type of method, to use to compute the separation and attachment lines on the 2D surface. These options are:

Phase Plane - Scheme that uses eigen-analysis on the velocity gradient tensor along with phase plane analysis to compute the separation and attachment line segments (see Algorithms).

Threshold Filter

Relational operators used to filter out line segments.

< Filter out any line segments less than the Threshold Value (default).
> Filter out any line segments greater than the Threshold Value.

Threshold Value

The value at which to filter the line segments.

Select Threshold Variable List

A list of possible variables that you may use to help filter out line segments. This list includes the vorticity magnitude scalar variable (named fx_sep_att_strength) which gets created when you Create/Update a separation and attachment part.

Threshold Slider Bar — Used to change the Threshold Value in increments dependent on the Min and Max settings. The stepper button on the left (and right) of the slide bar is used to decrement (and increment) the Threshold value.

Min — The minimum value of the Threshold Variable. The stepper button on the left (and right) side of the Min text filed is used to decrease (and increase) the order of magnitude, or the exponent, of the min value.

Max — The maximum value of the Threshold Variable. The stepper button on the left (and right) side of the Max text field is used to decrease (and increase) the order of magnitude, or the exponent, of the Max value.

Create — Creates separation and attachment lines that correspond to the selected 2D part in the part list, based on the respective settings.

Apply New Variable — Applies the threshold settings to the separation and attachment line segments based on the threshold variable that is highlighted in the Select Threshold Variable list.

Advanced — Will open the Feature Detail Editor and select the Separation/Attachment icon for more advance control of the Separation/Attachment Part.

Feature Detail Editor (Separation/Attachment) — Double clicking on the Separation/Attachment Lines Icon opens the Feature Detail Editor (Separation/Attachment Lines), the Creation Attributes Section of which provides access to the same functions available in the Quick Interaction Area, as well as one more.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

Display offset — This field specifies the normal distance away from a surface to display the separation/attachment lines. A positive value moves the lines away from the surface in the direction of the surface normal.

Please note that there is a hardware offset that will apply to contours, vector arrows, separation/attachment lines, and surface restricted particle traces that can be turned on or off in the View portion of Edit->Preferences. This preference ("Use graphics hardware to offset line objects...") is on by default and generally gives good images for everything except move/draw printing. This hardware offset differs from the display offset in that it is in the direction perpendicular to the computer screen monitor (Z-buffer).

Thus, for viewing, you may generally leave the display offset at zero. But for printing, a non-zero value may become necessary so the lines print cleanly.

Note: Separation and Attachment Line feature extraction does not work with multiple cases.

Troubleshooting Separation/Attachment Lines

Problem	Probable Causes	Solutions
Error creating separation and attachment lines	Invalid part selected in part list	Highlight 2D or 3D part
Undefined (colored by part color) regions on sep/attach lines	Sep/Attach line segment node was not mapped within a corresponding 3D field element	Make sure corresponding 3D field part is defined.
Separation/attachment lines do not print well.	See Display Offset discussion above	Enter a non-zero Display Offset

7.18 Boundary Layer Variables Create/Update

EnSight creates the following Boundary Layer Variables simultaneously on a 2D boundary part directly from velocity information of its corresponding 3D flow field part. Their corresponding variable names are included in all appropriate EnSight variable lists, i.e. Color Parts variable list, etc.

Variable Name	Description	Symbol
(N) bl_thickness	Boundary layer thickness	δ
(N) bl_disp_thickness	Displacement thickness	δ^*
(N) bl_momen_thickness	Momentum thickness	Θ
(N) bl_shape_parameter	Shape parameter	H
(N) bl_skin_friction_Cf	Skin friction coefficient	C_f

Only nodal (values per node) variables are created. Any dependent elemental variables (values per element) are averaged to nodal variables before processing. (See Definitions below.)

Whether these variables are mapped onto the 2D boundary part, or used in conjunction with other EnSight features (such as Elevated Surfaces of the boundary layer thickness off the 2D boundary part, Vortex Cores, Separation and Attachment Lines, Shock, etc.), these variables help provide valuable insight into the formation and location of possible boundary layers.

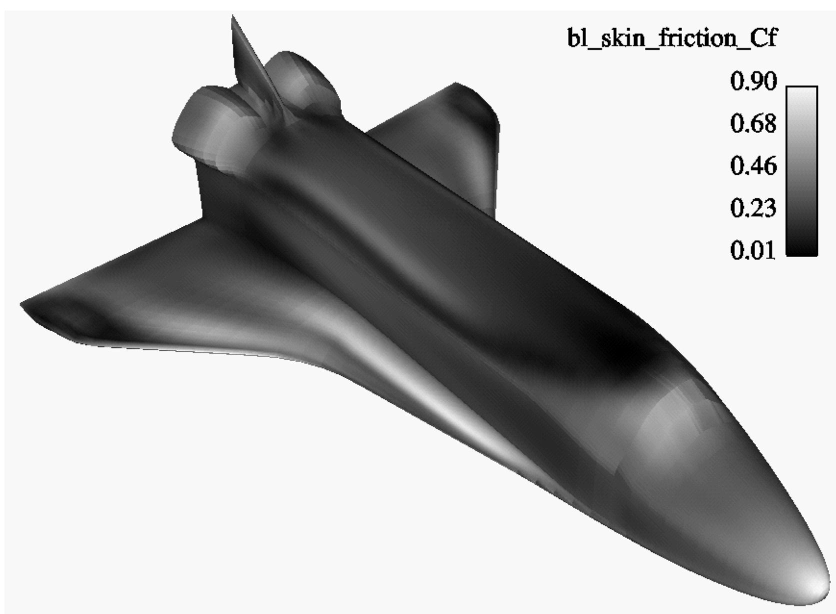


Figure 7-101
Skin Friction Coefficient

Boundary Layer

A boundary layer is a relatively thin region that confines viscous diffusion near the surface of a flow field, where the velocity gradient in the normal direction to the surface goes through an abrupt change. Although multiple boundary layers may be considered (especially in areas of flow separation), our current implementation provides boundary layer parameters based on the former concept. In these thin regions, the thickness of the boundary layer typically increases in the downstream direction, and the velocity parallel to the surface is much larger than the velocity normal to the surface.

Boundary Surfaces

Boundary parts are typically 2D surface part(s) that correspond to a 3D field. These surfaces may either be boundary parts defined directly from the data file, or created parts (i.e. 2D IJK sweeps of a structured part, or an isosurface of zero velocity of either an unstructured or structured part).

Velocity-Magnitude Gradient Vector

Changes of the velocity in the normal direction from the surface into the 3D flow field are utilized to determine the boundary layer. EnSight automatically creates a velocity-magnitude gradient vector for all 3D model parts prior to creating the boundary layer variables. These gradient values are then mapped to all corresponding 2D model parts, and inherited by all created parts.

Note: The velocity-magnitude gradient vector variable will continue to be created for all 3D model parts until it is deactivated.

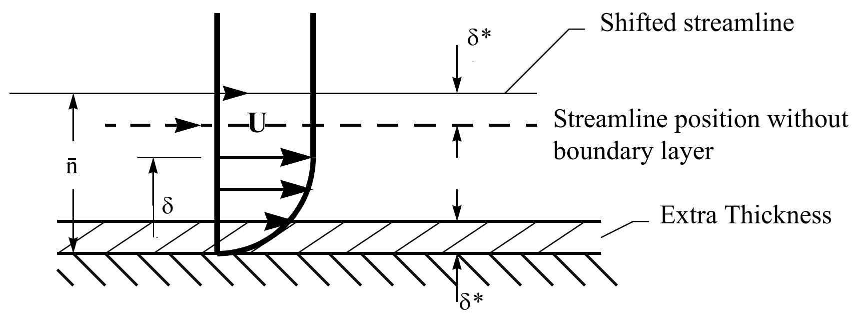
This vector variable behaves like any other created variable, and may be deactivated via the Feature Editor (Variables) dialog.

Definitions

Boundary Layer Thickness

$$\delta = \bar{n}\big|_{u/U = 0.995}$$

The distance normal from the surface to where u/U = 0.995,
where: u = magnitude of the velocity at a given location in the boundary layer,
U = magnitude of the velocity just outside the boundary layer.



Displacement Thickness

$$\delta^* = \frac{1}{U}\int_0^{\delta}(U - u)\,dn$$

Provides a measure for the effect of the boundary layer on the “outside” flow. The boundary layer causes a displacement of the streamlines around the body.

Momentum Thickness

$$\Theta = \frac{1}{U^2}\int_0^{\delta}(U - u)u\,dn$$

Relates to the loss of momentum in the air in the boundary layer.

Shape Parameter

δ^*/Θ

Used to characterize boundary layer flows, especially to indicate potential for separation.

This parameter increases as a separation point is approached, and varies rapidly near a separation point.

Note: Separation has not been observed for H < 1.8, and definitely has been observed for H = 2.6; therefore, separation is considered in some analytical methods to occur in turbulent boundary layers for H = 2.0.

In a Blasius Laminar layer (i.e. flat plate boundary layer growth with zero pressure gradient), H = 2.605. Turbulent boundary layer, H ~= 1.4 to 1.5, with extreme variations ~= 1.2 to 2.5.

Skin Friction Coefficient

$$C_f = \frac{\tau_w}{0.5\rho_\infty (V_\infty)^2}$$

where: $\tau_w = \mu\left(\frac{\partial u}{\partial n}\right)_{n=0}$ = fluid shear stress at the wall.

μ = dynamic viscosity of the fluid.

May be spatially and/or temporarily varying quantity (usually a constant).

n = distance normal to the wall.

ρ_∞ = freestream density

V_∞ = freestream velocity magnitude.

This is a non-dimensionalized measure of the fluid shear stress at the surface. An important aspects of the Skin Friction Coefficient is:

$C_f = 0$, indicates boundary layer separation.

Other Notes:

Factor Determining Velocity at Boundary-Layer Thickness (δ)

The factor (default = 0.995) which determines the velocity magnitude (u) at the boundary-layer thickness (δ) with respect to the velocity magnitude (U) just outside the boundary layer (i.e. δ is the distance normal to the surface at which u = 0.995U), may be changed by issuing the following command via the command line processor (see Section 2.5, Command Files):

```
test: blt_factor #
```

where # is the corresponding factor (> 0.).

References

Please refer to the following texts for more detailed explanations.

P.M. Gerhart, R.J. Gross, & J.I. Hochstein, Fundamentals of Fluid Mechanics, 2nd Ed., (Addison-Wesley: New York, 1992),

C.A.J. Fletcher, Computational Techniques for Fluid Dyanmics, Vol. 2, 2nd Ed., (Springer: New York, 1997)

Access

Clicking once on the Boundary Layer Variable Create/Update Icon opens the Boundary Layer Variables Editor in the Quick Interaction Area, which is used to both create and update (make changes to) the boundary layer variables.

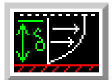


Figure 7-102
Boundary Layer Variables Create/Update Icon

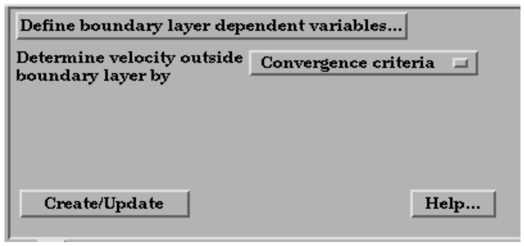


Figure 7-103
Quick Interaction Area - Boundary Layer Variables Editor

Define Boundary Layer Dependent Variables...

Opens the Boundary Layer Variable Settings dialog which allows the user to identify and set the dependent variables used in computing the boundary layer variables (see Definitions above). This dialog has a list of current accessible variables to choose from. Immediately below is a list of dependent variables with corresponding text field and SET button. The variable name in the list is tied to a dependent variable below by first highlighting a the listed variable, and then clicking the corresponding dependent variable's SET button, which inserts the listed variable into its corresponding text field.

All text fields are required, except you may specify either Density and Momentum (which permits velocity to be computed on the fly), or Velocity. Default constant values are provided which may be changed by editing the text field.

Clicking Okay activates all specified dependent variables and closes the dialog.

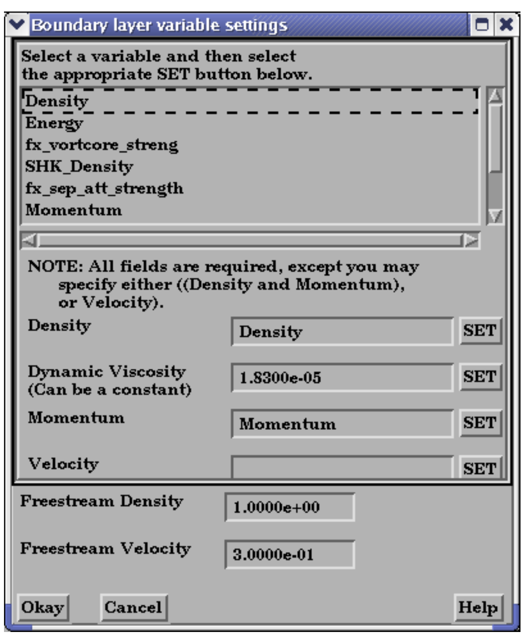


Figure 7-104
Boundary Layer Variable Settings Dialog

Freestream Density

Constant 'upstream' density value (near flow inlet). Only used for skin-friction coefficient, Cf.

Freestream Velocity — Constant ‘upstream’ velocity magnitude value (near flow inlet). Only used for skin-friction coefficient, Cf.

Determine Velocity Outside Boundary Layer By — Opens a pop-up dialog for the specification of which type of method to determine the constant velocity just outside the boundary layer (U) (see Definitions above). The following options determine (U) at each node of the surface in the direction normal from the surface into the 3D field by:

Convergence Criteria - monitoring the velocity profile until either the velocity magnitude goes constant or its gradient goes to zero.

Distance From Surface - specifying the Normal Distance from the surface into the field at which to extract the velocity and assign as U. Then monitor the velocity profile from the surface into the field until U is obtained.

Normal Distance - Text field that contains the distance normal from the surface into the 3D field at which to extract the velocity for U.

Velocity Magnitude - specifying the Velocity Magnitude to assign as U. Then monitor the velocity profile from the surface into the field until U is obtained.

Velocity Magnitude - Text field that contains the specified velocity magnitude to assign as U.

Note: Boundary Layer Variable feature extraction does not work with multiple cases.

Troubleshooting Boundary Layer Variables

Problem	Probable Causes	Solutions
Error creating boundary layer variables.	Non-2D part selected in part list.	Highlight 2D part.
Undefined (colored by part color) regions on boundary surface.	2D boundary surface node was not mapped to corresponding 3D field boundary node.	Make sure corresponding 3D field part is defined.

7.19 Material Parts Create/Update

EnSight enables you to create and modify Material Parts from material descriptions defined on model parts. The Material Parts feature allows you to extract single or combined regions of specified materials, as well as boundary interfaces between two or more specified materials.

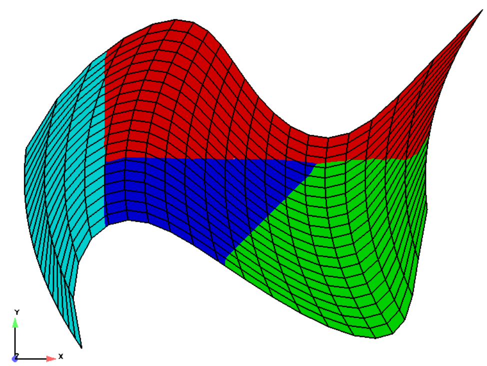


Figure 7-105
Material Parts Illustration

Material Parts can only be created from model parts that have material ids assigned to them. Therefore, Material Parts can not be created from any Measured or Created Parts. In addition, material information is not transferred to Created Parts.

Material Parts are created and reside on the server. They are Created Parts that provide proper updating of all dependent parts and variables - except they do not inherit any material data themselves.

Material Parts are created and modified by specifying parent model parts, as well as selecting material descriptions listed in the Materials List. A Material Part is extracted from only 2D and 3D elements. A Material Part is created as either a Domain or an Interface.

Domain A material Domain defines a solid region composed of one or more specified materials. Parts with 2D elements yield 2D material elements, and parts with 3D elements yield 3D material elements.

Interface A material Interface defines a boundary region between at least two or more adjacent specified materials. Parts with 2D elements yield 1D material elements, and parts with 3D elements yield 2D material elements.

Null Materials Two categories of materials are reflected in the Materials List; namely, given materials and a "null_material". All given material descriptions correspond to a material assigned a positive material number, or id. Any material that has an id less than or equal zero (<= 0) is grouped under the "null_material" and assigned the material id of zero (0). This allows the null material to act as a valid material. The "null_material" description always appears in the Materials List whether or not there are any null materials.

Formats

Materials may be defined either by the three sparse files (i.e. material ids, mixed ids, and mixed values), or materials may be designated as a set of per element scalar variable descriptions; but not by both. See Section 11.1, EnSight Gold Casefile Format for format details.

Algorithm

Two algorithms are implemented to compute the material part: the “smooth” algorithm and the Young’s algorithm. The “smooth” algorithm is based on a probability based approach to material interface reconstruction (see reference below). Essentially volume fractions are averaged for every cell to its nodes, edges/faces, and center. Each cell is then decomposed and/or subdivided into subcells. Each subcell is then repeatedly assigned, compared, and appropriately interpolated with volume fractions for each material. The resulting material cells reflect the maximum volume fraction portion(s) of the interpolated subcells.

The Young’s algorithm partitions each mixed-material cell into regions which exactly match the material fractions. The partitioning is based on an orientation vector that determines the diretion of the lines (or planes) used to subdivide the cell. Materials are sliced off in the order assigned by the user.

Reference

Meredith, Jeremy S. “A Probability Based Approach to Material Interface Reconstruction for Visualization”, ECS277 Project 4, Spring 2001

D.L. Young, “Time-dependent multi-material flow with large fluid distortion,” in Numerical Methods for Fluid Dynamics (K. W. Morton and J. J. Baines, eds.), pp. 273-285, Academic Press, 1982.

Caveats

Material resolution tends to diminish (and at times distorts) at boundary cells that lack adjacent ghost cells. The volume fractions at these cells simply lack the proper weighting. This is remedied by providing material ghost cells.

Thus, materials that contribute half or less of the total portion on a boundary element, typically do not appear without ghost cells.

Specie(s)

Species may also be associated to a material (see MATERIAL Section under EnSight Gold Case File Format and EnSight Gold Material Files Format), but are not involved in any of the computational aspects of creating/updating a material part. Rather, a material species variable may be created via the new variable calculator (see MatSpecies under 4.3 Variable Creation).

Note: Species are only supported with the three sparse material files format, and are not supported by the materials as scalars per element format.

Access

Clicking once on the Material Parts Create/Update Icon opens the Material Parts Editor in the Quick Interaction Area which is used to both create and update (make changes to) the material parts.

Figure 7-106
Material Parts Create/Update Icon

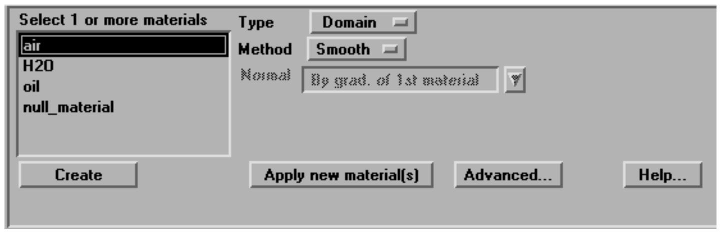


Figure 7-106
Quick Interaction Area - for Material Parts Type Domain/Interface

Materials List List reflecting the available materials in the model parts. Any material that has an id less than or equal to zero (<= 0) comprises the "null_material".

Type Opens a pull-down menu for specification of whether the Material Part results in a Domain or Interface. Changing the Type of existing Material Parts will automatically update them to the new specified type.

Domain Creates a solid region composed of one or more specified materials. Parts with 2D elements yield 2D material elements, and parts with 3D elements yield 3D material elements.

Interface Creates a boundary region between at least two or more specified materials. Parts with 2D elements yield 1D material elements, and parts with 3D elements yield 2D material elements.

Method Opens a pull-down menu for specification of the algorithm. Currently, use the Smooth to compute the material part. Changing the method of existing Material Parts will automatically update them to the new specified method.

Smooth Create/update a material part via the smooth algorithm (default)

Young's Create/update a material part via the Young's algorithm

Normal Not available for Smooth algorithm. Opens a pull-down menu to select the method for computing the orientation vector for Young's algorithm. The orientation can be computed using the gradient of the first (non-droplet) material found in the cell (*By grad. of the 1st material*), or it can be given an element-centered vector variable (*vector*)

Since the order of the materials is significant in the Young's algorithm, it is important to be able to change the order of the materials. Order of the materials can be changed by right-clicking on the materials in the materials list and selecting Move Up, Move Down, Move to Top, or Move to Bottom.

Apply New Material(s)	Recreates the Material Part selected in the Main Parts List according to the selections in the material list.
Advanced	Will open the Feature Detail Editor and select the Material icon for more advance control of the Material Part.

Feature Detail Editor (Material Parts)

Double Clicking on the Material Parts Create/Update Icon (or Edit > Part Feature Detail Editors > Materials Parts...) opens the Feature Detail Editor (Material Parts), the Creation Attributes of which offers the same options for the Material Parts as the Quick Interaction Area Editor.

(See: How To Create Material Parts, and under Section 11.1, EnSight Gold Casefile Format, see EnSight Gold Material Files Format)

Troubleshooting Material Parts

Problem	Probable Causes	Solutions
No Type Domain Material Part created for specified material description(s)	Model part(s) not selected.	Select only model part(s).
	Model part(s) void of that material	Nothing wrong.
No Type Interface Material Part created for specified material description(s)	Model parts not selected.	Select only model part(s).
	Two or more materials not selected.	Select at least two (or more) materials.
	Selected materials are not adjacent across a 3D face or 2D edge.	Nothing wrong.
No "null_material" Material Part created for a specified "null_material" selection.	Model parts do not contain any null materials.	Nothing wrong.
There are no null materials, but selecting "null_material" produces a visible region.	Incorrect indexing in the material ids file.	Material ids file possibly has a negative index to an incorrect position into the mixed-material id file.
Increasing the Subdivide level does not increase the material fraction detail	Increasing the Subdivide level typically only increases the element resolution.	Typically nothing wrong.
Changing the Type and/or Level as well while simultaneously changing the material selections did not update the selected Material Part to the new material selections.	Material reselection is only updated via the Apply New Material(s) button.	Update the new materials first, then change the type.
		Delete the Material Part. Make new material(s) selection and Type and/or Subdivide specification. Then Create a new Material Part.
Orientation does not appear correct when using Young's algorithm.	Materials were not ordered correctly prior to creation.	Order materials in list so that first material gradient gives proper orientation.

7.20 Tensor Glyph Parts Create/Update

Tensor glyphs visualize the direction and tension/compression of the eigenvectors at discrete points (at nodes or at element centers) for a given tensor variable.

Tensor glyph Parts are dependent Parts known only to the client. They cannot be used as a parent Part for other Part types and cannot be used in queries. As dependent Parts, they are updated anytime the parent Part and/or the creation tensor variable changes (unless the general attribute Active flag is off).

Tensor glyphs can be filtered to show just the tensile or compressive eigenvectors. Further, the visibility for each of the eigenvectors (Major, Middle, and Minor) can be controlled.

Tensor glyphs will appear for each of the nodes/elements for the Parent part's visual Representation. Thus, for a border Representation of a Part, only the border nodes/elements will be candidates for a tensor glyph.

The tensile and compressive eigenvectors can be visualized by modifying the tensile/compressive component's line width and color.

Clicking once on the Tensor Glyph Create/Update Icon opens the Tensor Glyph Editor section of the Quick Interaction Area which is used to both create and update (make changes to) tensor glyph Parts.

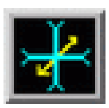


Figure 7-107
Tensor Glyph Parts Create/Update Icon

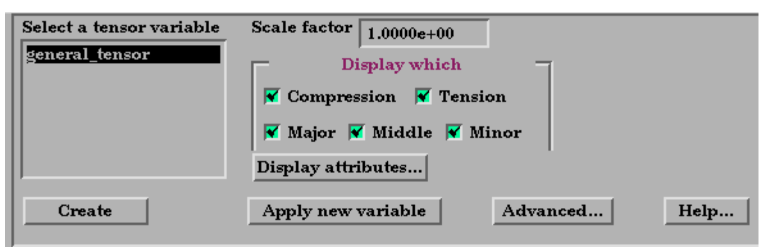


Figure 7-108
Quick Interaction Area - Tensor Glyph Parts Editor

Scale Factor The size of the tensor glyph.

Display Which Controls which eigenvectors will be displayed

Compression	Show the eigenvectors that are in compression
Tension	Show the eigenvectors that are in tension
Major	Show the major eigenvector
Middle	Show the middle eigenvector
Minor	Show the minor eigenvector

Display Attributes Opens the Tensor Display Attributes dialog.

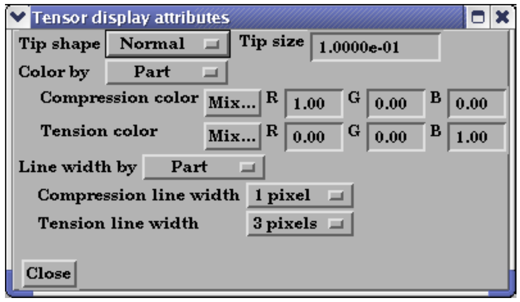


Figure 7-109
Tensor Display Attributes Dialog

Tip Shape Opens a pop-up menu to select the tip shape

None	Displays eigenvectors as lines with no tips.
Normal	Displays “classical” tips.
Triangles	Displays triangle tips.

Tip Size Controls the size of the tips.

Color By The tensor glyphs can be colored according to the part color, or have a separate color for compression and tension.

Compression Color	Specify the compressive color
Tension Color	Specify the tensile color

Line Width By The tensor glyphs can use the part line width, or have a separate line width for compression and tension.

Compression Line Width	Specify the compressive line width
Tension Line Width	Specify the tensile line width

Apply New Variable Changes the tensor Variable used to create the Tensor Glyphs to that currently selected in the “Select a Tensor Variable” list.

Advanced Will open the Feature Detail Editor and select the Vector Arrow icon for more advance control of the Vector Arrow Part.

Feature Detail Editor (Tensor Glyph) Double clicking on the Tensor Glyph Create/Update Icon opens the Feature Detail Editor for Tensor Glyphs, the Creation Attributes Section of which provides access to the same functions available in the Quick Interaction Area. For a detailed discussion of the remaining Feature Detail Editor turn-down sections (which are the same for all Part types)

(see Section 3.3, Part Editing and How to Create Tensor Glyphs)

Troubleshooting Tensor Glyphs

Problem	Probable Causes	Solutions
No tensor glyphs created	No real eigenvectors exist.	None
	Scale Factor too small.	Increase Scale Factor.
	Parent parts have non-visual attributes.	Re-specify parent parts or modify parent part's Element Representation.
	Parent parts do not contain selected tensor variable.	Re-specify parent parts.
Too many glyphs	Parent parts have too many points at which tensor glyphs are to be displayed.	Consider using a grid clip as the parent part.

7.21 Developed Surface Create/Update

A Developed Surface is generated by treating any 2D Part (or parent Part) as a surface of revolution, and mapping specific curvilinear coordinates of the revolved surface into a planar representation.

A Developed Surface derives its name from the implied process that defines a developable surface. A surface is considered “developable” if it can be unrolled onto a plane without distortion. Although every 2D Part in EnSight is not by definition a developable surface, each 2D Part can nevertheless be developed into a planar surface which is distorted according to the type of developed projection specified. For example, a Cylinder Clip Part is by definition a developable surface, because it can be developed into planar surface without distortion. Whereas, a Sphere Clip Part is not a developable surface, because it can not be developed into a planar surface without distortion.

Parent Parts

Only 2D Parts are developed. Also, only one Part is developed at a time. While all 2D Parts qualify as candidate parent Parts, only 2D Parts of revolution are developed coherently. The current developed surface algorithm treats all parent Parts as surfaces of revolution that are developed according to a local origin and axis of revolution. These attributes are either inherited from the parent Part, or must be specified according to the parent Part.

A developed surface permanently inherits the local origin and axis of revolution information from any parent Part created via the cylinder, cone, sphere, or revolution Clip tools. Whereas, surfaces developed from non-Clip Parts require this information to be specified via the Orig. and Axis fields in the Attributes (Developed Surfaces) dialog. The latter case is the only time the values in these fields are used. Although default values are provided, it is up to you to make sure that valid values are specified. In the former case, the Orig and Axis fields only provide convenient feedback of the selected Clip Part. Note that developed surfaces resulting from parent Parts of revolution created via the general quadric Clip tool do not inherit the local origin and axis of revolution attributes from the General Quadric Clip parent; rather, these attributes must be specified.

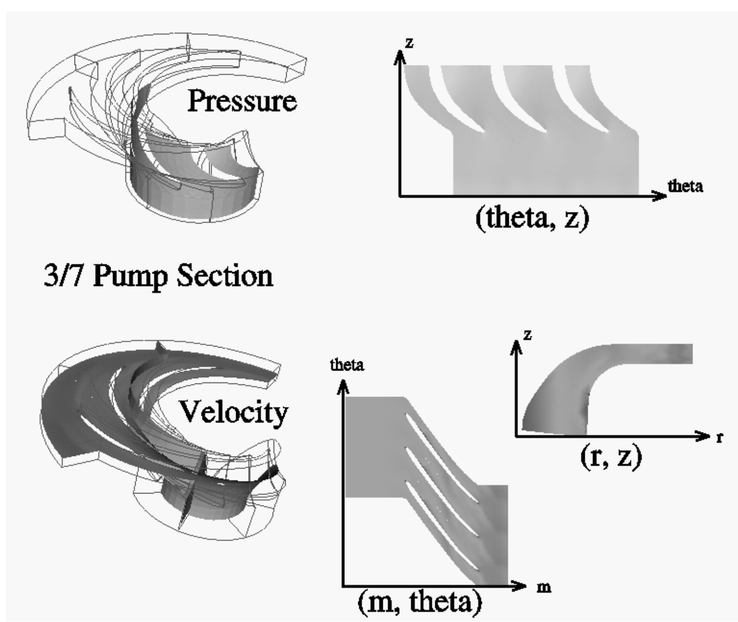


Figure 7-110 Developed Surface Examples

Developed Projections A parent Part is developed by specifying one of three curvilinear mappings called *developed projections*; namely, an (r,z), (θ,z), or (m,θ) projection. The curvilinear coordinates r, θ, z, and m stand for the respective radius, θ, z, and meridian (or longitude) directional components which are defined relative to the local origin and axis of revolution of the parent Part. The meridian component is defined as $m = SQRT(r^2 + z^2)$.

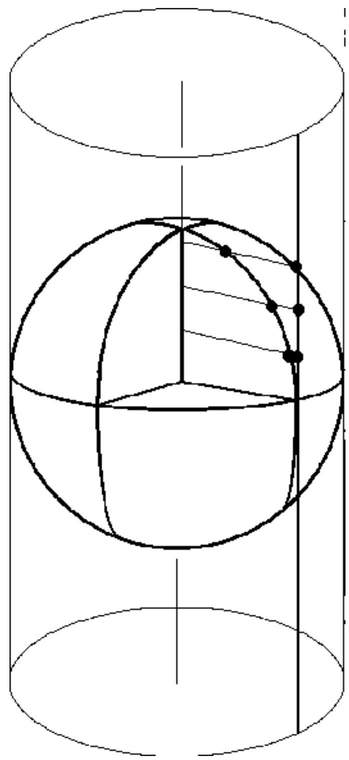


Figure 7-111
Developed Equiareal Projection

Essentially, each topological projection first surrounds the parent Part of revolution with a virtual cylinder of constant radius. The curvilinear coordinates of the parent Part are then projected along the normals of (and thus onto) the virtual cylinder. Finally, the virtual cylinder is slit along a straight line, or generator, and unwrapped into a plane. This process yields an *equiareal,* or *area preserving*, mapping which means that the area of any enclosed curve on the surface of the parent Part is equal to the area enclosed by the image of the enclosed curve on the developed plane. Although equiareal mappings provide reduced shape distortion, they do suffer from angular distortions of local scale.

Vector fields of the parent Part (for all three developed projections) are developed such that a vector’s angle to its surface normal is preserved. For example, a vector normal to the parent surface remains normal when developed onto the planar surface.

Seam Line A surface of revolution is developed about its axis, starting at its “seam” line (or zero meridian) where the surface is to be slit. Surface points along the seam are duplicated on both ends of the developed Part. The seam line is specified via a vector that is perpendicular to and originates from the axis of revolution, and which points toward the seam which is located on the surface at a constant value. This vector can be specified either manually or interactively. Interactive seam line display and manipulation is provided via a slider in the Attributes (Developed Surfaces) dialog.

Clicking once on the Developed Surface Create/Update Icon opens the Developed Surface Editor in the Quick Interaction Area which is used to both create and update (make changes to) developed surface Parts.

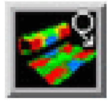


Figure 7-112
Developed Surface Create/Update Icon

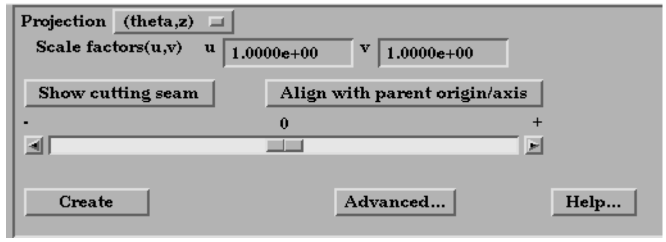


Figure 7-113
Quick Interaction Area - Developed Surface Editor

Projection

Opens a pop-up dialog for the specification of which type of (u,v) projection, or mapping, you wish to use for developing a surface of revolution; where u,v denotes curvilinear components of the parent Part that are mapped into the xy-plane of reference Frame 0. The options are:

(m,r)	denotes the meridian and radial components of the revolved surface
(m,theta)	denotes the meridian and theta components of the revolved surface.
(r,z)	denotes the radial and z-directional components of the revolved surface
(theta,z)	denotes the theta and z-directional components of the revolved surface.
(m,theta)	denotes the meridian and theta components of the revolved surface.

The meridian component is the curvilinear component along a revolved surface that runs in the direction of its axis of revolution (e.g. the meridonal and z-directional components along a right cylinder are coincident, and for a sphere the meridian is the longitude).

Scale Factors (u,v)

These fields specify the scaling factors which will be applied to the u and v projections.

Show Cutting Seam

Click this button to display the current seam line location about the circumference of the revolved surface. The seam line is manipulated interactively via the Slider Bar.

Align with Parent Origin/Axis

Retrieves the Origin and Axis information from the Parent Part. Must be done if Parent Part is a quadric clip.

Feature Detail Editor (Developed Surfaces

Double clicking on the Developed Surface Create/Update Icon opens the Feature Detail Editor for Developed Surfaces, the Creation Attributes Section of which provides access to all of the functions available in the Quick Interaction Area plus several more:

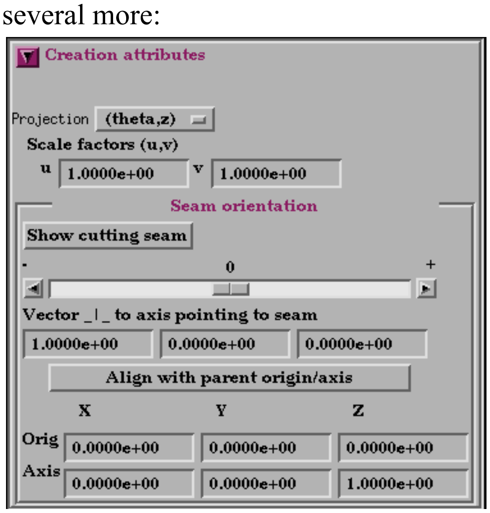


Figure 7-114
Feature Detail Editor (Developed Surfaces)

Vector _|_ To Axis Pointing To Seam	These fields allow you to precisely specify the position of the Cutting Seam Line by specifying the direction of the vector perpendicular to the axis of revolution which points in the direction of the seam line.
Orig X Y Z	These fields specify a point on the axis of revolution.
Axis X Y Z	These fields specify a vector, which when used with the Axis Origin defines the axis of revolution.

The Feature Detail Editor also allows you to make changes in batch; that is, to make several changes to the menus and fields which do not effect the Graphics Window until you click in the Apply Changes button. It is sometimes quicker (with respect to CPU time) to make several changes at once rather than one at a time as in the Quick Interaction Area.

(see Section 3.3, Part Editing for a detailed discussion of the remaining Feature Detail Editor turn-down sections which are the same for all Parts),

(see How To Create Developed (Unrolled) Surfaces)

Troubleshooting Developed Surfaces

Problem	Probable Causes	Solutions
Error message is encountered while creating a Developed Surface Part.	Parent Part is invalid.	Only 2D Parts can be developed.
Developed Surface is created, but is either not visible, Partially visible, or obstructed by other Parts which may be other developed Parts	Since all Developed Surfaces are projected about the origin on the xy-plane of the reference frame of the parent Part, they may map outside the viewport, intersect other Parts, or pile up on each other.	Set the Developed Surface to be viewed in its own viewport and initialize the viewport. Use different u/v scaling. Assign the developed Part to its own local reference frame and transform it accordingly.
Developed Surface Part is a line.	Wrong Projection type was specified.	Select a different Projection.
Developed Surface Part does not update to new Orig and/or Axis values.	The Orig and Axis values can not be specified if the Parent Part is created from either a cylinder, sphere, cone, or revolution quadric clip. These values can only be specified if the 2D parent Part is not a quadric clipped surface.	Since values entered for this condition are not used, click the Get Parent Part Defaults button to update the fields based on the selected parent Part in the Parts & Frames list.

7.22 Point Parts Create/Update

Point parts are composed only of nodes. They can be created by reading an external file containing the xyz coordinates of the nodes, and/or by placing the cursor tool at desired locations and adding nodes. This feature can be used to essentially place probes in the model at particular locations. It can also be used to create parts that can be meshed with the 2D or 3D meshing capability within EnSight.

Clicking once on the Point Part Create/Update Icon opens the Point Part Editor section of the Quick Interaction Area which is used to both create and update (make changes to) Point Parts.

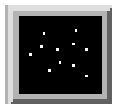


Figure 7-115
Point Parts Create/Update Icon

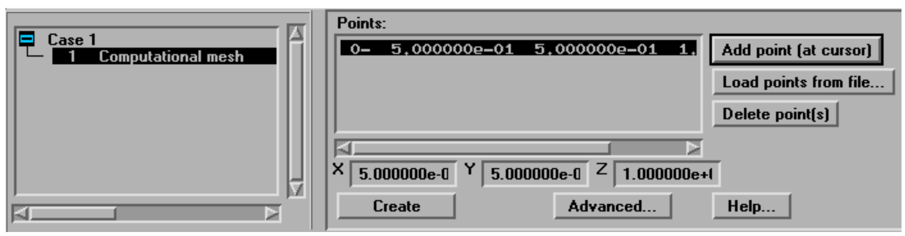


Figure 7-116
Quick Interaction Area - Point Parts Editor

Points: The list of points that will be used to create a point part, or that are in an already created point part. When points are added, they will show up in this list. Use this list to select points for modification or deletion.

Add point (at cursor) Adds a point (at the xyz location of the cursor) to the list of points.

Load points from file... Brings up the file selection dialog, so a file which contains the xyz locations of points, can be selected. Points in the file will be added to the Points list. (see Section 11.17, Point Part File Format).

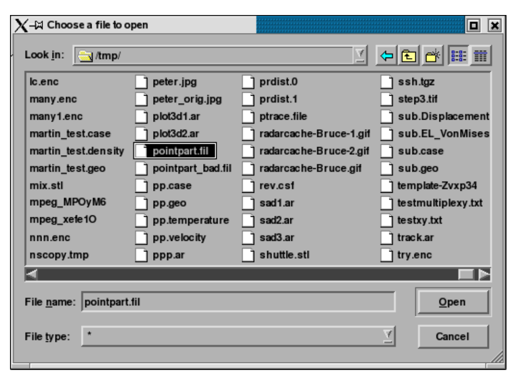


Figure 7-117
File Section Dialog - Point Parts

Delete point(s) Deletes any points selected in the Points list.

XYZ Shows the xyz coordinates of a selected point. Allows for editing of the values.

Create Creates a Point part, composed of the points in the Points list. The part will show up in the Parts list.

Advanced... Opens the Feature Detail Editor for Point Parts, which contains the same basic capabilities described in the Quick Interaction Area above.

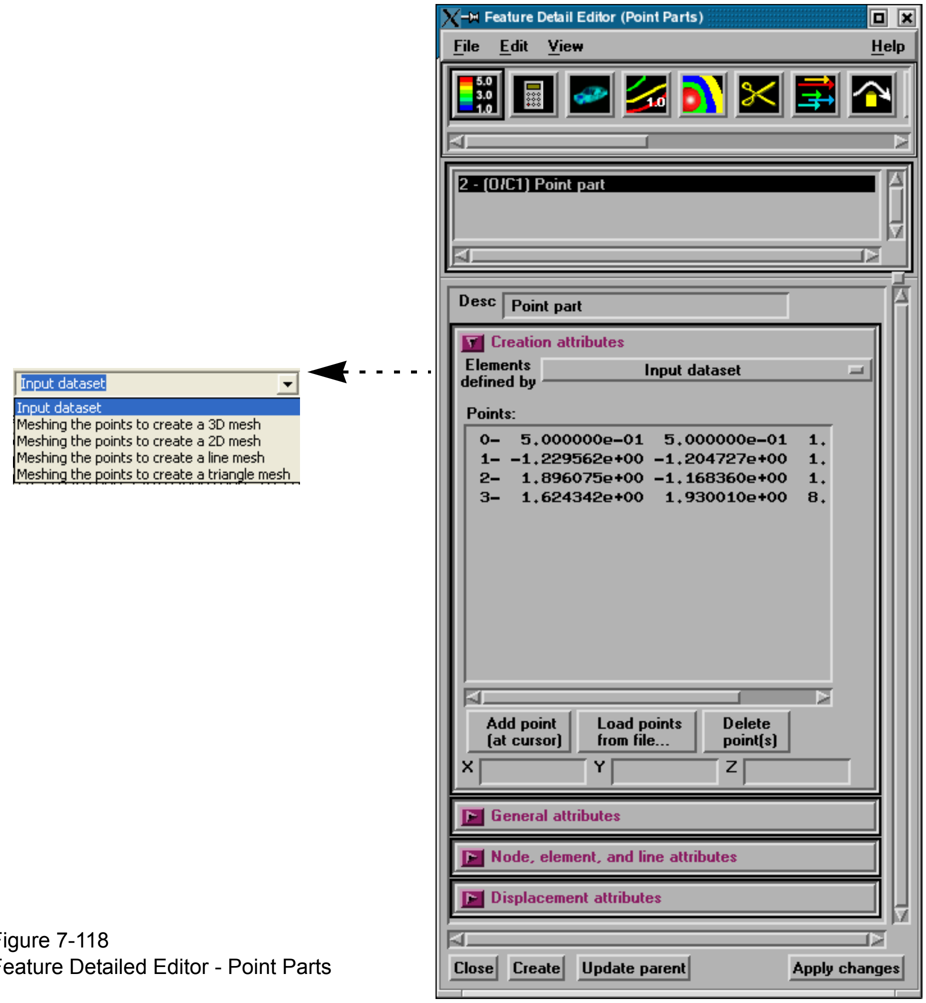


Figure 7-118
Feature Detailed Editor - Point Parts

Elements Defined by Points can be used to create 3D, 2D, or 1D elements.

- Input dataset — Simply use the input dataset as 0D point elements.
- Meshing points — Several Meshing point options follow:
  - 3D Mesh — Create a 3D volume mesh of tetrahedral elements.
  - 2D Mesh — Create a 2D surface mesh of triangle elements.
  - Line Mesh — Connect the points together in the order they were created into 1D line elements.
  - Triangle Mesh — Connect the points together in the order they were created in sets of 3 into 2D triangle elements. Effectively this tesselates the points.

Help... Opens context sensitive help for Point Parts.

See also How To Use Point Parts

7.23 Extrusion Parts Create/Update

Extrusion parts are created by "extruding" a part in a directional or rotational manner to produce a part of next higher order. For example, a 2D axi-symmetric surface can be extruded rotationally about the proper axis to produce a 3D representation of the complete model. As another example, a 1D line can be extruded in a direction to produce a 2D plane.

Clicking once on the Extrusion Create/Update Icon opens the Extrusion Editor section of the Quick Interaction Area which is used to both create and update (make changes to) extrusion Parts.

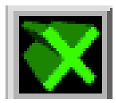


Figure 7-119
Extrusion Parts Create/Update Icon

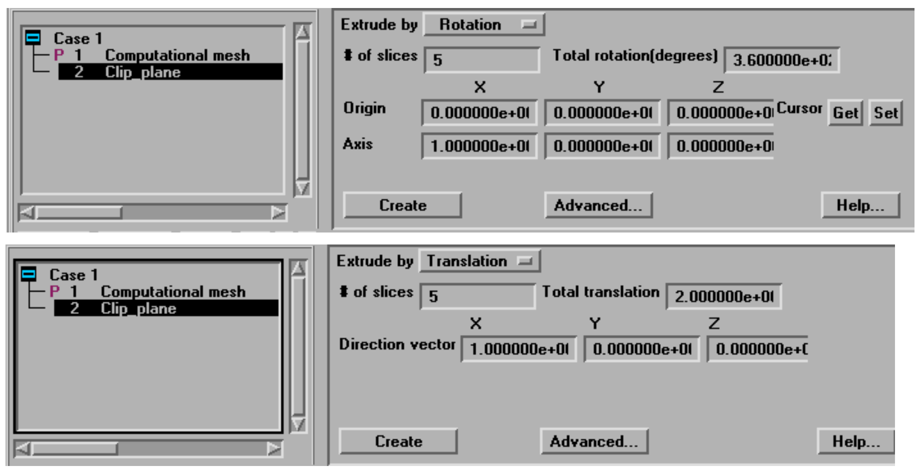


Figure 7-120
Quick Interaction Area - Extrusion Parts Editor

Extrude by	Controls the type of extrusion to use.
	Rotation — To extrude the selected parts by revolving about an axis. This is what you would choose for an axi-symmetric part.
	Translation — To extrude the selected parts by translating in a given direction.
# of slices	Sets the number of elements that will be created in the "extrusion" direction. For rotation, it would be the number of "slices" around the "pie". For translation, it would be the number of elements along the extrusion vector direction.
Total rotation (degrees)	For rotation, sets the total number of degrees to rotate. It must be between -360 and +360.
Origin	For rotation, sets x,y,z values for the origin of the axis of rotation.
Axis	For rotation, sets the direction vector components for the axis of rotation.
Cursor Get/Set	Can be used to get the origin values from the current cursor location, or to set the location of the cursor to be at the origin location.
Total translation	For translation, sets the total distance of extrusion travel.

Direction vector	For translation, sets the direction vector components for the directional extrusion.
Advanced	Will open the Feature Detail Editor and select the Extrusion icon for more advance control of the Extrusion Part.
Feature Detail Editor (Extrusion	Double clicking on the Extrusion Create/Update Icon opens the Feature Detail Editor for Extrusions, the Creation Attributes Section of which provides access to the same functions available in the Quick Interaction Area. For a detailed discussion of the remaining Feature Detail Editor turn-down sections (which are the same for all Part types)

(see Section 3.3, Part Editing and How to Extrude Parts)

Troubleshooting Extrusions

Problem	Probable Causes	Solutions
No extrusions created	Parent Part is not a valid server-side part.	Don’t try to extrude client-side parts (particle traces, contours, etc.)

8 Modes

This chapter describes the five different Modes through which you can work in the Graphics Window as well as the Quick Desktop Buttons that are located above the Graphics Window. The “active” Mode determines both the configuration and what functions are available through the Mode Icon Bar. The Quick Desktop Buttons are always available regardless of the mode and are useful as quick shortcuts for often-used tasks.

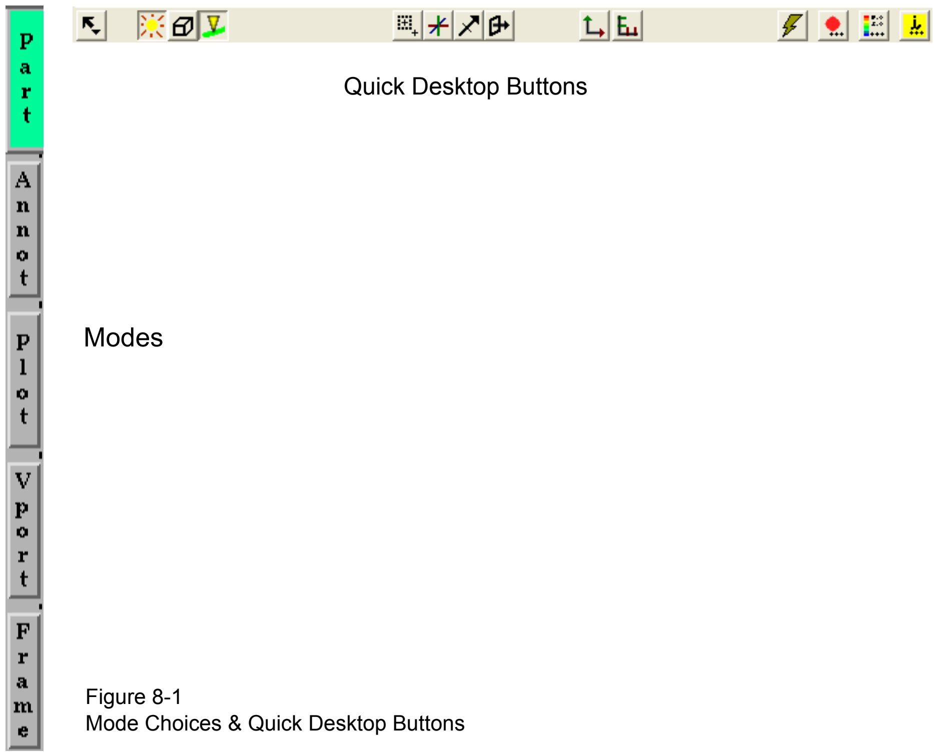


Figure 8-1
Mode Choices & Quick Desktop Buttons

Section 8.1, Part Mode describes the layout of the Mode Icon Bar and the functions available when **Part** is the active Mode.

Section 8.2, Annot Mode describes the layout of the Mode Icon Bar and the functions available when **Annot** is the active Mode.

Section 8.3, Plot Mode describes the layout of the Mode Icon Bar and the functions available when **Plot** is the active Mode.

Section 8.4, VPort Mode describes the layout of the Mode Icon Bar and the functions available when **VPort** is the active Mode.

Section 8.5, Frame Mode describes the layout of the Mode Icon Bar and the functions available when **Frame** is the active Mode. *By default, this mode is not available unless it has been enabled under Edit > Preferences... General User Interface - Frame Mode Allowed*

Section 8.6, Quick Desktop Buttons describes the layout of the Quick Desktop Buttons which are always available.

8.1 Part Mode

Part Mode is used to adjust a number of attributes for individual Parts and to specify the desired type of Pick operation.

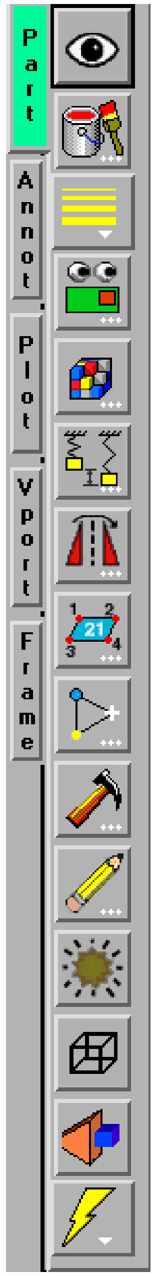


Figure 8-2
Mode Selection Area - Part Selected

For a complete discussion about Parts:

(see Chapter 3, Parts)

Part Visibility Toggle Icon

Determines the global (in all viewports and in all Modes) visibility of the selected Part(s). Note you can just right click on the part and choose ‘Hide’ to make it invisible.

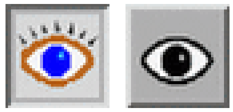


Figure 8-3
Part Mode - Part Visibility Toggle ON - OFF Icons

Part Color, Lighting, Transparency Icon

Clicking once on the Part Color, Lighting, Transparency Icon opens a dialog which allows you to assign color, lighting characteristics, and transparency levels to the individual Part(s) which has(have) been selected in the Parts List. If no Parts are selected, modifications will affect the default Part color and all Parts subsequently loaded or created will be assigned the new default color. Note, you can just right click on a part and choose how to color it..

Figure 8-4
Part Color, Lighting, Transparency Icon

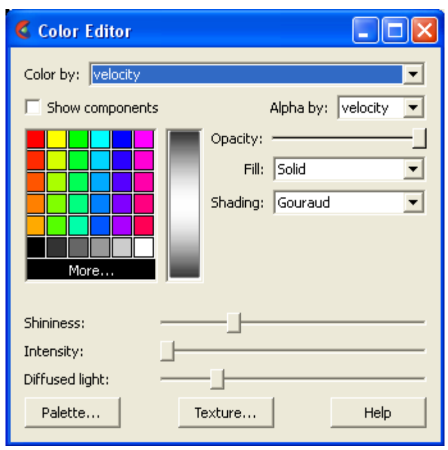


Figure 8-5
Part Mode - Part Color, Lighting & Transparency dialog

Shading

Opens a pop-up menu for selection of appearance of Part surface when Shaded Surface is on. Normally the mode is set to Gouraud, meaning that the color and shading will interpolate across the polygon in a linear scheme. You can also set the shading type to Flat, meaning that each polygon will get one color and shade, or Smooth which means that the surface normals will be averaged to the neighboring elements producing a “smooth” surface appearance. Not valid for all Part types. Options are:

Flat	Color and shading same for entire element
Gouraud	Color and shading varies linearly across element
Smooth	Normals averaged with neighboring elements to simulate smooth surfaces

Shininess

Shininess factor. You can think of the shininess factor in terms of how smooth the surface is. The larger the shininess factor, the smoother the object. A value of 0 corresponds to a dull finish and larger values correspond to a more shiny finish. To change, use the slider.

Intensity

Highlight intensity (the amount of white light contained in the color of the Part which is reflected back to the observer). Highlighting gives the Part a more realistic appearance and reveals the shine of the surface. To change, use the slider. Will have no effect if Highlight

Shininess parameter is zero.

Diffused Light
: Diffusion (minimum brightness or amount of light that a Part reflects). (Some applications refer to this as *ambient* light.) The Part will reflect no light if value is 0.0. If value is 1.0, no lighting effects will be imposed and the Part will reflect all light and be shown at full color intensity at every point. To change, use the slider.

Opacity
: The opaqueness of the selected Part(s). A value of 1.0 indicates that the Part is fully opaque, while a value of 0.0 indicates that it is fully transparent. Setting this attribute to a value other than 1.0 will adversely affect the graphics performance. Opacity is disabled for line parts.

Fill Pattern
: Opens a pop-up menu for selection of a fill pattern which can provide pseudo-transparency for shaded surfaces. Default is Solid which uses no pattern (produces a solid surface), while Fill patterns 1 through 3 produce a EnSight defined fill pattern.

Color By pulldown
: Allows you to choose whether to color the selected Part(s) by a Constant Color or by the Variable selected in the pulldown list.

Constant Color
: The selected Part(s) may be assigned a constant color by selecting it from the pre-defined matrix of color cells.

More...
: Alternatively, you can click on the More... area and the Color Selector dialog will open.

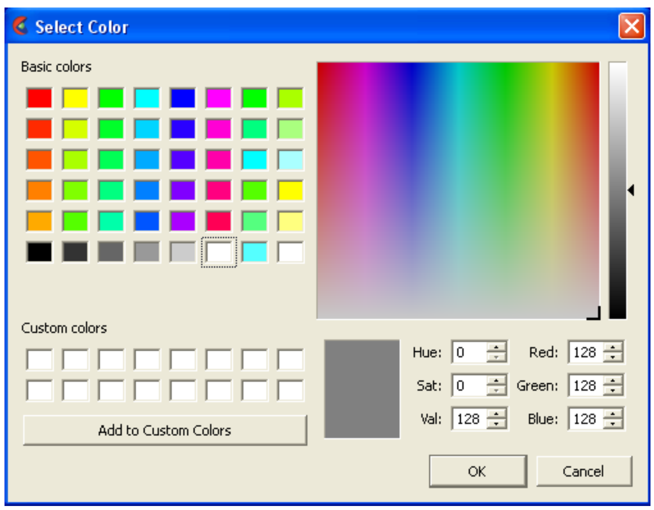


Figure 8-6
Color Selector dialog

You can choose a color by entering RGB or HSV values directly, picking a color from the matrix, or custom color lists, or by utilizing the color square and slider. Regardless of which method you use to define a color, it will not be applied to the selected Part(s) until you click the OK button.

Variables
: Alternatively, the Part(s) may be colored by a variable selected in the pulldown list. The color palette for each Variable associates a color with each value of the variable and these colors are used to color the selected Part(s).

If coloring by a nodal variable, the default coloring will be continuously varying - even within a given element. If you are coloring by a per-element variable, the coloring will not vary within a given element. If you desire to see per-element variables in a continuously varying manner, you can toggle on “Use continuous palette for per element vars” under Edit->Preferences... Color Palettes.

Show Components
: If you are coloring by a vector variable, this toggle will expand the list to include the vector components - thus you can choose to color using the magnitude or a component of the vector.

Palette...
: Clicking on Palette... will open the Palette Editor dialog.

See How To Edit Color Palettes

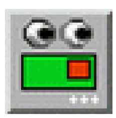
Texture... Clicking on Texture... will open the Textures dialog.
See How To Map Textures

Part Line Width Pulldown Icon Opens a pulldown menu for the specification of the desired display width for Part lines. Performs the same function as the Line Representation Width field in the Node, Element, and Line Attributes section of the Feature Detail Editor (Model).

Figure 8-7
Part Mode - Part Line Width Pulldown Icon

Access: Part Mode : Part Line Width Pull-down Icon

Part Visibility in Viewport Icon Opens the "Part Visible in Which Viewport?" dialog. If the global visibility of a Part is on, this dialog can be used to selectively turn on/off visibility of the selected Part(s) in different viewports simply by clicking on a viewport's border symbol within the dialog's small window. The selected Part(s) will be visible in the green viewports invisible in the black viewports.



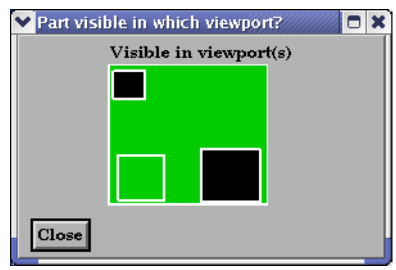


Figure 8-8
Part Mode - Part Visibility in Viewport Icon and Part Visible in Which Viewport? dialog

Access: Part Mode: Part Visibility in Viewport Icon

Element Visual Representation Pulldown Icon Opens a dialog for the specification of the desired representation for elements of the selected Part(s). Performs the same function as the Element Representation Visual Rep. pulldown menu in the Node, Element, and Line Attributes section of the Feature Detail Editor (Model).

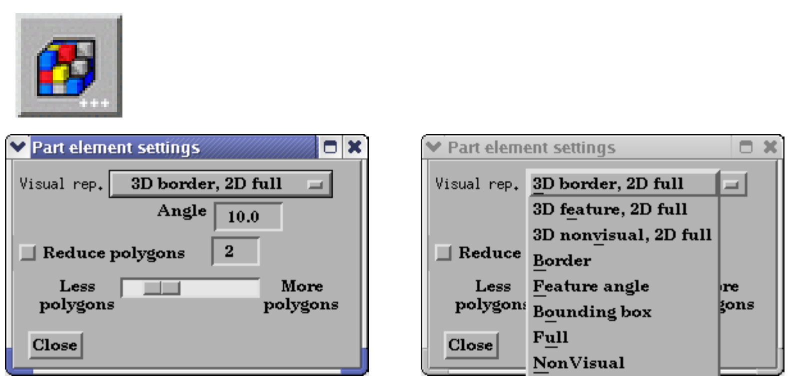
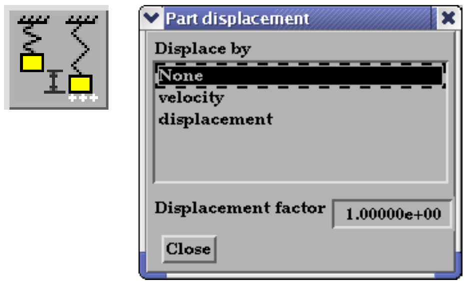


Figure 8-9
Part Mode - Element Representation Icon

(see Element Representation in Section 3.3, Part Editing)
Access: Part Mode : Element Representation Pull-down Icon

Part Displacement — *Opens the Part Displacement dialog which allows you to choose the vector variable. The model geometry is displaced by this variable vector value.*



Displacement factor — The vector variable can be scaled by this factor.

Each node of a Part is displaced by a distance and direction corresponding to the value of a vector variable at the node. The new coordinate is equal to the old coordinate plus the vector times the specified Factor, or:

$C_{new} = C_{orig} + \text{Factor} * \text{Vector},$

where C_{new} is the new coordinate location, C_{orig} is the coordinate location as defined in the data files, Factor is a scale factor, and Vector is the displacement vector.

You can greatly exaggerate the displacement vector by specifying a large Factor value. Though you can use any vector variable for displacements, it certainly makes the most sense to use a variable calculated for this purpose. Note that the variable value represents the *displacement* from the original location, not the *coordinates* of the new location.

Displacement Factor — This field specifies a scale factor for the displacement vector. New coordinates are calculated as: $C_{new} = C_{orig} + \text{Factor*Vector}$, where C_{new} is the new coordinate location, C_{orig} is the original coordinate location as defined in the data file, Factor is a scale factor, and Vector is the displacement vector. Note that a value of 1.0 will give you "true" displacements.

Visual Symmetry Icon — Opens the Part Visual Symmetry dialog which allows you to control the display of mirror images of the selected Part(s) in each of the seven other quadrants of the Part's local frame or the rotationally symmetric instances of the selected parts. This performs the same function as the Visual Symmetry menu in the General Attributes section of the Feature Detail Editor (Model).

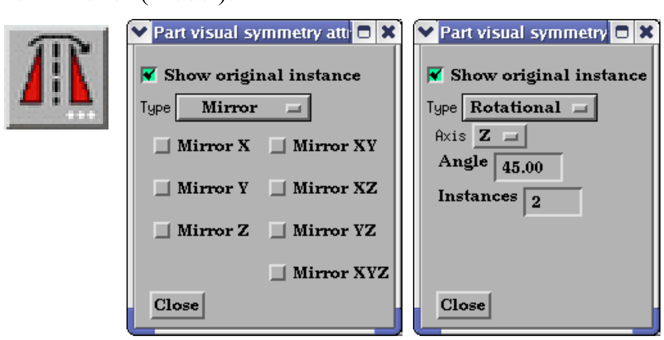


Figure 8-10
Part Mode - Visual Symmetry Icon

Symmetry enables you to reduce the size of your analysis problem while still visualizing the "whole thing." Symmetry affects only the displayed image, not the data, so you cannot query the image or use the image as a parent Part. However, you can create the same effect by creating dependent Parts with the same symmetry attributes as the parent Part.

You can mirror the Part to more than one quadrant. If the Part occupies more than one quadrant, each portion of the Part mirrors independently. Symmetry works as if the local frame is Rectangular, even if it is cylindrical or spherical. The images are displayed with the same attributes as the Part. For each toggle, the Part is displayed as follows. The default for all toggle buttons is OFF, except for the original representation - which is ON.

Mirror X	quadrant on the other side of the YZ plane.
Mirror Y	quadrant on the other side of the XZ plane.
Mirror Z	quadrant on the other side of the XY plane.
Mirror XY	diagonally opposite quadrant on the same side of the XY plane.
Mirror XZ	diagonally opposite quadrant on the same side of the XZ plane.
Mirror YZ	diagonally opposite quadrant on the same side of the YZ plane.
Mirror XYZ	quadrant diagonally opposite through the origin.
Show Original Instance	the original part instance

Rotational visual symmetry allows for the display of a complete (or portion of a) “pie” from one “slice” or instance. You control this option with:

Axis	X	rotates about the X axis
	Y	rotates about the Y axis
	Z	rotates about the Z axis
Angle		specifies the angle (in degrees) to rotate each instance from the previous
Instances		specifies the number of rotational instances.
Show Original Instance		show the original instance or not

Access: Part Mode : Visual Symmetry Icon

Element -Node Icon

Opens the Node/Element labelling attributes dialog. Toggles on/off the visibility of the element and/or node labels (assuming the result file contains them) for the selected Part(s). The Global Element Label Toggle (Main Menu>View>Label Visibility) must be on in order to see any element labels. Likewise, the Global Node Label Toggle (Main Menu>View>Label Visibility) must be on in order to see any element labels.

Figure 8-11
Part Mode - Element/Node Label Toggle Icon

Access: Part Mode : Element/Node Label Toggle Icon

.

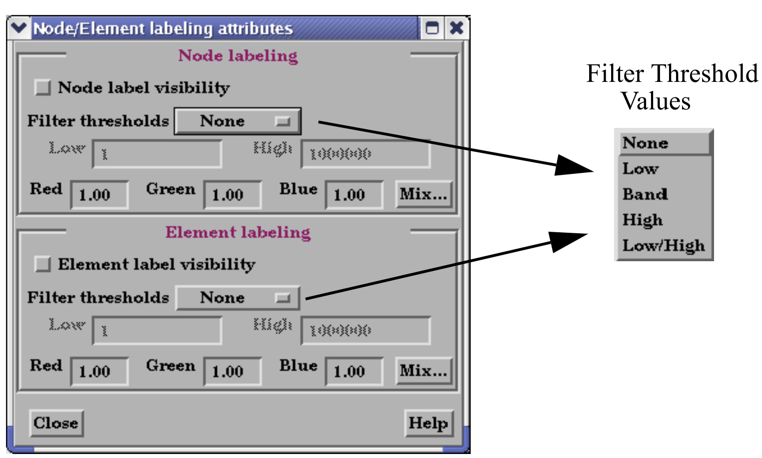


Figure 8-12
Part Mode - Element/Node Label Toggle Dialog

Access: Part Mode : Node Label Toggle Icon

Element/Node Label Visibility

Toggles on/off the visibility of the element or node labels (assuming the result file contains them) for the selected Part(s). Performs the same function as the Label Visibility Node toggle in the Node, Element, and Line Attributes section of the Feature Detail Editor (Model). Default is OFF.

Filter Thresholds

A pulldown menu containing the following
Low - All element/node ids below the value in the low field are invisible
Band - All element/node ids between the values in the low and high fields are invisible.
High - All element/node ids above the value in the high field are invisible.
Low/High - All element/node ids below the low and above the high field values are invisible.

Mix

Enter the element/node id label color, or click on the Mix Button and pick your color.

Node Representation Icon

Opens the Part Node Representation dialog. Performs the same function as the Node Representation area in the Node, Element, and Line Attributes section of the Feature Detail Editor (Model).

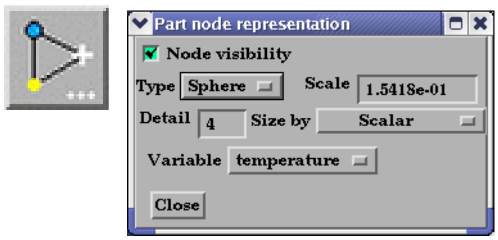


Figure 8-13
Part Mode - Node Representation Icon and Part Node Representation dialog

Node Visibility Toggle

Toggles-on/off display of Part's nodes whenever the Part is visible. Default is OFF.

Type

Opens a pop-up menu for the selection of symbol to use when displaying the Part's nodes or point elements. Default is Dot. Options are:
Dot to display nodes as one-pixel dots.
Cross to display nodes as three-dimensional crosses whose size you specify.
Sphere to display the nodes as spheres whose size and detail you specify. Use care when choosing this option for large numbers of nodes as this will require large amount of memory per node: 250 MB per million nodes (no display lists) up to 1 GB per million nodes (display lists on).

Scale

This field is used to specify scaling factor for size of node symbol. If Size By is Constant, this field will specify the size of the marker in model coordinates. If Size By is set to a variable, this field will be multiplied by the variable value. Not applicable when node-symbol Type is Dot.

Detail

This field is used to specify how round to draw the spheres when the node-symbol type is Sphere. Ranges from 2 to 10, with 10 being the most detailed (e.g., roundest spheres). Higher values take longer to draw, slowing performance. Default is 4.

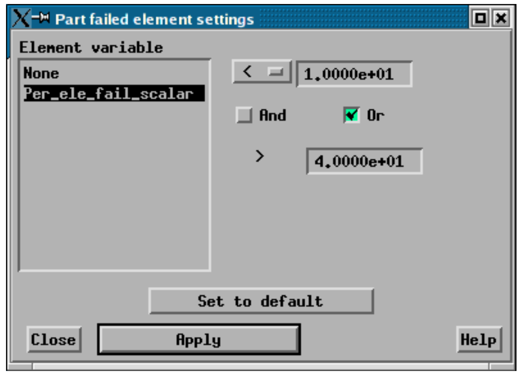
Size By

Opens a pop-up menu for the selection of variable-type to use to size each node-symbol. For options other than Constant, the node-symbol size will vary depending on the value of the selected variable at the node. Not applicable when node-symbol Type is Dot. Default is Constant. Options are:
Constant sizes node using the Scale factor value.
Scalar sizes node using a scalar variable.
Vector Mag sizes node using magnitude of a vector variable.
Vector X-Comp sizes node using magnitude of X-component of a vector variable.
Vector Y-Comp sizes node using magnitude of Y-component of a vector variable.
Vector Z-Comp sizes node using magnitude of Z-component of a vector variable.

Variable — Selection of variable to use to size the nodes. Activated variables of the appropriate Size By type are listed. Not applicable when node-symbol Type is Dot or Size By is Constant.

Access: Part Mode : Node Representation Icon

Part Failed Element Icon — Opens the Part Failed Element Settings dialog. Choose a *per-element scalar* variable. This does a deep element removal of the elements from the selected model part on the server based on logical operators on the variable.



Failure operator — Select a per-element scalar variable. Choose a pulldown threshold (<, >, = or !=).

And / Or — If you toggle on a logical operator then choose second threshold.

Set to default — To set the threshold values back to the default of 0.0.

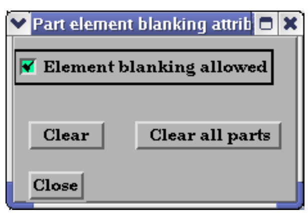
Part Element Blanking/Selection — Brings up the Part Element Blanking Attributes Dialog. Element blanking is the visual removal of elements on the graphics screen. The elements still remain on the server and are still used in calculations, they are just not visible in the graphics window. Note that blanking is done using element IDs as tags. If the element IDs change each timestep, this can result in different elements becoming invisible each timestep.

See How To Remove Failed Elements

Element blanking — Toggles on/off whether element blanking allowed

Clear — Clears blanked elements and restores them to visible for the selected Part(s)

Clear all parts — Clears blanked elements and restores them to visible for all Part(s)



See How To Do Element Blanking

Part Shaded Toggle Icon — Toggles on/off Shaded display of surfaces for the selected Part(s) assuming that Global Shaded has been toggled ON in the Main Menu > View > Shaded. Performs the same function as the Shaded Toggle in the General Attributes section of the Feature Detail Editor (Model). Default for all Parts is ON.

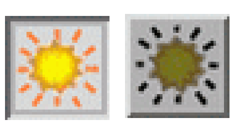


Figure 8-14
Part Mode - Part Shaded Toggle ON / OFF Icon

Access: Part Mode : Part Shaded Toggle Icon

Part Hidden Line Toggle Icon — Toggles on/off hidden line display of surfaces for the selected Part(s) assuming that the Global Hidden Line has been toggled ON in the Main Menu > View > Hidden Line. Performs the same function as the Hidden Line Toggle in the General Attributes section of the Feature Detail Editor (Model). Default for all Parts is ON.

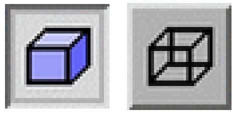


Figure 8-15
Part Mode - Part Hidden Line Toggle ON / OFF Icon

Access: Part Mode : Part Hidden Line Toggle Icon

Part Auxiliary Clipping Toggle Icon

Toggles on/off whether the selected Part(s) will be affected by the Auxiliary Clipping Plane feature. Performs the same function as the Aux Clip toggle in the General Attributes section of the Feature Detail Editor (Model). Default is ON. Auxiliary clipping is simply a visual clipping that occurs only on the client and does not affect the underlying model geometry, only it's view on the screen.
Note: The Global Auxiliary Clipping Toggle (in the View Mode Icon Bar) must be on in order for any Parts to be affected by the Aux Clip Plane.

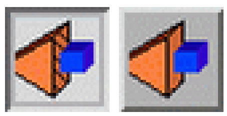


Figure 8-16
Part Mode - Part Auxiliary Clipping Toggle ON / OFF Icon

Access: Part Mode : Part Auxiliary Clipping Toggle Icon

Fast Display Toggle Icon

Opens a pull-down menu for the specification of the desired fast display representation in which a Part is displayed. The Part fast display representation corresponds to whether the view Fast Display Mode (located in the View Menu) is on. The Fast Display pull-down icon performs the same function as the Fast Display pop-up menu button in the General Attributes section of the Feature Detail Editor (of all parts).

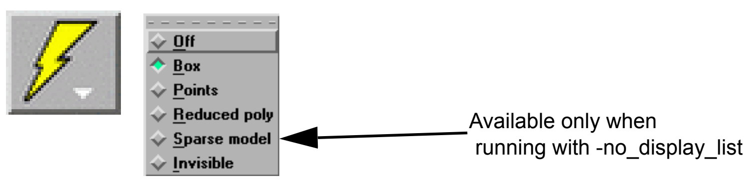


Figure 8-17
Part Mode - Detail Representation Pulldown Icon

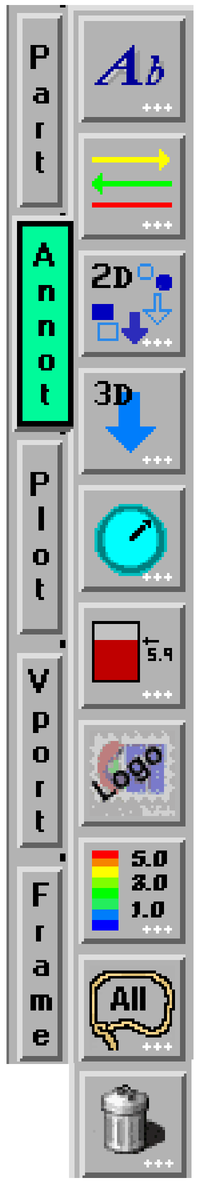
Box	causes selected Part(s) to be represented by a bounding box of the Cartesian extent of all Part elements (default)
Points	causes selected Part(s) to be represented by a point cloud
Reduced poly	causes selected Part(s) to be represented by reduced number of polygons
Sparse Model	decimates part elements by a factor determined in the Preferences (Edit>Preferences>Performance and enter in a factor from 1 (sparse) to 100 (full) in the Sparse Model Field. This is only available when running in immediate mode using the -no_display_list option at startup.
Invisible	causes the selected Part(s) to be invisible

(see General Attributes in Section 3.3, Part Editing, How To Set Global Viewing))

Access: Part Mode : Detail Representation Pull-down Icon

8.2 Annot Mode

Annot (Annotation) Mode is used to create and edit text strings, lines, 3D annotation arrows, and import logos into the Graphics Window., and to adjust their visibility, size, color. and position. It is also used to adjust the type, size, format, and position of legends.



Text Create / Edit Icon

Line Create / Edit Icon

2D Shapes Create / Edit Icon

3D Annotation Arrow Create / Edit Icon

Dial Create / Edit Icon

Guage Create / Edit Icon

Logo Import / Edit Icon

Legend Create / Edit Icon

Select All Icon

Delete Icon

Figure 8-18
Mode Selection Area - Annot Selected

When in Annot Mode, you are always modifying the objects selected in the Graphics Window. Selected Annotation objects are outlined in the “selection color”, while unselected objects are outlined in a white color. To select an object, click the mouse over it. To select multiple objects, hold the Control key down while clicking on the objects.

Annotation objects, other than text and 3D annotation arrows, are positioned in the main Graphics Window; they are not tied to specific viewports.

Text Create / Edit Icon Opens the Annotation Creation dialog with the Text icon highlighted.

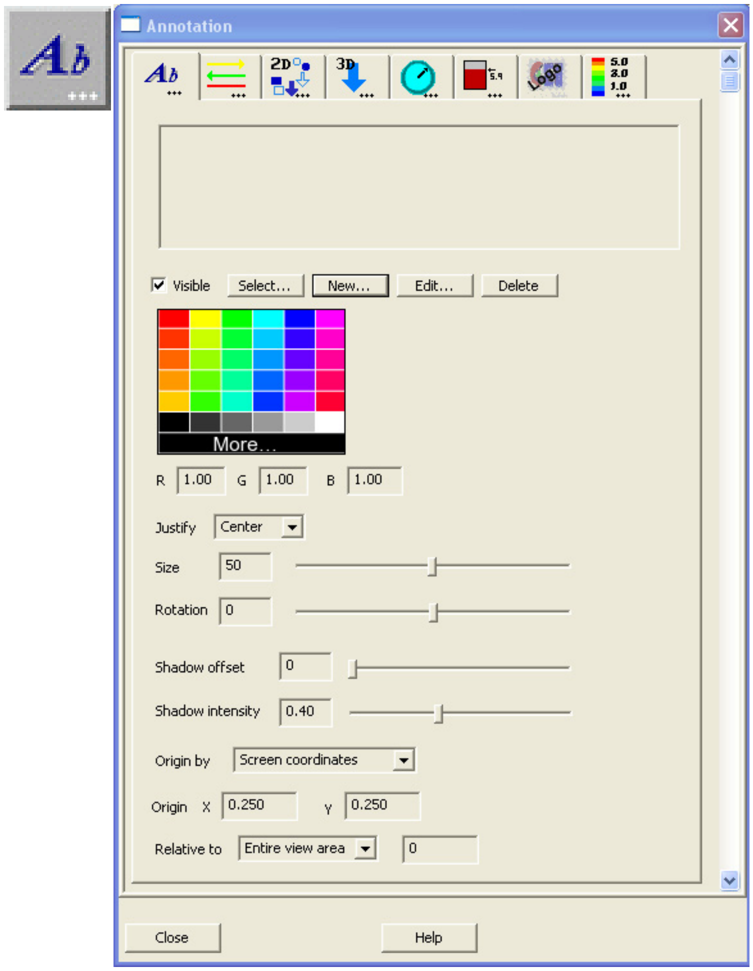


Figure 8-19
Annot Mode - Text Creation Icon and Text Annotation Creation dialog

Text strings may be created and inserted into the Graphics Window using the Text Annotation dialog. This dialog provides a list of text items.

Select — A pulldown that allows text item(s) to be selected. They can also be selected using the mouse in the text menu, or in the graphics window by clicking on them. Select options include any one of the following:

Select all	Select all text items.
Select visible	Select text whose visible toggle (located below the Select Button) is on.
Select invisible	Select text whose visible toggle (located below the Select Button) is off.
Deselect all	No text items are selected.

New / Edit

Clicking New or Edit button opens the Text Annotation Editing Dialog allowing the creation or modification of a text item. If you entered the dialog by way of the New... button, the text area will be blank and ready for your input. If you entered the dialog by way of the Edit... button, the text area will contain the selected text string and be ready for your editing. Note that pressing 'enter' inserts a new line into the annotation. To see the results of the edit, click on the 'Update text' button or check the 'Dynamic update' box. When dynamic updates are enabled, the annotations in the main EnSight window are redrawn with every keystroke. Note that this option causes EnSight to record every keystroke into the command stream as well. (Note that there is a maximum of 1024 characters per annotation that the GUI will display.)

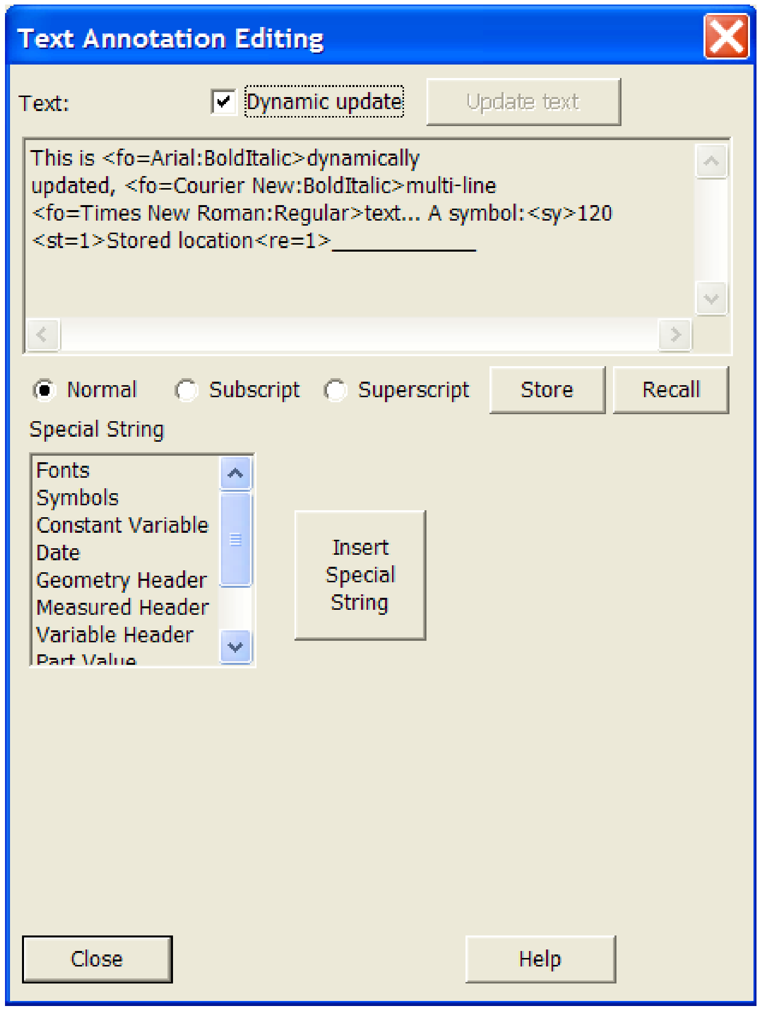


Figure 8-20
Annot Mode - Text Annotation Editing dialog

Normal, Superscript, Subscript

These toggles can be used to do superscripting and subscripting in your text string. They place codes in the string (<no> for normal, <up> for superscript, and <dn> for subscript) to delineate the desired mode.

Store, Recall

These buttons allow the user to insert the <st=1> and <re=1> codes into the text string for saving and recalling a text position on the screen.

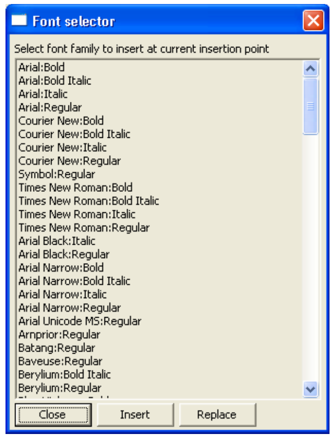
Special String

Lists a number of different special strings that can be inserted into the annotation currently being edited. These include information contained in results data or from other dialogs.

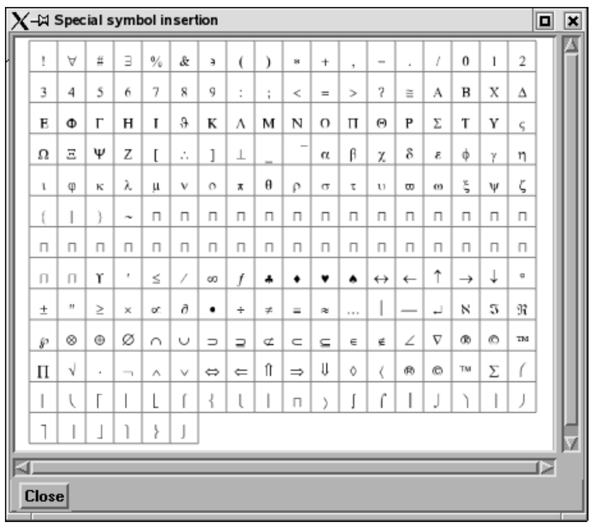
Insert Special String

Inserts the selected Special String into the text field at position of cursor.
Choices are:

Fonts brings up a TrueType font selection dialog allowing control over font selection inserts.



Symbols brings up a symbol selection dialog allowing selection of symbols to be inserted.



Constant Variable inserts the value of the constant variable selected and displays it in the selected format
Date inserts the Day of Week, Month, Date, Time, Year
Geometry Header inserts either the first or second header lines of the geometry file
Measured Header inserts the header line of the measured results file
Variable Header inserts the header line of the selected variable data file
Part Value inserts the value used to create the Isosurface or Clip Plane Part. Works for Isosurface Parts, or some (XYZ, IJK, RTZ, plane aligned with axis, or attached to a spline) Clip Plane Parts.
Part Description inserts the description of the Part selected in a Parts List which pops up within the Text Annotation Creation dialog
Version inserts the EnSight version number, including the (letter).
For example, 9.2.0 (b) would be 9.2.0(b)

See How To Manipulate Fonts for additional information on working with fonts.

Delete	Allows text item(s) to be deleted.
Visible	Toggles Text Visibility ON/OFF.
RGB	Color of Text.
Justify	Pulldown for Left, Right or Center justification.
Size	Field and slide for text size

Rotation/Degrees	Field for rotation in degrees (-360 to 360).
Shadow Offset	Specifies an offset for a drop shadow.
Shadow Intensity	The shadow intensity where 0 is black and 1 is white.
Origin by	Screen Coordinates indicates to locate the annotation in normalized screen space. If set to 3D coordinates the annotation is located in world coordinates and thus a X, Y, Z position must be specified.
Relative to	If Origin is by 3D coordinates this specifies which reference frame the X, Y, Z position is in reference to.

Access: Annot Mode : Text Creation Icon

Line Create /Edit Icon Opens the Annotation Creation dialog with the Line icon selected.

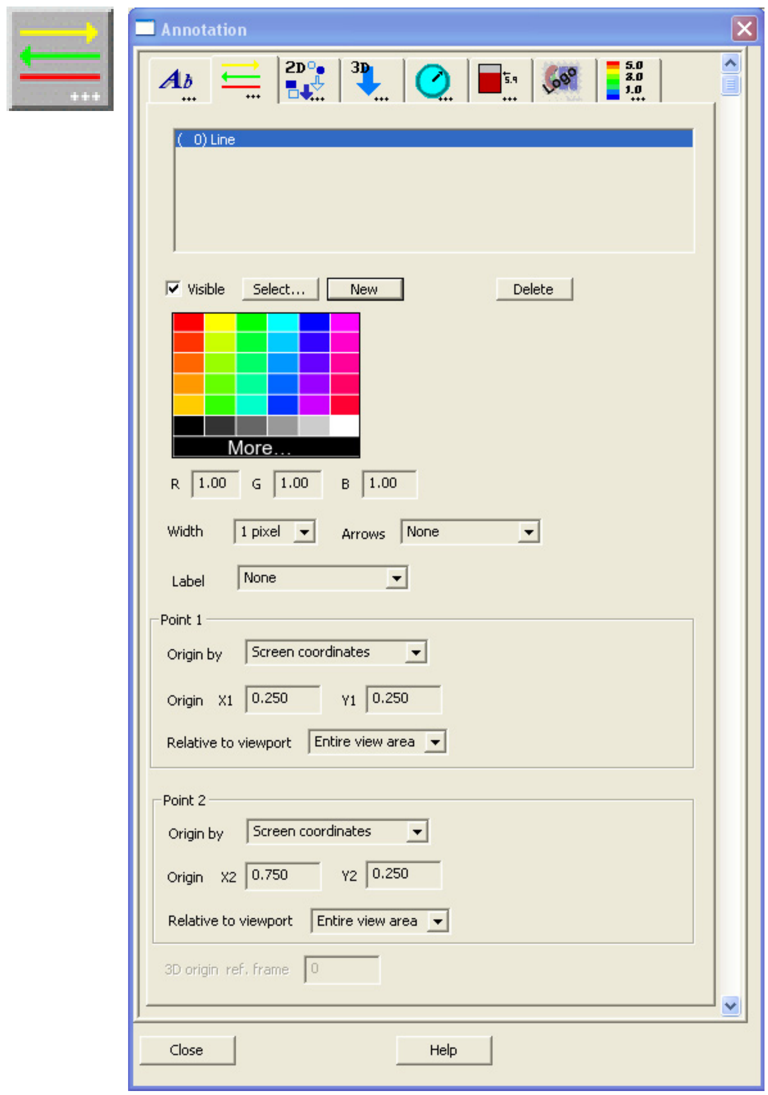
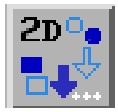


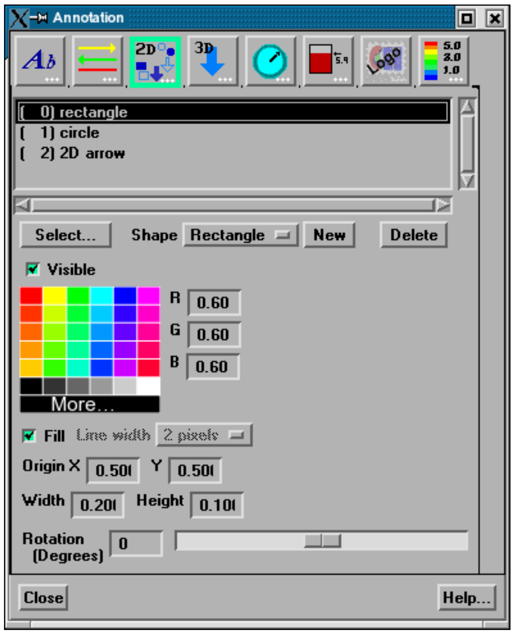
Figure 8-21
Annot Mode - Line Creation Icon

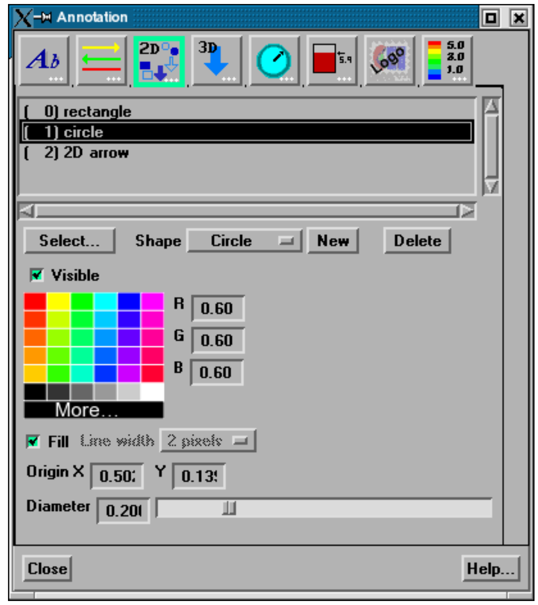
Select	Allows line(s) to be selected. They can also be selected using the mouse in the text menu, or by clicking on them in the Graphics Window.
New	Clicking New button creates a new line and places it in the graphics window.
Delete	Deletes the selected line(s) and removes the line(s) from the graphics window.
Visible	Toggles Line Visibility ON/OFF.

Width	Controls the line width in pixels.
Label	A text annotation can be used to label the line annotation. The text is aligned with the line orientation.
Arrows	Allows placement of arrow on line.
Origin by	Screen Coordinates indicates to locate the end point in normalized screen space. If set to 3D coordinates the end point is located in world coordinates and thus a X, Y, Z position must be specified.
Relative to viewport	For Screen Coordinates the origin can be located by screen coordinates or can be relative to the viewport specified. For 3D coordinates the end point can be visible in All viewports or in the viewport specified.
3D origin Ref Frame	If either of the end point origins are in 3D coordinates then the reference frame is specified here.

2D Shapes Create / Edit Icon — Clicking on the 2D Shapes Icon opens the Annotation Creation dialog with the 2D Shapes icon highlighted.







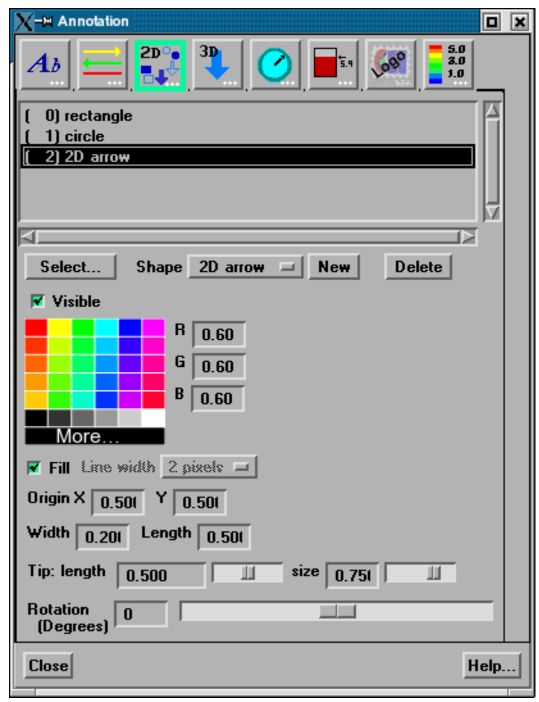


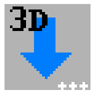
Figure 8-22
Annot Mode - 3D Annotation Arrow Creation Icon

Select	Allows 2D shapes to be selected. They can also be selected using the mouse in the text menu, or by clicking on them in the Graphics Window.
Shape	Allows selection of the type of 2D shape that will be created when the New button is selected.
New	Clicking New button creates a new 2D shape (according to the type selected in the Shape pulldown) and places it in the graphics window.
Delete	Deletes the selected 3D arrow(s) and removes the arrow(s) from the graphics window.
Visible	Toggles shape visibility ON/OFF.
Mix	Allows control of shape color
Fill	Toggles whether shape will be filled or just an outline.

Line Width	Pull down for line width selection, if not filling the shape.
Origin X Y	Fields for setting the center of rectangles and spheres, or tip of arrows in normalized coordinates - range 0 to 1.
Width/Height	Fields for setting the width and height of a rectangle shape in normalized coordinates - range 0 to 1.
Width/Length	Fields for setting the width and length of a shape in normalized coordinates - range 0 to 1.
Diameter	Field and slider for setting the diameter of a circle shape in normalized coordinates - range 0 to 1.
Tip: Length/Size	Fields and sliders for setting the arrow tip length and size.
Rotation/Degrees	Field and slider for rotation in degrees (-360 to 360).
	Access: Annot Mode : 2D Shape Creation Icon

3D Annotation Arrow Create / Edit Icon

Clicking on the 3D Annotation Arrow Icon opens the Annotation Creation dialog with the 3D Arrow tab highlighted.



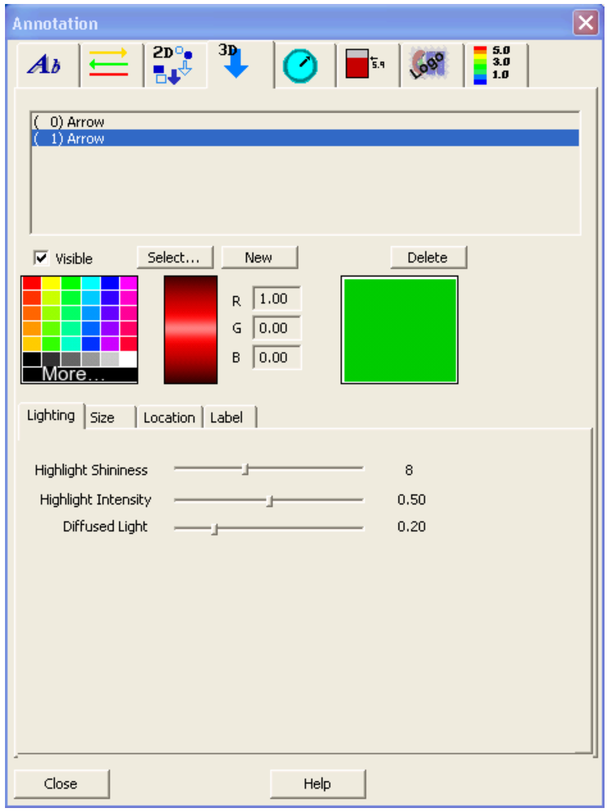


Figure 8-23
Annot Mode - 3D Annotation Arrow Creation Icon

Select — Allows 3D Arrow(s) to be selected. They can also be selected using the mouse in the text menu, or by clicking on them in the Graphics Window.

New — Clicking New button creates a new 3D arrow and places it in the graphics window at the center of the visible parts.

Delete — Deletes the selected 3D arrow(s) and removes the arrow(s) from the graphics window.

Visible — Toggles arrow visibility ON/OFF.

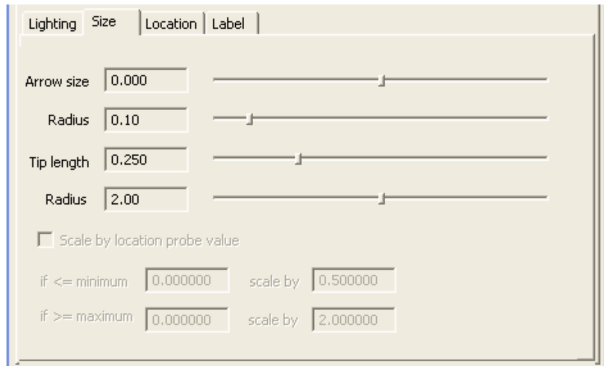
Mix — Allows control of arrow color

Viewport Visibility — Toggles the visibility of the 3D arrow(s) in the viewport. Clicking on the viewport toggles the visibility. Green indicates visible.

Highlight Shininess — Shininess factor. You can think of the shininess factor in terms of how smooth the surface is. The larger the shininess factor, the smoother the object. A value of 0 corresponds to a dull finish and larger values correspond to a more shiny finish. To change, use the slider.

Highlight Intensity — Highlight intensity (the amount of white light contained in the color of the arrow which is reflected back to the observer). Highlighting gives the arrow a more realistic appearance and reveals the shine of the surface. To change, use the slider. Will have no effect if Highlight Shininess parameter is zero.

Diffused Light — Diffusion (minimum brightness or amount of light that the arrow reflects). (Some applications refer to this as *ambient* light.) The arrow will reflect no light if value is 0.0. If value is 1.0, no lighting effects will be imposed and the arrow will reflect all light and be shown at full color intensity at every point. To change, use the slider.



Arrow:

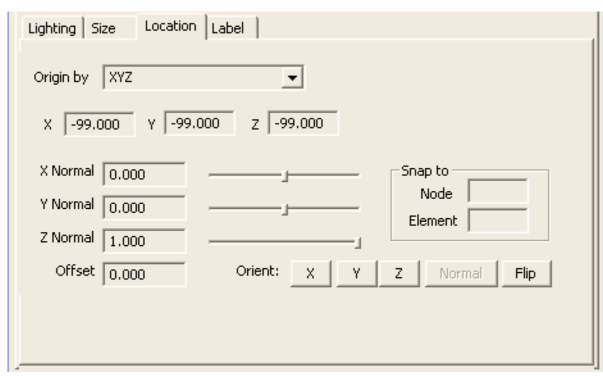
- Size — Set the length of the arrow in global units.
- Radius — Set the radius of the arrow as a percentage of the size.

Tip:

- Length — Set the length of the arrow tip as a percentage of the arrow size.
- Radius — Set the radius of the arrow tip as a percentage of the arrow size.

Scale by location probe/force/moment value (depends on which location method is used):

- low value — if below this value, scale by low scale factor
- low scale factor — the scale factor to use if below the low value
- high value — if above this value, scale by high scale factor
- high scale factor — the scale factor to use if above the high value



Arrow Location

Origin by:

Origin by:	Interactive probe query	Location set via the probe query capability
	XYZ	Location set by x,y,z coordinates
	One or more forces	Location set by external force vector glyph
	One or more moments	Location set by external moment vector glyph

If by Interactive probe query

- Probe # — The probe to use.
- Offset — This field specifies the distance away from the Arrow Location to display the arrow(s). A positive value moves the arrow(s) in the opposite direction that the arrow is pointing along the arrow axis.
- Orient: X, Y, Z — Orients the arrow direction parallel to the X, Y, or Z axis, respectively.
- Normal — Orients the arrow normal to a surface. This is only available if the Arrow Location has been picked on a 2D surface (see below).

Flip	Flips the arrow 180 degrees.
Normal X, Y, Z	Rotates the arrow axis about the X, Y, or Z axis.

If by XYZ:

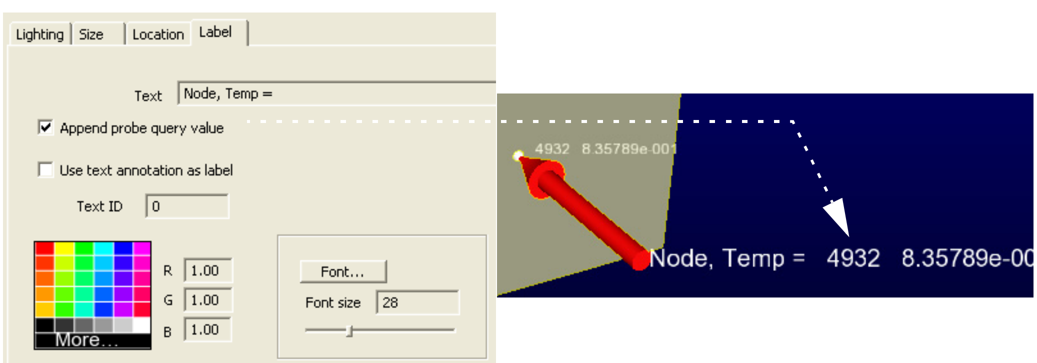
X	X-location of the arrow tip.
Y	Y-location of the arrow tip.
Z	Z-location of the arrow tip.
Snap to: Node	Enter the Node ID and press Enter and the arrow tip will be located at the coordinates of this Node. This field will set the Arrow Location X, Y, and Z values and is not used subsequently. That is, the arrow location is specified by the X, Y, and Z location and cannot be tied to a particular node number as the node changes location due to transient data, etc.
Snap to: Element	Enter the Element ID and press Return and the arrow tip will be located at the centroid coordinates of this Element. This field will set the Arrow Location X, Y, and Z values and is not used subsequently. That is, the arrow location is specified by the X, Y, and Z location and cannot be tied to a particular node number as the node changes location due to transient data, etc.
Offset	This field specifies the distance away from the Arrow Location to display the arrow(s). A positive value moves the arrow(s) in the opposite direction that the arrow is pointing along the arrow axis.
Orient: X, Y, Z	Orients the arrow direction parallel to the X, Y, or Z axis, respectively.
Normal	Orients the arrow normal to a surface. This is only available if the Arrow Location has been picked on a 2D surface (see below).
Flip	Flips the arrow 180 degrees.
Normal X, Y, Z	Rotates the arrow axis about the X, Y, or Z axis.

If by one or more forces:

Offset	This field specifies the distance away from the Arrow Location to display the arrow(s). A positive value moves the arrow(s) in the opposite direction that the arrow is pointing along the arrow axis.
Selection list	Descripton of loaded force vector glyphs from which to choose.

If by one or more moments:

Offset	This field specifies the distance away from the Arrow Location to display the arrow(s). A positive value moves the arrow(s) in the opposite direction that the arrow is pointing along the arrow axis.
Selection list	Descripton of loaded moment vector glyphs from which to choose.



Arrow Label

Text	Enter the label to be displayed on the 3D arrow. Field is not used if the "Use text annotation as label" toggle is on.
Append probe query value	If the Origin (Location tab) of the 3D arrow is set to "Interactive probe query" the probe value can be appended onto the label.

Use text annotation	A text annotation can be used as the 3D label. Specify the Text ID number in the field.
RGB	Color for the label.
Font	Pick the font for the label text.
Size	Size of the font used as the label

Picking the arrow location is accomplished using the pick icon above the graphics screen.

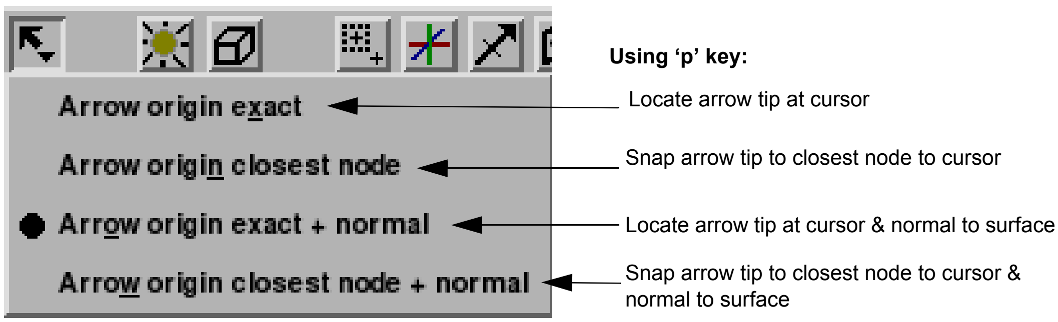


Figure 8-24
Annot Mode - Pick Menu

Dial Create/ Edit Clicking the Dial Icon opens the Annotation dialog. In order to use this feature, you must have one or more constant variables.

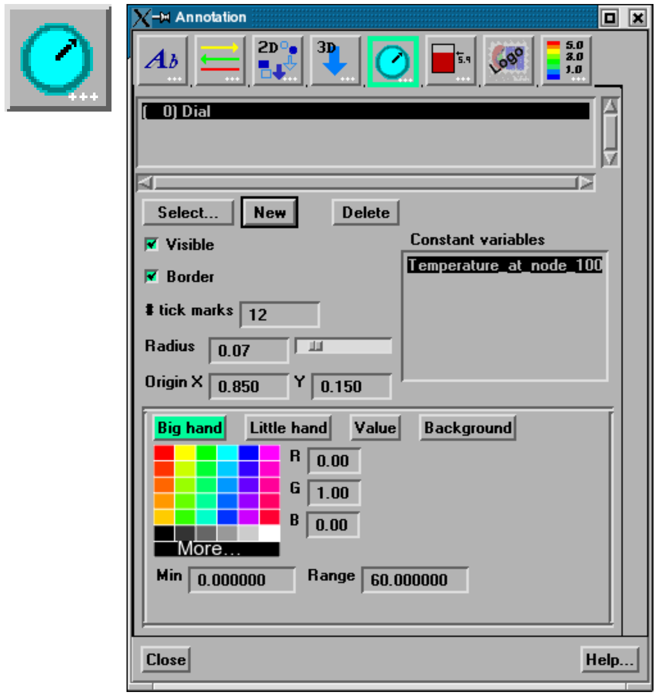
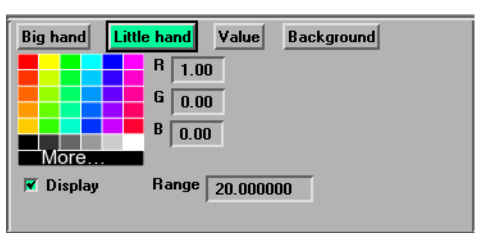


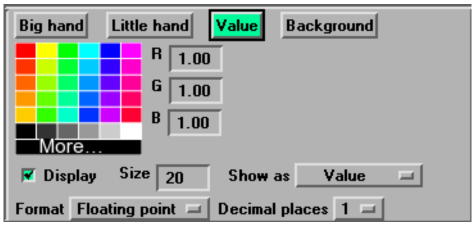
Figure 8-25
Annot Mode - Dial Icon

Access: Annot Mode : Dial Icon

Select	Allows Dial(s) to be selected. They can also be selected using the mouse in the text menu, or by clicking on them in the Graphics Window.
New	Clicking New button will create a new dial according to the current attributes. You must have first selected a constant variable or an error message will be issued.
Delete	Deletes the selected Dial(s) and removes the Dial(s) from the graphics window.
Constant Variable	List of available constant variables.
Visible	Toggles Dial Visibility ON/OFF.
Border	Toggles border for the dial ON/OFF.
# tick marks	Field to control the number of tick marks around the dial.
Radius	Field and slider to control the radius of the dial.
Origin	Location of center of the dial in normalized X and Y coordinates.
Big hand	Controls color, min, and range of the big hand on the dial.
Little hand	Controls color, range and display toggle of the little hand on the dial.



Value — Controls the size (and format) of the displayed value, whether to use value or revolution, and the display toggle.



Background — Controls the color and display toggle of the dial background.

Gauge Create / Edit Clicking the Gauge Icon opens the Annotation dialog. In order to use this feature, you must have one or more constant variables.

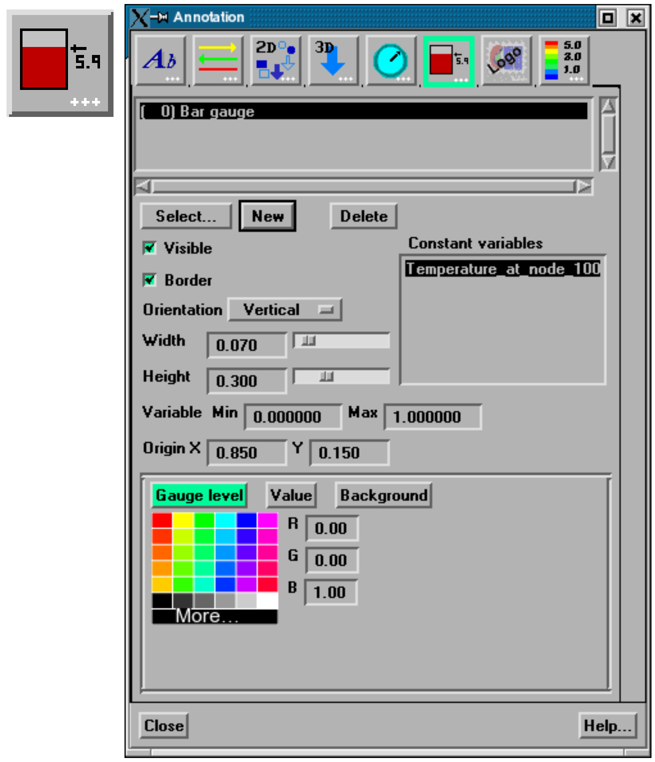
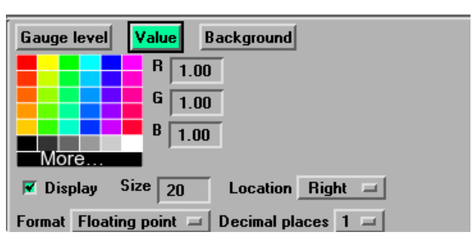


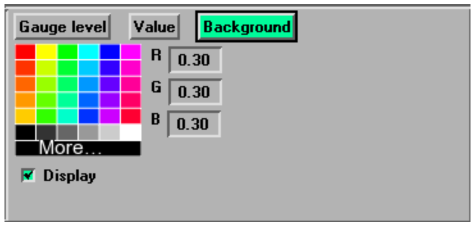
Figure 8-26
Annot Mode - Gauge Icon

Access: Annot Mode : Gauge Icon

Select	Allows Gauge(s) to be selected. They can also be selected using the mouse in the text menu, or by clicking on them in the Graphics Window.
New	Clicking New button will create a new gauge according to the current attributes. You must have first selected a constant variable or an error message will be issued.
Delete	Deletes the selected Gauge(s) and removes the Gauge(s) from the graphics window.
Constant Variable	List of available constant variables.
Visible	Toggles Gauge Visibility ON/OFF.
Border	Toggles border for the gauge ON/OFF.
Orientation	Pulldown for setting the gauge to be vertical or horizontal.
Width/Height	Fields and sliders to control the size of the gauge.
Variable Min/Max	Fields for setting the min and max for the variable.
Origin	Location of lower left of the gauge in normalized X and Y coordinates.
Gauge level	Controls color of the gauge level indicator.
Value	Controls color, size, location, format, and display toggle of the gauge label.



Background Controls color and display toggle of the gauge background.



Logo Import / Edit Icon

Clicking the Logo Import Icon opens the File Selection dialog for the specification of the file name containing the desired logo.

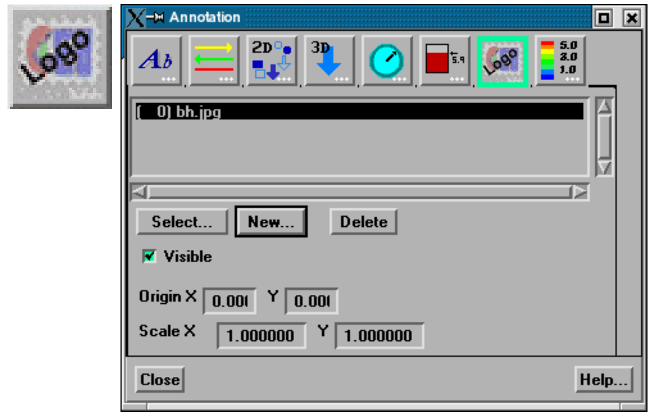


Figure 8-27
Annot Mode - Logo Import Icon

Access: Annot Mode : Logo Import Icon

Select	Allows Logo(s) to be selected. They can also be selected using the mouse in the text menu, or by clicking on them in the Graphics Window.
New	Clicking New button opens a dialog to load a new Logo.
Delete	Deletes the selected Logo(s) and removes the Logo(s) from the graphics window.
Visible	Toggles Logo Visibility ON/OFF.
Origin	Normalized X and Y location
Scale	X and Y scaling

Legend Create / Edit Clicking on this icon opens the Annotation dialog.

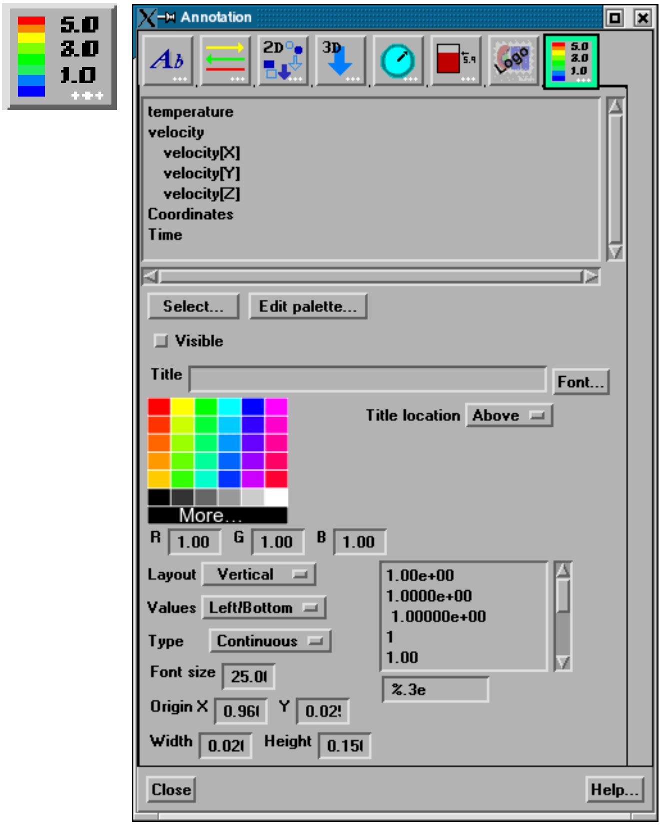


Figure 8-28
Annot Mode - Legend Create / Edit Icon and Dialog

Select	Allows Legend(s) to be selected. They can also be selected using the mouse in the text menu, or by clicking on them in the Graphics Window.
Edit Palettes	Brings up the Feature Detail Editor for Variables and allows you to edit palette attributes.
Visible	Toggles Legend Visibility ON/OFF.
Mix	Click on a color to change the legend text color, or click on More to create a color yourself.
Title	Field for entering text for the legend title.
Font	Brings up dialog for selecting font for legend title.
Title location	Location of the Title: Above, Below or None
Layout	Layout of the Legend Bar: Vertical or Horizontal
Values	Location of the legend values: Left/Bottom, Right/Top, or None
Type	Color distribution: Continuous or Discrete
Font Size	Choose a Font Size
Origin	Location of Legend can be typed in precisely, or just drag it in the Graphics Window where you want it located
Width, Height	Type in an exact value, or just grab one of the legend bar highlighted corners and drag it.
Value Format	Choose a format for the legend numbers.

Select All...

Brings up the Annotation Selection Options Dialog which allows selection of the various annotation types.

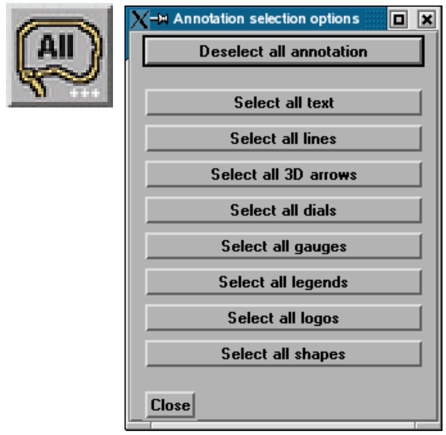


Figure 8-29
Annot Mode - Select All... Icon and Annotation Selection Options Dialog

Delete Icon

Deletes selected text, line or logo Annotation object(s).

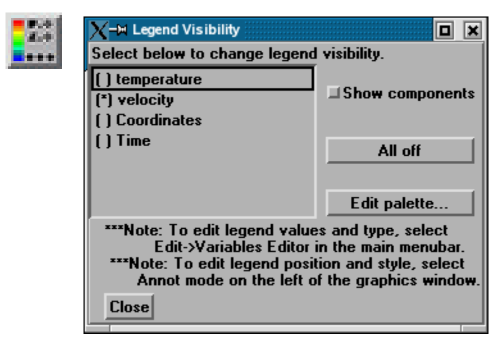
Figure 8-30
Annot Mode - Delete Icon

You cannot delete legends using the Delete Icon. Once a legend for a specific variable has been made visible using the Show Legend button, it can be made non-visible using either the Annot: Visibility Toggle or the Annot: Global Legend Visibility Toggle Icon.

Access: Annot Mode : Delete Icon

Legend...

Clicking the Legend... button (located on the Desktop above the graphics window) will allow the user to control which legends are visible.



8.3 Plot Mode

Plot Mode is used to adjust the attributes of 2D graphs and curves that you have created. You create graphs using the Query/Plot Editor in the Quick Interaction Area

In Plot Mode, there are two things you can select.

1. You can click the plot window and the whole plot window will have a highlighted box indicating its selection (green in color by default).

2. You can click on an individual curve to cause the curve line to thicken - indicating selection.

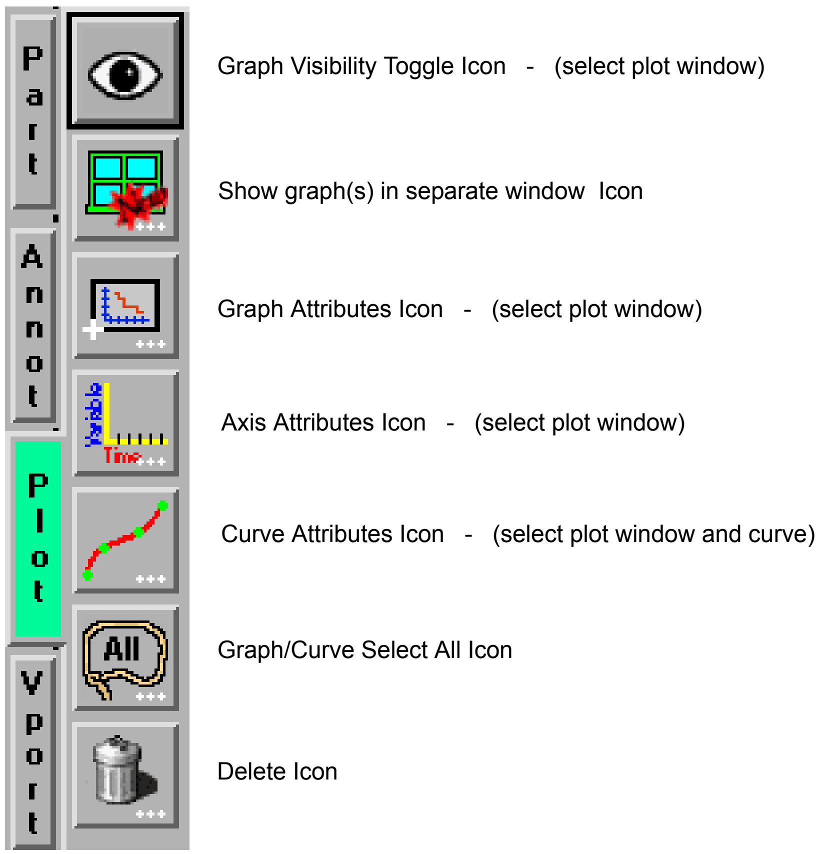


Figure 8-31
Mode Selection Area - Plot Selected

The plotting capability is limited to simple x versus y data. Most often this data is in the form variable value versus Time or Distance.

Graph Visibility Toggle Icon

Determines the visibility of the selected plotters. The selected plotters for which visibility has been toggled off will appear grayed-out in Plot Mode.

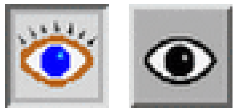


Figure 8-32
Plot Mode - Plotter Visibility Toggle Icon

Access: Plot Mode : Plotter Visibility Toggle Icon

Show Graph in Separate Window Icon

Opens the Plot detached window options dialog for the specification of the background color, and visibility of the detached window for the selected plot(s).

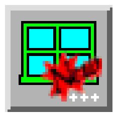
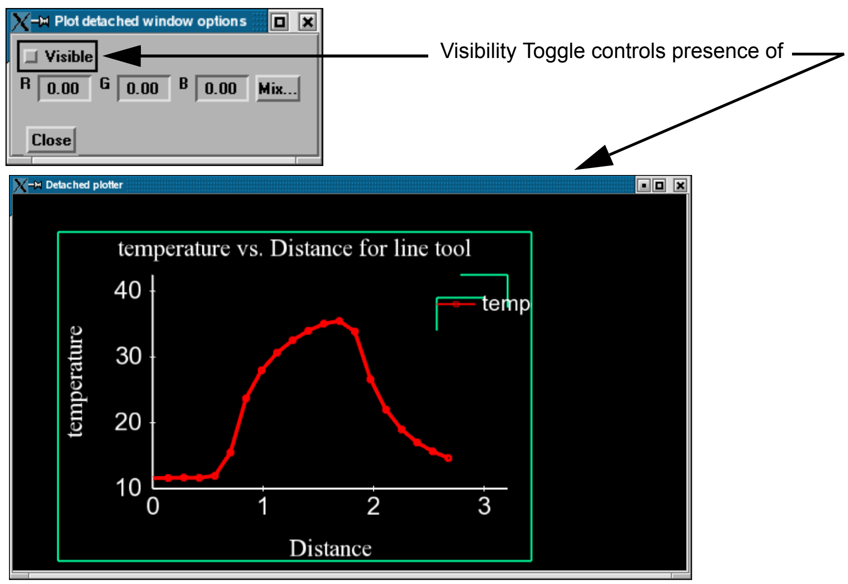


Figure 8-33
Plot Mode - Show Graph in Separate Window Icon and Plotter Specific Attributes dialog



Graph Attributes Icon Opens the Plotter Specific Attributes dialog for the specification of attributes for Title, background, legend, border and position of the selected plotters.

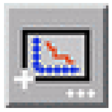
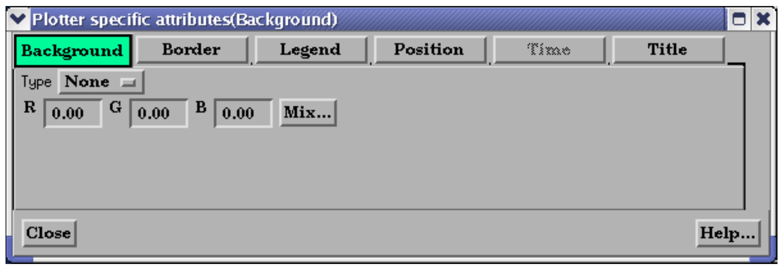


Figure 8-34
Plot Mode - Plotter Attributes Icon and Plotter Specific Attributes dialog

Clicking the Background tab causes the dialog to configure itself for Plotter Background editing

Background

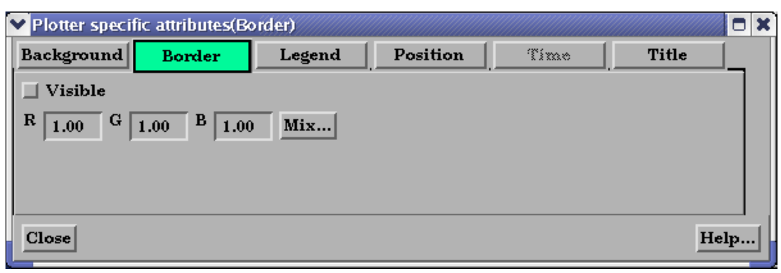


Type Opens a pop-up menu for the specification of plotter background color. Choices are:
None no background (the color of the Graphics Window or the viewport underneath will show through the Plotter)
Solid allows a solid color to be specified for the Plotter Background

RGB Mix... Color for the Plotter background may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button.

Border

Clicking the Border tab causes the dialog to configure itself for Plotter Border editing



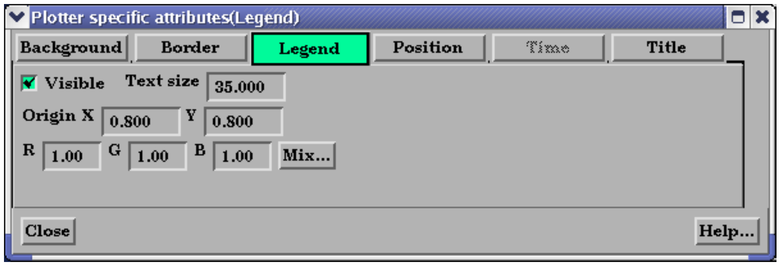
Visible Toggle Toggles on/off the visibility in the other five Modes of the Border for the selected Plotters.

RGB Mix... Color for the plotter border may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button. Note that the border color is not shown while the plotter is selected - while selected the border is shown in green.

See Part Color, Lighting, Transparency

Legend

Clicking the Legend tab causes the dialog to configure itself for Plotter Legend editing. The legend shows a line of the appropriate color, width, and marker next to the name of the curve plotted using this line style.



Visible Toggle — Toggles on/off the visibility of the legend for the selected Plotters.

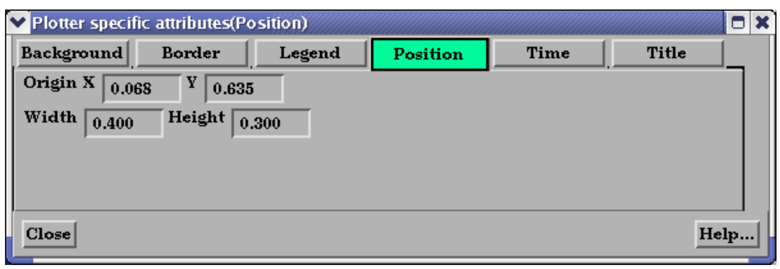
Text Size — This field specifies the desired size of the Legend text.

Origin X Y — These fields specify the location of the Legend within a Plotter's border. Values range from 0.0 to 1.0 and resulting distances are measured from the Border origin (lower left corner). These fields provide an alternative to interactively positioning the plotter Legend.

RGB Mix... — Color for the Legend text may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button.

Position

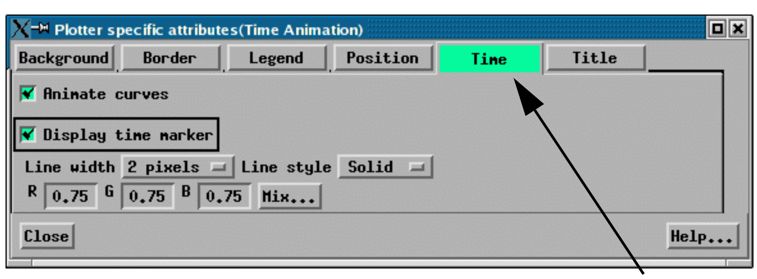
Clicking the Position tab causes the dialog to configure itself for Plotter Position.



Origin X Y — These fields specify the location of the selected Plotter within the Graphics Window. Values range from 0.0 to 1.0 and resulting distances are measured from the Graphics Window origin (lower left corner). These fields provide an alternative to interactively positioning the plotter which is done simply by clicking within the Plotter and dragging it to the desired position.

Width, Height — These fields specify the width and height of the Plotter. Resulting distances are measured from the Border origin (lower left corner). These fields provide an alternative to interactively resizing the plotter which is done simply by clicking on a side or corner and dragging.

Time



This tab is enabled only for plots with transient data.

Animate Curve — When the solution time is changed or streamed (data loaded from disk one step after another) or loaded in the flipbook, toggling this ON will cause the transient curve to plot the data timestep by timestep.

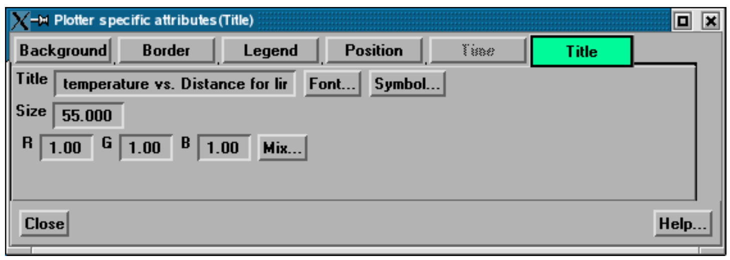
Display Time Marker — The Time Marker is a vertical time indicator line that appears only on a transient plot. When the solution time is changed or streamed (data loaded from disk one step after another) or loaded in the flipbook, toggling this ON will cause the vertical line Time Marker to indicate the current timestep value on the plot. Click on the time indicator line and drag it while in part mode to change the time, and the time indicator line will jump to the next time value on your plot.

Line Width — Set the line width of the vertical time indicator line on the transient plot.

Line Style — Set the line style of the vertical time indicator line on the transient plot.

RGB — Set the color (RGB values) of the vertical time indicator line on the transient plot.

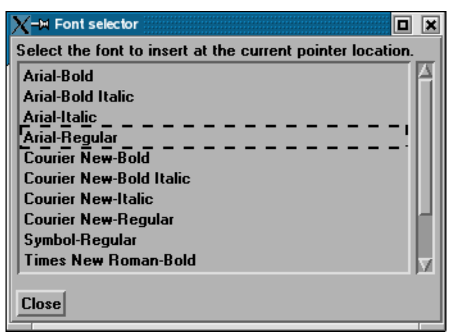
Title



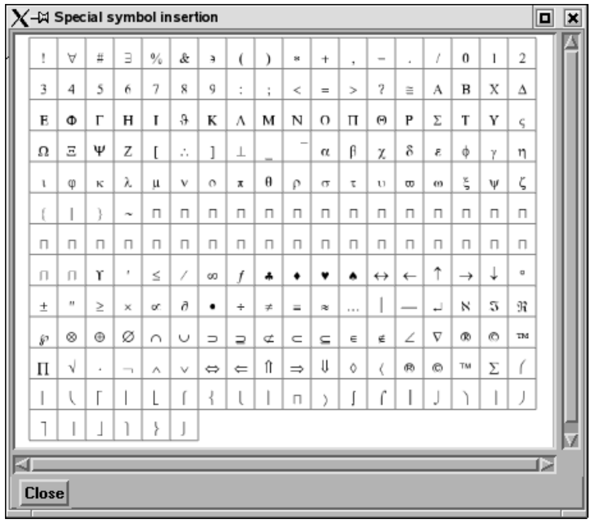
Clicking the Title tab causes the dialog to configure itself for Graph Title editing.

Title — This field allows you to edit the existing plotter title.

Font... — This button brings up a dialog allowing you to pick the desired font. It inserts codes into the text string at the location of the cursor.



Symbol... This button brings up a matrix of symbols to pick from to allow you to insert symbols into the plotter title.



Size This field allows you to specify the title text size.

RGB Mix... Color for the Title text may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button.

Access: Plot Mode : Plotter Attributes Icon

Axis Attributes Icon Opens the Axis Specific Attributes dialog for the specification of attributes for axes of the selected Plotters.

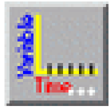
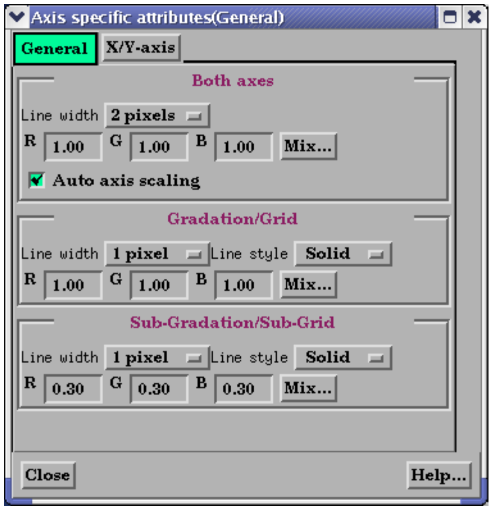


Figure 8-35
Plot Mode - Axis Attributes Icon and Axis Specific Attributes dialog

General



Clicking the General tab causes the dialog to configure itself for General Plotter Axis editing.

Both Axes

Line Width — Opens a pop-up menu for the specification of the desired line width (1 to 4 Pixels) for Plotter axes.

RGB Mix... — Color for the axes may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button

Auto Axis Scaling Toggle — When toggled on, the axis range and number of divisions will be scaled to make nice “round” numbers.

Gradation/Grid

Line Width — Opens a pop-up menu for the specification of the desired width (1-4 Pixels) for Gradation Lines or Ticks.

Line Style — Opens a pop-up menu for the specification of the style of line (Solid, Dotted, or Dashed) desired for gradations. (The lines are normally not visible and so this specification is only valid if Grad Type has been selected to Grid in the X-Axis and/or Y-Axis configuration of the Axis Specific Attributes dialog.)

RGB Mix... — Color for the Gradation Lines or Ticks may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button.

Sub-Gradation/Sub-Grid

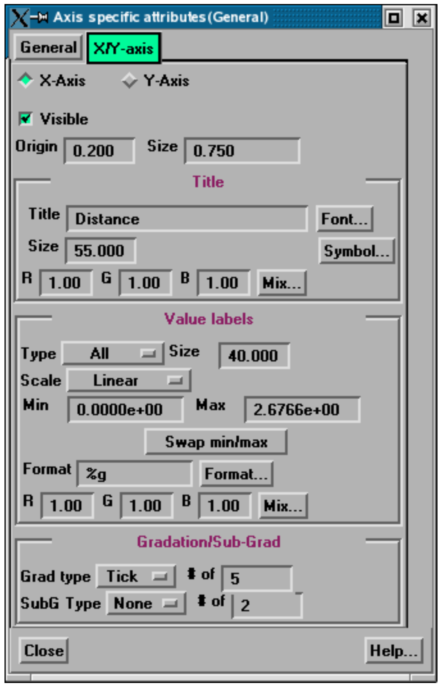
Line Width — Opens a pop-up menu for the specification of the desired line width (1-4 Pixels) for Sub-Gradation Lines or Ticks (those between the Gradation Lines or Ticks).

Line Style — Opens a pop-up menu for the specification of the style of line (Solid, Dotted, or Dashed) desired for sub-gradations. (The lines are normally not visible and so this specification is only valid if SubG Type has been selected to Grid in the X-Axis and/or Y-Axis configuration of the Axis Specific Attributes dialog.)

RGB Mix... — Color for the Sub-Gradation Lines or Ticks may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button

.

X-Axis or Y-Axis

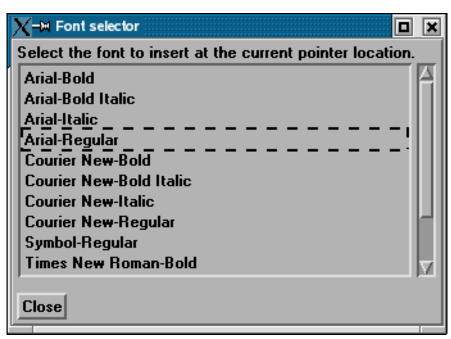


Clicking the X-Y Axis tab causes the dialog to configure itself for editing of Attributes specific to either the X or the Y Axis. If X-Axis toggle has been clicked, the dialog will affect the X-Axis attributes only. Likewise for Y-Axis.

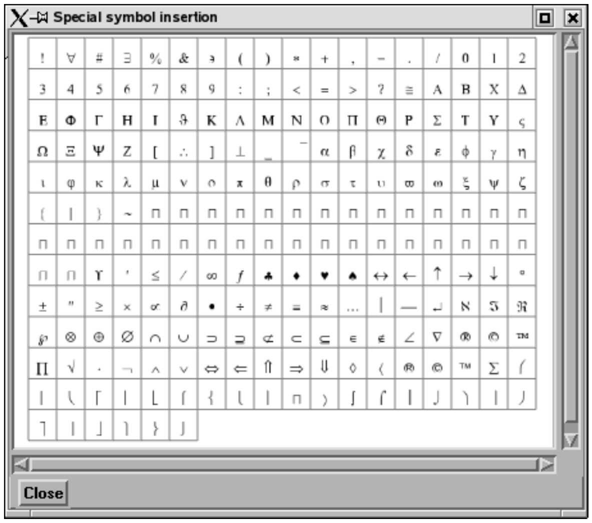
Visible Toggle	Toggles on/off the visibility of the X (or Y) Axis line.
Origin	This field specifies the location of the X (or Y) Axis origin. Values range from 0.0 to 1.0 and resulting distances are measured from the left side (or bottom) of the Plotter.
Size	This field specifies the length of the X (or Y) Axis line. Values range from 0.0 to 1.0 and resulting distances are measured from the X (or Y) Axis Origin.

Title

Title	This field allows you to edit the existing X (or Y) Axis title.
Size	This field allows you to specify the title text size.
Font...	This button brings up a dialog allowing you to pick the desired font. It inserts codes into the graph axis title at the location of the cursor.



Symbol... This button brings up a matrix of symbols to pick from to allow you to insert symbols into the graph axis title.



RGB Mix... Color for the Title text may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button.

Value Labels

Type Opens a pop-up menu for selection of desired number (None, All, or Beg/End) of X (or Y) Axis labels.

Size This field allows you to specify the size of X (or Y) Axis labels.

Scale This field allows you to specify a linear or log10 scale for the Axis.

Min This field contains the minimum value of the X (or Y) Axis. If Auto Axis Scaling is on, it is only an approximation to the value which will be used.

Max This field contains the maximum value of the X (or Y) Axis. If Auto Axis Scaling is on, it is only an approximation to the value which will be used.

Format This field specifies the format used to display the X (or Y) Axis. Any C language *printf* format is valid in this field.

Format... This button will open the Format dialog which allows you to select a pre-defined format.

Gradation/Sub-Grad

Grad Type
 # of Opens a pop-up menu for selection of desired marker (None, Grid, or Tick) for major gradations. # of field specifies the number of major gradations you wish along the X (or Y) Axis. If Auto Axis Scaling is on, it is only an approximation to the value which will be used.

SubG Type
 # of Opens a pop-up menu for selection of desired marker (None, Grid, or Tick) for sub gradations (between the major gradations. # of field specifies the number of sub gradations you wish between each major gradation along the X (or Y) Axis.

Access: Plot Mode : Axis Attributes Icon

Curve Attributes Icon Opens the Curve Specific Attributes dialog for the specification of attributes for an individual curve which has been selected in a Plotter. A curve is selected by clicking the mouse cursor on top of the curve. The selected curve will be drawn by a wider line than is normally used to display the curve.

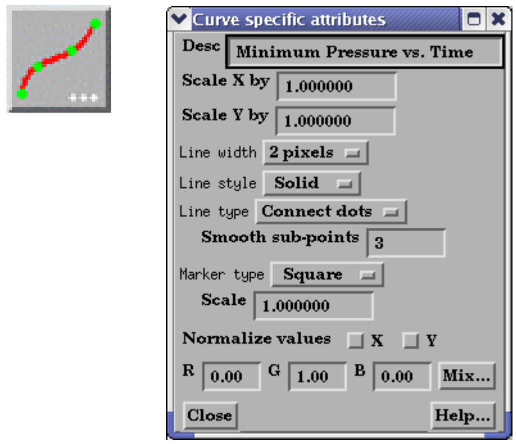


Figure 8-36
Plot Mode - Curve Attributes Icon and Curve Specific Attributes dialog

Desc. This field initially contains the Legend description of the selected Curve which was assigned to it when the Curve was created. Edit and press Return to change the description.

Scale X/Y By Scales the respective X or Y values by the specified factor.

Line Width Opens a pop-up menu for the specification of the desired line width (1-4 Pixels) for the selected Curve.

Line Style Opens a pop-up menu for the specification of the style of line (Solid, Dotted, or Dashed) desired for the selected Curve.

Line Type Opens a pop-up menu for the specification of the type of line desired for the selected Curve. Options are:
None No lines will be drawn between points
Connect Dot Lines will be drawn between the points
Smooth A piece wise spline will connect the points

Smooth Sub-points This field specifies the number of sub-points to use between data points in drawing the curve when Smooth Line Type is selected.

Marker Type Opens a pop-up menu for the specification of the desired type of data point marker (None, Dot, Circle, Triangle, Square) on the curve.

Scale This field specifies the size of the data point markers for the selected curve.

RGB Mix... Color for the selected Curve may be specified using either the RBG fields or the Color Selector dialog which is opened by clicking the Mix... button

Access: Plot Mode : Curve Attributes Icon

Select All... Brings up the Plot Selection Option Dialog which allows selection of all curves or all plotters.

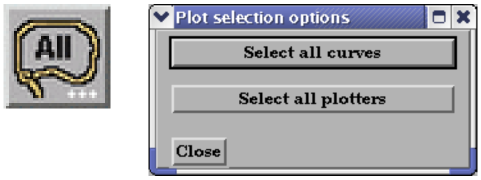


Figure 8-37
Plot Mode - Select All... Icon and Plot Selection Options Dialog

Delete Icon Deletes the selected Plotters.

Figure 8-38
Plot Mode - Delete Icon

Access: Plot Mode : Delete Icon

8.4 VPort Mode

VPort Mode is used to create, adjust the attributes of, and delete viewports.

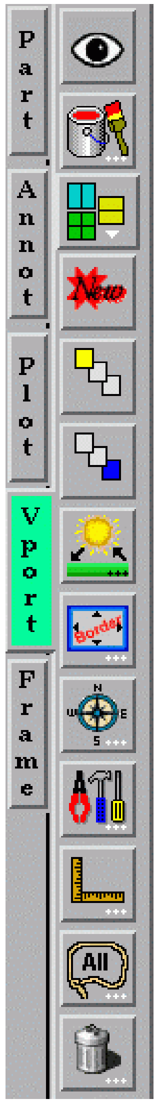


Figure 8-39
Mode Selection Area - VPort Selected

The default EnSight configuration shows one view of your model in the “main” Graphics Window. This “initial viewport”, which covers the Graphics Window. cannot be removed and is always used to clear (erase) the Graphics Window prior to a redraw. Using the VPort Mode, you can create up to fifteen additional viewports that will overlay the Graphics Window. These viewports can be interactively resized and relocated within the Graphics Window using the mouse and the visibility of each Part can be controlled on a per viewport basis. Transformations, and Z-clip location settings can also be made independently in each viewport.

When in VPort Mode, you are always modifying the viewports selected in the Graphics Window. Selected viewports are outlined in the “selection color”, while unselected objects are outlined in a white color. To select a viewport, click the mouse over it. To select multiple viewports, hold the Control key down while clicking on them.

Multiple viewports are helpful in showing the same object from multiple views, showing different axes in each viewport, showing the same parts with different attributes, etc. Max number of viewports is 16.

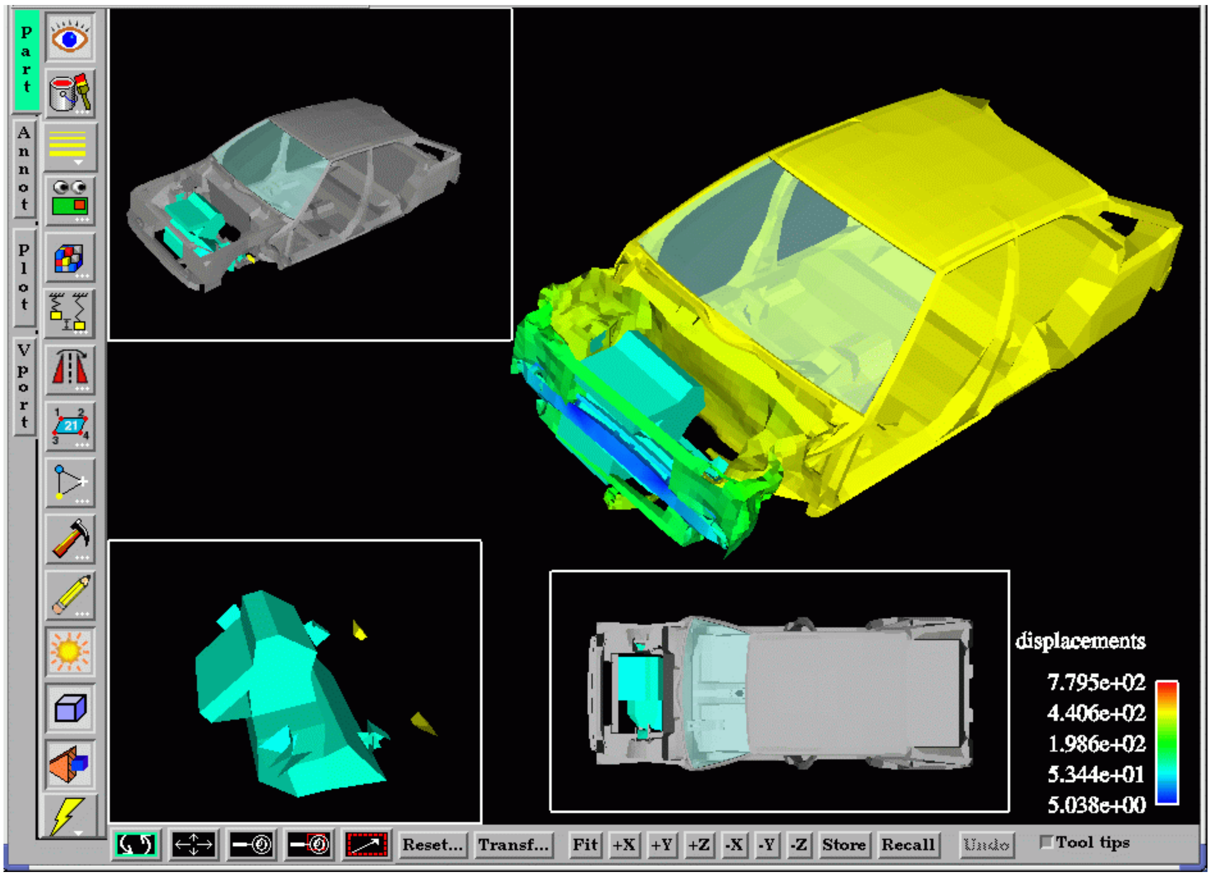


Figure 8-40
Viewport Example

Viewport Visibility Toggle Icon

Determines the visibility of the selected viewport(s). The border of a viewport which, in the VPort Mode, has its visibility toggled off will still be visible in the VPort Mode only - and then as a dotted rather than a solid line.

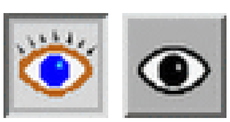


Figure 8-41
VPort Mode - Viewport Visibility Toggle Icon ON / OFF

Access: VPort Mode : Viewport Visibility Toggle Icon

Color Icon

Opens the Viewport Background Color Attributes dialog for the specification of the color you wish to assign to the background of the selected viewport(s).

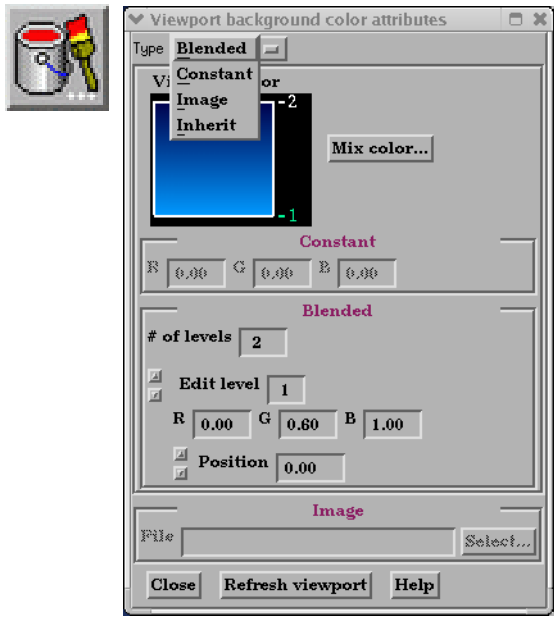


Figure 8-42
VPort Mode - Color Icon

Type

Opens a pull-down menu for the specification of the type of background you wish to assign to a viewport.

Blended

Allows you to specify a background comprised of 2 to 5 blended colors.

# of Levels

This field specifies the number of levels (from 2 to 5) at which a color will be specified. The default is 2.

Edit Level

This field specifies which of the levels you wish to edit. You may select the desired level using the stepper buttons, by entering a value in the field, or interactively by clicking on its number on the right side of the Viewport Color window.

Position

This field specifies the vertical position of the edit level as a fraction (from 0 to 1) of the vertical height of the Viewport Color window, where 0.0 is at the bottom and 1.0 is at the top. You may adjust a level to the desired position using the stepper buttons, by entering a value in the field, or interactively by selecting and dragging a level's number on the right side of the Viewport Color window. The position of any level can not be below the position of the next lower level.

Constant

Allows you to specify a constant color using the RGB fields or the Color Selector dialog which is accessed by clicking the Mix Color... button.

Image

Allows you to choose an image as a background for your viewport.

Inherit

Causes the viewport to display the same background color attributes as the main Graphics Window. Only applicable for created viewports, not the main Graphics Window.

Mix Color...

Opens the Color Selector dialog. (See Section 8.1, Color Selector)

Refresh Viewport

Will redraw the selected viewport(s) with the defined viewport background settings.

Access: VPort Mode : Color Icon

Viewport Layouts Icon This icon opens a pull-down menu of icons which indicate standard viewport layouts.

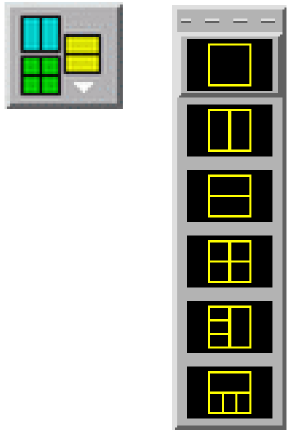


Figure 8-43
Viewport Mode - Viewport Layouts Icon and Menu

Access: Vport Mode : Viewport Layouts Icon

New Viewport Icon Clicking this button creates a new Viewport within (and on top of) the “main” Graphics Window. The location and size of the viewport can be modified interactively in the Graphics Window by a) clicking and dragging within the viewport to move it of b) clicking and dragging the edge or corner to resize it. More precise modifications may be performed using the Location... Icon.

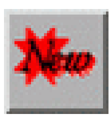


Figure 8-44
VPort Mode - New Viewport Icon

Access: VPort Mode : New Viewport Icon

Viewport Forward Icon Clicking this button moves the selected viewport(s) “forward” in the Graphics Window to occlude any viewports which it (they) may overlap. Viewport 0 cannot be “popped”.

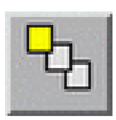


Figure 8-45
VPort Mode -Viewport Forward Icon

Access: VPort Mode : Viewport Forward Icon

Viewport Back Icon Clicking this button moves the selected viewport(s) “back” in the Graphics Window to be occluded by any viewports which may overlap it (them). Viewport 0 cannot be “pushed”.

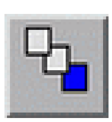


Figure 8-46
VPort Mode - Viewport Back Icon

Access: VPort Mode : Viewport Back Icon

Lighting Opens the Viewport Lighting Attributes Dialog.

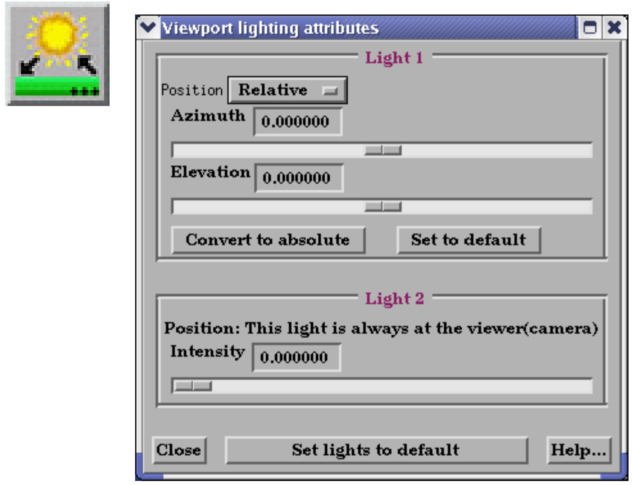


Figure 8-47
VPort Mode - Viewport Border Attributes Icon and Viewport Border Attributes dialog

Position	Position of Light source 1 is either absolute or relative to the camera position.
Azimuth	Azimuth Angle of Light source 1 (-180 to +180 degrees).
Elevation	Elevation of Light source 1 (-90 to +90 degrees).
Convert to Absolute	Converts a relative Azimuth and Elevation angle to Absolute Coordinates.
Set to Default	Restores Light Source 1 values back to default.
Intensity	Intensity of Light source 2 (0 to 1).

Access: VPort Mode : Viewport Lighting Attributes Icon

Viewport Border Attributes Icon Opens the Viewport Border Attributes dialog for the specification of a constant color for the border of the selected viewports. Be aware that the color assigned will only be visible in the other five Modes, not in VPort mode.

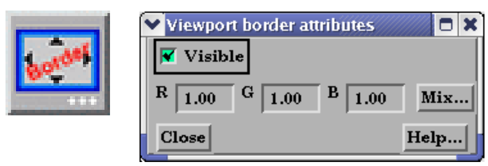


Figure 8-48
VPort Mode - Viewport Border Attributes Icon and Viewport Border Attributes dialog

Visible Toggle	Toggles on/off visibility of a viewport's border in the other five Modes. The border of each viewport will always be visible in VPort Mode.
RGB	These fields specify the RGB values for the color you wish to assign.
Mix...	Opens the Color Selector dialog. See Section 7.1 Color

Access: VPort Mode : Viewport Border Attributes Icon

Viewport Location Attributes Icon

Opens the Viewport Location Attributes dialog for the specification of the desired location in the main Graphics Window for the selected viewports. This dialog provides a more precise alternative to moving and resizing the viewports interactively.

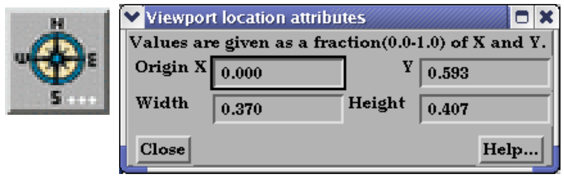


Figure 8-49
VPort Mode - Viewport Location Attributes Icon and Viewport Location Attributes dialog

Origin X Y

These fields specify the location for the X and Y coordinates of the selected viewport's origin (lower left corner) in the main Graphics Window. Values range from 0.0 to 1.0.

Width, Height

These fields specify the width and height of a selected viewport in X and Y coordinates from the viewport's origin. Values range from 0.0 to 1.0.

Access: VPort Mode : Viewport Location Attributes Icon

Viewport Special Attributes Icon

Opens the Viewport Special Attributes dialog for the specification of whether the global settings for Perspective versus Orthographic display, hidden surface display, and hidden line display will apply in the selected viewport(s). In addition, a viewport can be designated as 2D, in which case only planar 2D parts can be displayed in the viewport

Note, Once you designate a viewport as a 2D viewport, all 3D parts are no longer visible in that viewport. To see the 3D parts in that viewport again, you will need to make the viewport 3D, select the 3D parts in the parts list, and make them visible again using the visibility per viewport icon.

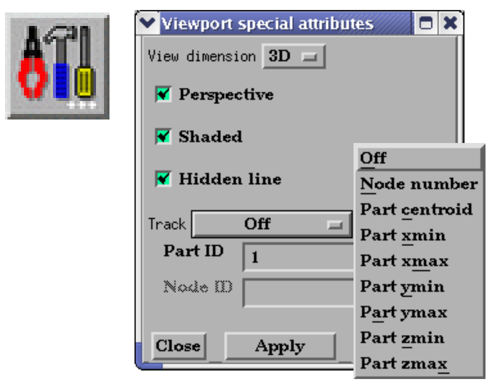


Figure 8-50
VPort Mode - Viewport Special Attributes Icon and Viewport Special Attributes dialog

Track

Will do a global translate of the geometry to track the motion with respect to one of the following:
Node number - Enter the node id in the Node ID field
Part Centroid - The geometric centroid of the part is used for tracking
Part xmin - The minimum x coordinate of the part is used for tracking.
Part xmax - The maximum x coordinate of the part is used for tracking
Part ymin - The minimum y coordinate of the part is used for tracking
Part ymax - The maximum y coordinate of the part is used for tracking
Part zmin - The minimum z coordinate of the part is used for tracking
Part zmax - The maximum z coordinate of the part is used for tracking

Part ID

ID number of Part used for camera tracking

Node ID
Node ID for the Part ID used for tracking if Node number selected in Track pulldown.

Access: VPort Mode : Viewport Special Attributes Icon

Part Bounds Attributes

Opens the Viewport Bounds Attributes dialog for the specification of part bounding box gradation and labeling.

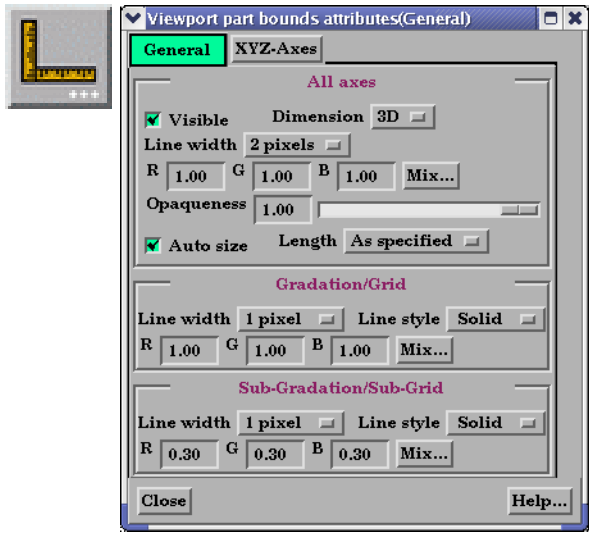


Figure 8-51
VPort Mode - Viewport Part Bounds Icon and Viewport 2D/3D Grid Attributes (General) dialog

General
Change the dialog to reflect overall extent grid attributes.

XYZ-Axis
Change the dialog to reflect XY or Z-Axis extent attributes (see below).

All Axes Section
Contains parameters that affect attributes of all axes of the bounding extent.

Visible Toggle
Toggles on/off visibility of the viewport 2D/3D extent axes.

Dimension
Opens a menu for the specification of the desired dimension (2D or 3D) of the bounding extent axis. Default is 3D. 2D is only available for 2D viewports and 3D viewports in orthographic mode.

Line Width
Opens a menu for the specification of the desired line width (1 - 4 pixels) of the bounding extent axis. Default is 2.

RGB
These fields specify the RGB values for the color you wish to assign.

Mix Color...
Opens the Color Selector dialog. See Part Color, Lighting, Transparency

Transparency
Specifies the degree of opaqueness for the axes of the bounding extent. This value may be adjusted by typing in a value from 0.0 to 1.0 in the field or by using the slider bar whose current value is reflected in the field. A value of 0. or 1. will render the axes completely transparent or completely opaque, respectively.

Auto Size Toggle
Toggles on/off the scaling of the axis range to nice "round" numbers.

Length
Opens a menu for the specification of the desired type of length gradation on the axes of the bounding extent.

As Specified Divides the gradations evenly along the length of the axis. (default)
Rounded Tries to round to the units of the common order of magnitude.

Gradation/Grid Section
Controls the specification of the gradation/grid of the axes of the bounding extent.

Line Width
Opens a menu for the specification of the desired line width (1 - 4 pixels) for the gradation and grid of the bounding extent axis. Default is 1.

Line Style — Opens a menu for the specification of the style of line (Solid, Dotted, or Dashed) desired for the gradation and grid of the bounding extent axis. Default is Solid.

RGB — These fields specify the RGB values for the color you wish to assign.

Mix Color... — Opens the Color Selector dialog. See Part Color, Lighting, Transparency

Sub-Gradation/ Sub-Grid Section — Controls the specification of the subgradation/subgrid of the axes of the bounding extent.

Line Width — Opens a menu for the specification of the desired line width (1 - 4 pixels) for the subgradation and subgrid of the bounding extent axis. Default is 1.

Line Style — Opens a menu for the specification of the style of line (Solid, Dotted, or Dashed) desired for the subgradation and subgrid of the bounding extent axis. Default is Solid.

RGB — These fields specify the RGB values for the color you wish to assign.

Mix Color... — Opens the Color Selector dialog. See Part Color, Lighting, Transparency

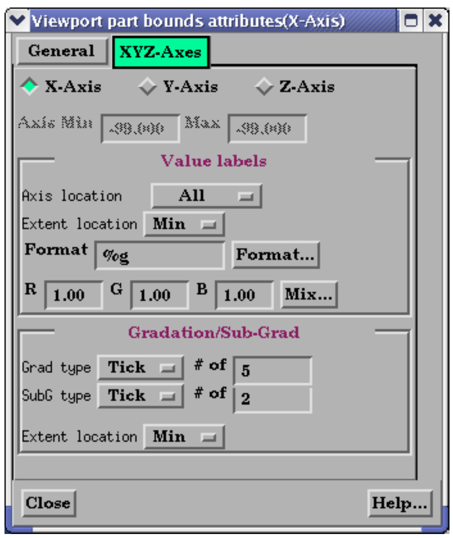


Figure 8-52
Viewports 2D/3D Grid Attributes (X-Axis or Y-Axis or Z-Axis dialog)

X-, Y-, Z-Axis — These toggles choose the axis.

Axis — These fields reflect the Min and Max values of the selected axis.

Value Labels Section — Controls attributes of the selected axis.

Axis Location — Opens a menu for the specification of the desired location (None, All (default), or Beg/ End) to place the labels of the selected axis.

Extent Location — Opens a menu for the specification of the desired extent (Min (default), Max, or Both) on which to display the labels of the selected axis.

Format — This field specifies the format used to display the labels of the selected axis. Any C language “printf” format is valid in this field.

Format... — Opens the Viewport Axis Text Format dialog which allows you to select a pre-defined format.

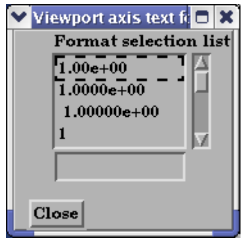


Figure 8-53
Viewport Axis Text Format dialog

Format Selection List — List of pre-defined formats. You can also enter any legal C language format string for floating point numbers.

Format Selection Field — The format specification.

RGB — These fields specify the RGB values for the color you wish to assign.

Mix Color... — Opens the Color Selector dialog. See Part Color, Lighting, Transparency

Gradation/Sub-Grad Section

Grad Type — Opens a menu for the selection of desired marker (None, Grid, or Tick (default)) for major gradations of the selected axis.

\# of — Specifies the number of major gradations you wish along the selected axis. If Auto Size is on, it is only an approximation to the value which will be used.

SubG Type — Opens a menu for the selection of desired marker (None, Grid, or Tick (default)) for sub-gradations (between the major gradations) along the selected axis.

\# of — Specifies the number of sub-gradations you wish between each major gradation along the selected extent axis.

Extent Location — Opens a menu for the specification of the desired extent (Min (default), Max, or Both) on which to display the gradation/sub-gradations of the selected axis.

Access: VPort Mode : Viewport Part Bounds Attributes Icon

Select All — Selects all of the currently defined viewports.

Figure 8-54
VPort Mode - Select All Icon

Delete Icon — Deletes individual, selected viewports. (You cannot delete viewport 0).

Figure 8-55
VPort Mode - Delete Icon

Access: VPort Mode : Delete Icon

8.5 Frame Mode

As EnSight reads in model Parts, they are all initially assigned to the same "Frame" of reference: Frame 0. Frame 0 corresponds to the model coordinate system (defined when the model was created). Using the Frame Mode, you can create additional frames, reassign Parts to different Frames, and specify various attributes of the Frames.

Transformations you make in View or Parts Mode (rotations, translations, etc.) are performed globally; all Frames, Parts, and Tools are transformed with respect to the Global Axis origin and orientation. Frame Mode, on the other hand, allows you to perform transformations on selected Parts. This is useful if you wish, for example, to create an animation with Parts moving in different directions (such as a door or hood opening to reveal Parts within) or to move Part copies away from each other in order to color the Parts by different variables (in fact, if you make a copy of a Part, a new Frame is automatically created and the Part copy is assigned to it). Note that the Frame mode coordinate transformations are visual only and occur only on the client. That is, the transformed coordinates cannot be used in the EnSight calculator.

In Frame Mode, transformations are always about the selected Frame's definition, that is, its origin position (with respect to Frame 0) and the orientation of its axes (with respect to Frame 0). Since this is the case, the Frame's orientation must be adjusted (if necessary) before any transformations are applied. If transformations are applied first, and the Frame's definition adjusted at a later time, the transformations will likely cause unexpected results (since the transformations originally performed were about a different axis definition than that about which transformations performed after the Frames definition changed occur). The necessary order is 1) define frame location and orientation, 2) assign part to frame, 3) perform transformations relative to the frame.

A Part can be assigned to only one Frame at a time. The Part will always be transformed by the Frame's transformation. A Part is not affected by a Frame's definition (other than transformations will be in reference to the definition). A Part's mirror symmetry operation (which can be thought of as a scaling transformation) is always about the Frame to which the Part is assigned.

The Tools (Cursor, Line, Plane, etc.) are always shown in reference to the selected Frame and are thus also transformed by the selected Frame's transformations.

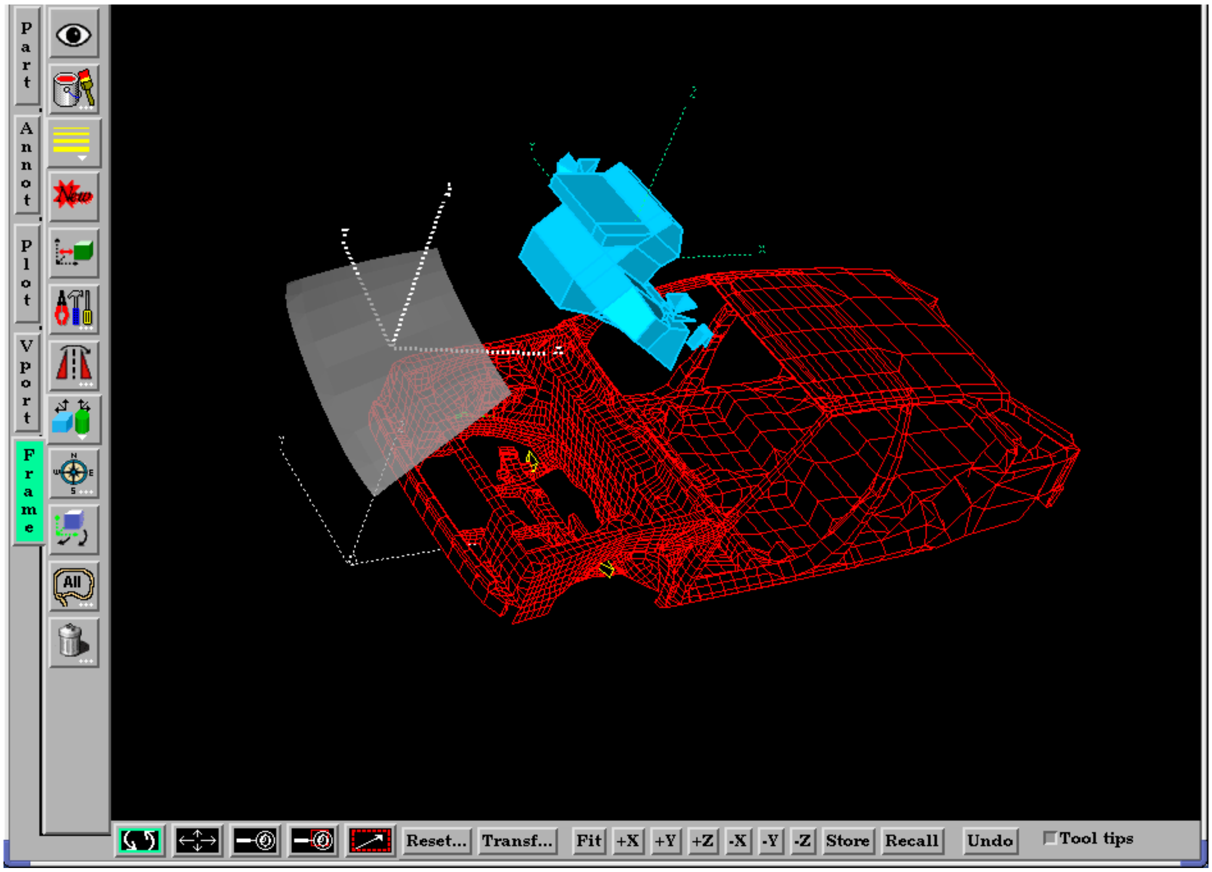
There are two transformation alternatives in Frame Mode: Frame Transform (the default) and Frame Definition. As pointed out earlier, a frame should first be defined (if necessary) before it is transformed.

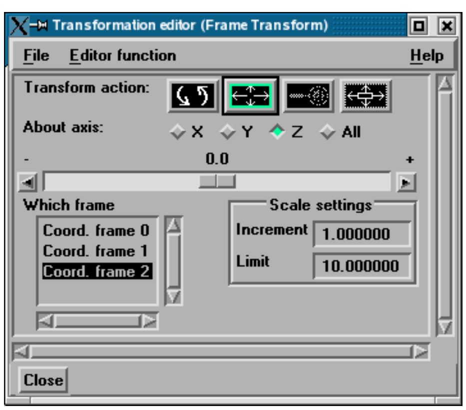
In Frame Mode the axis triads for all Frames will be visible in the Graphics Window. Invisible Frames will be shown with a dotted frame axis. Selected Frames are shown in green. A Frame may be selected by clicking on its axis triad or by selecting its description from the Frame List in the Transformation Editor dialog (which is opened by clicking the Transf Edit... button in the Transformation Control Area).

By default, Frame mode is not available unless it has been enabled under Edit > Preferences... General User Interface - Frame Mode Allowed.

For further discussion concerning the transformation of Frames:

(see Section 9.3, Frame Transform and Section 9.2, Frame Definition)

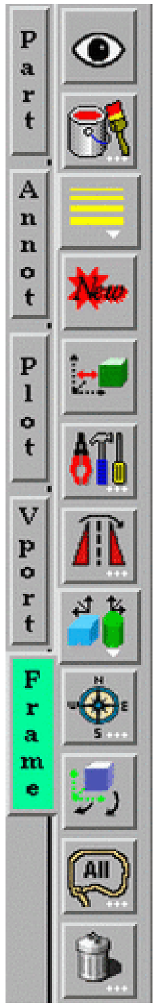




Three Frames exist in this example.

Figure 8-56
Frame Mode - Frame Example

When Frame Mode is selected, the Mode Icon Bar appears as follows.:



Frame Axis Triad Visibility Toggle Icon

Color Icon

Frame Line Width Icon

New Frame Icon

Part Assignment Icon

Axis Triad Attributes Icon

Computational Symmetry Attributes Icon

Coordinate System Pulldown Icon

Frame Location Attributes Icon

Transform/Definition Icon

Frame Select All Icon

Delete Icon

Figure 8-57
Mode Selection Area - Frame Selected

Frame Axis Triad Visibility Toggle Icon

Determines the visibility of the axis triad(s) of selected Frame(s). Invisible Frames are drawn in dotted lines while in Frame Mode. Default is Off.

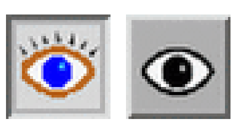


Figure 8-58
Frame Mode - Frame Axis Triad Visibility Toggle Icon

Access: Frame Mode : Frame Axis Triad Visibility Toggle Icon

Color Icon

Opens the Color Selector dialog for the specification of the color you wish to assign to a selected Frame’s axis triad A selected Frame will always be shown in the selection color while in Frame Mode.

Figure 8-59
Frame Mode - Color Icon

Access: Frame Mode : Color Icon

Frame Line Width Pulldown Icon

Opens a pulldown menu for the specification of the width for Frame axis triad lines for the selected Frame(s).

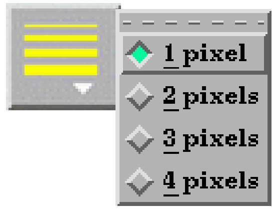


Figure 8-60
Frame Mode - Frame Line Width Pulldown Icon

Access: Frame Mode : Frame Line Width Icon

New Frame Icon

Creates a new Frame to which you can assign Parts. Be aware that each time you make a copy of a Part EnSight creates a new Frame and assigns the copy to the new Frame. If Parts are selected in the Main Parts List, the new Frame's origin will be positioned at the center of the selected Parts.

Figure 8-61
Frame Mode - Create New Frame Icon

Access: Frame Mode : Create New Frame Icon

Part Assignment Icon

Clicking this icon reassigns Part(s) selected in the Main Parts List to the currently selected Frame. (An alternative method for reassigning Parts is to edit the Ref. Frame field in the General Attributes section of the Feature Detail Editor.)

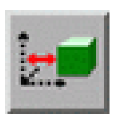


Figure 8-62
Frame Mode - Part Assignment Icon

Access: Frame Mode : Part Assignment Icon

Axis Triad Attributes Icon

Opens the Frame Axis Attributes dialog for the specification of axis triad line length and labels for the selected Frame(s).

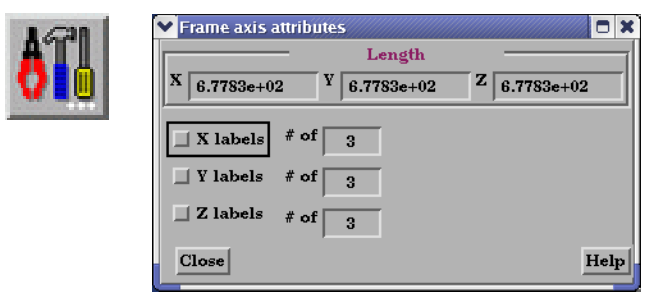


Figure 8-63
Frame Mode - Axis Triad Attributes Icon

X Y Z

These fields allow you to specify the desired length, in model coordinates, of each of the three axes of the selected Frame's axis triad.

X Y Z Labels Toggles
: Toggles on/off the display of Labels on the respective line of a selected Frame's axis triad. Labels show distance along each axis.

X Y Z # of
: These fields specify the number of Labels which will appear on the respective axis.

Access: Frame Mode : Axis Triad Attributes Icon

Computational Symmetry Attributes Icon
: Opens the Frame Computational Symmetry Attributes dialog for the specification of the type of periodic conditions which will be applied to all assigned **Model** Parts of the selected Frame. ***(Note, computational symmetry does NOT work on created parts.)***

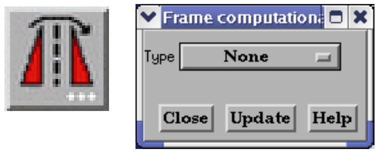
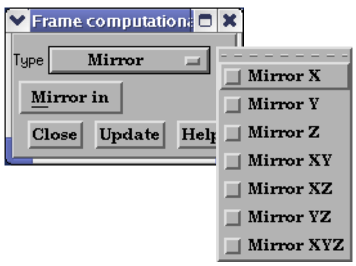


Figure 8-64
Frame Mode - Computational Symmetry Attributes Icon

(see How To Set Symmetry)

Type
: Opens a pop-up menu for the selection of whether you wish the selected Frame to have no periodicity (None as shown above) or to be mirror, rotational, or translational periodic.

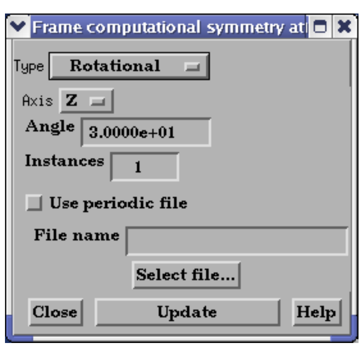
Mirror



Mirror In
: Specification of the type of mirror periodicity.

Mirror X	face-sharing quadrant on other side of the Y-Z plane
Mirror Y	face-sharing quadrant on other side of the X-Z plane
Mirror Z	face-sharing quadrant on other side of the X-Y plane
Mirror XY	diagonally opposite quadrant on same side of the X-Y plane
Mirror XZ	diagonally opposite quadrant on same side of the X-Z plane
Mirror YZ	diagonally opposite quadrant on same side of the Y-Z plane
Mirror XYZ	quadrant diagonally opposite through origin

Rotational



Axis
: The Frame axis about which to rotate.

Angle
: This field specifies the rotational angle (in degrees) about the selected Frame's z-axis for rotational periodicity.

Instances
: This field specifies the number of periodic instances for rotational periodicity.

Use Periodic — If toggled On, the periodic match file specified in File Name is used for rotational symmetry.

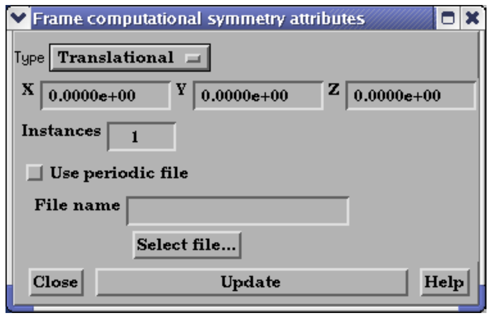
File Name — This field specifies the name of the periodic match file you wish to use.

Select File... — Opens the File Selection dialog for the selection of a periodic match file.

(see Section 11.9, Periodic Matchfile Format)

Update — Changes made in the dialog will not be applied until this button is clicked.

Translational



X Y Z — These fields specify the translational offset in reference to the selected Frame's orientation.

Instances — This field specifies the number of periodic instances for translational periodicity.

Use Periodic — If toggled On, the periodic match file specified in File Name is used for translational symmetry.

File Name — This field specifies the name of the periodic match file you wish to use.

Select File... — Opens the File Selection dialog for the selection of a periodic match file.

(see Section 11.9, Periodic Matchfile Format)

Update — Changes made in the dialog will not be applied until this button is clicked.

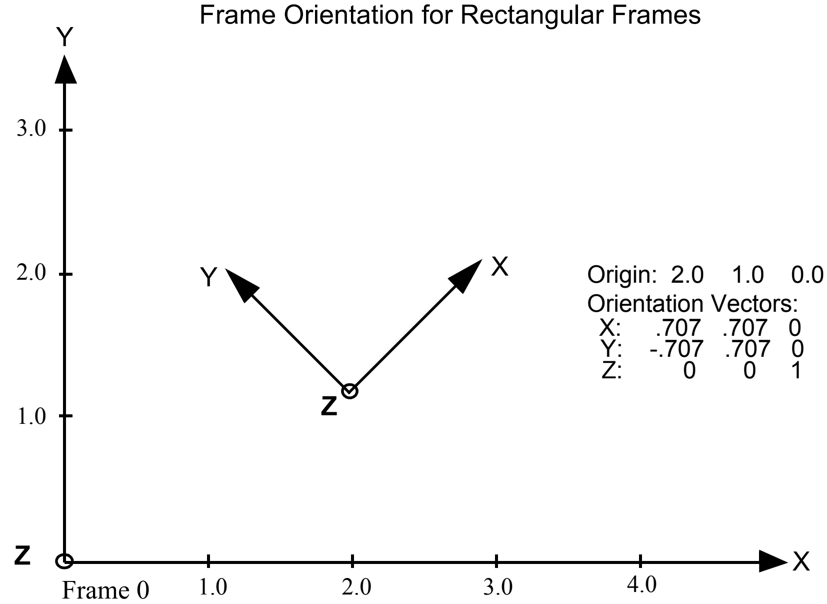
Access: Frame Mode : Computational Symmetry Attributes Icon

Coordinate System Pull-down Icon — Opens a pulldown menu for the selection of the type of coordinate system (rectangular, cylindrical, spherical) you wish to use for a selected Frame. All three are defined in reference to Frame 0, which is rectangular. Note that each frame's orientation vectors (which describe its orientation to Frame 0) are rectangular (as is their on-screen representation) no matter what the frame's coordinate system type. However, functions that access the frame will behave different depending on the frame's coordinate system type.

Figure 8-65
Frame Mode - Coordinate System Pulldown Icon

Rectangular The Figure below shows a rectangular frame. The origin is in reference to the Frame 0 origin, while the orientation is in reference to Frame 0's orientation.

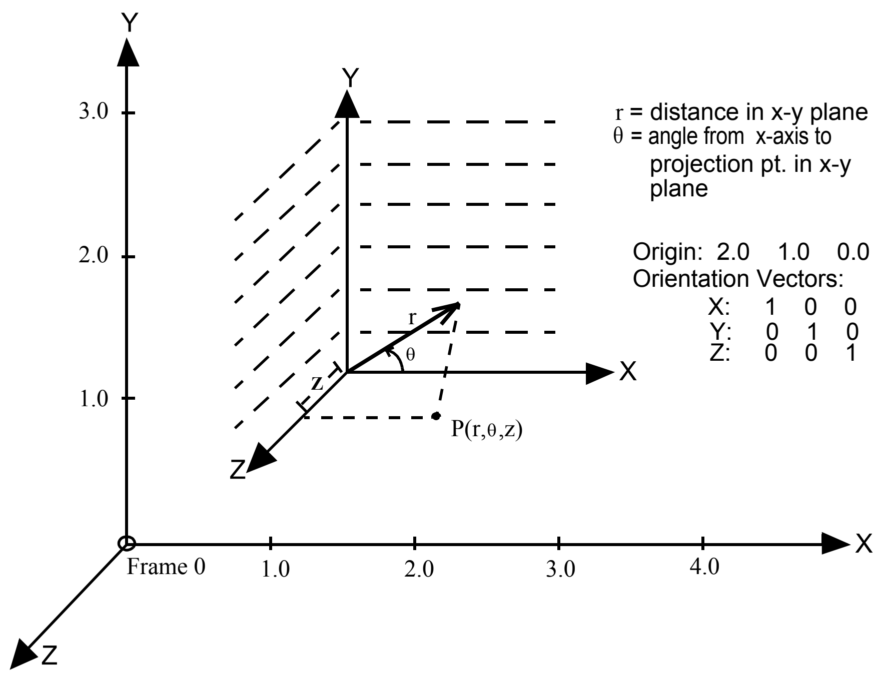
Frame Orientation for Rectangular Frames



Cylindrical The figure below shows a cylindrical frame. The origin is in reference to the Frame 0 origin, while the orientation is in reference to Frame 0's orientation. Any function which accesses a cylindrical frame will do so in cylindrical coordinates:

r The distance from the origin to projection point in the X–Y plane.
Θ The angle from the X-axis to the projection point in the X–Y plane.
Z The Z-coordinate

Frame Orientation and Coordinate System Key for Cylindrical Frames



Spherical

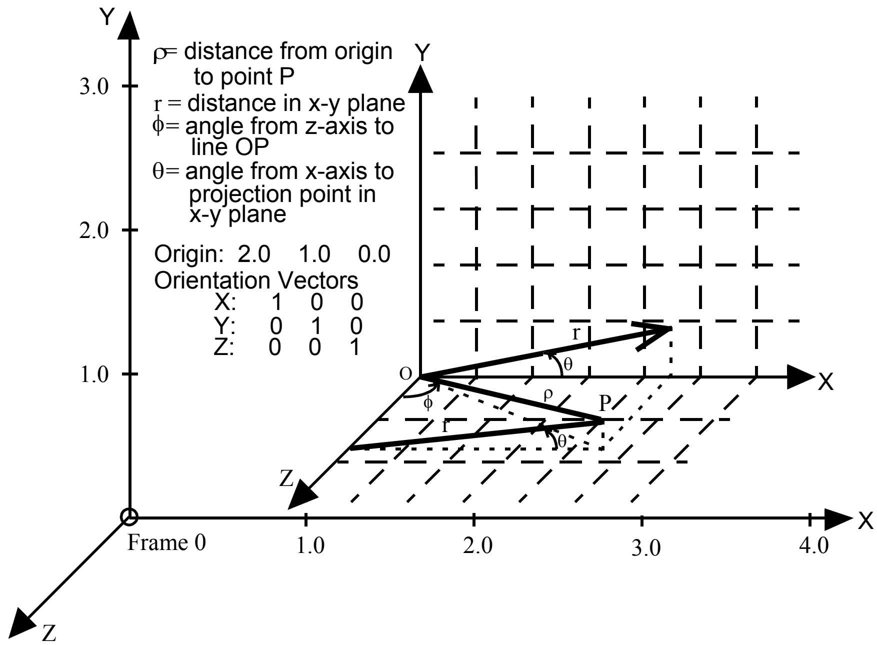
The figure below shows a spherical frame. The origin is in reference to the Frame 0 origin, while the orientation is in reference to Frame 0's orientation. Any function which accesses a spherical frame will do so in spherical coordinates:

ρ The distance from the origin to the point in question.

Φ The angle measured from the Z-axis towards the projection point in the X–Z plane.

Θ The angle from the X-axis to the projection point in the X–Y plane.

Frame Orientation and Coordinate System Key for Spherical Frames



Access: Frame Mode : Coordinate System Pull-down Icon

Frame Location

Opens the Transformations Editor dialog to permit precise definition of the selected Frame(s).

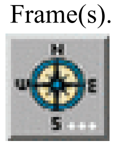


Figure 8-66
Frame Mode - Frame Location Attributes Icon

(see Section 9.3, Frame Transform)

Access: Frame Mode : Frame Location Attributes Icon

Transform/Definition Pulldown Icon

Opens a pop-up menu for selection of desired method of Frame transformation.

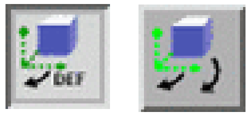


Figure 8-67
Frame Mode - Define / Transform Pulldown Icon

Transform - Transformations will cause the Parts assigned to the selected Frame(s) to be transformed as well as the selected Frame's axis triad. Translations will move the Frames' axis triad(s) and the assigned Parts. Rotations of Parts will take place about the selected Frame(s) axis origin.

Definition - User interaction in the Graphics Window or Transformation Editor will modify the selected Frame(s) origin location and/or axis orientation.

Access: Frame Mode : Transform/Definition Pull-down Icon

(see Section 9.3, Frame Transform and Section 9.2, Frame Definition)

Select All

Selects all frames.

Figure 8-68
Frame Mode - Select All Icon

Delete Icon

Deletes the selected Frame(s). (Will be prohibited if currently used by a part).

Figure 8-69
Frame Mode - Delete Icon

Access: Frame Mode : Delete Icon

8.6 Quick Desktop Buttons

Quick Desktop Buttons are always available regardless of the Mode (Part, Annot, Plot, Vport, or Frame) above the Graphics Window.

.

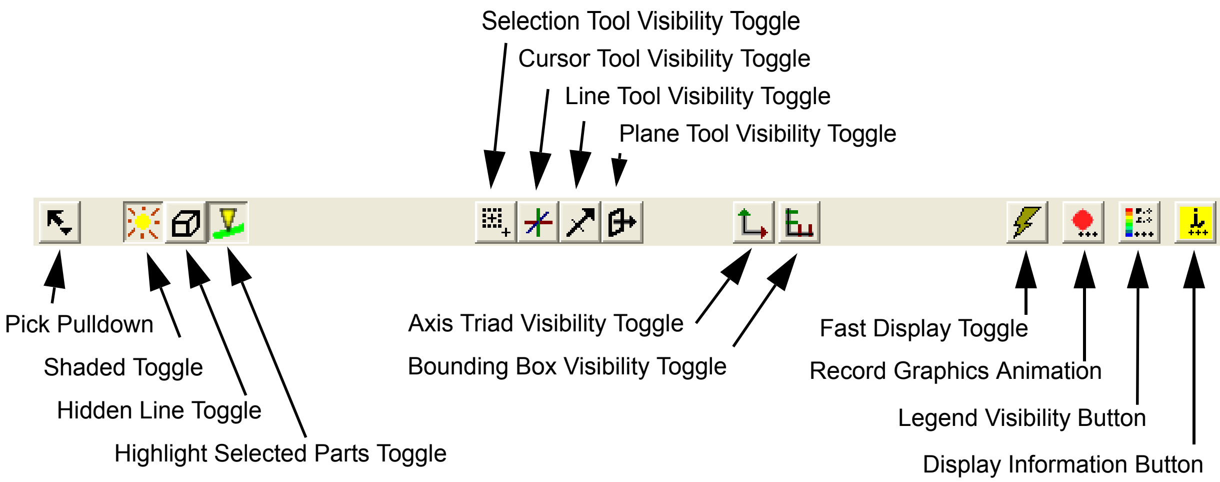


Figure 8-70
Mode Selection Area - View Selected

Pick Pull-down Icon Opens a pull-down menu for the specification of the desired type of Pick operation. The actual Pick operation is normally assigned to the "P" key on the keyboard, unless it has been reassigned under Main Menu: Edit > Preferences... Mouse and Keyboard...

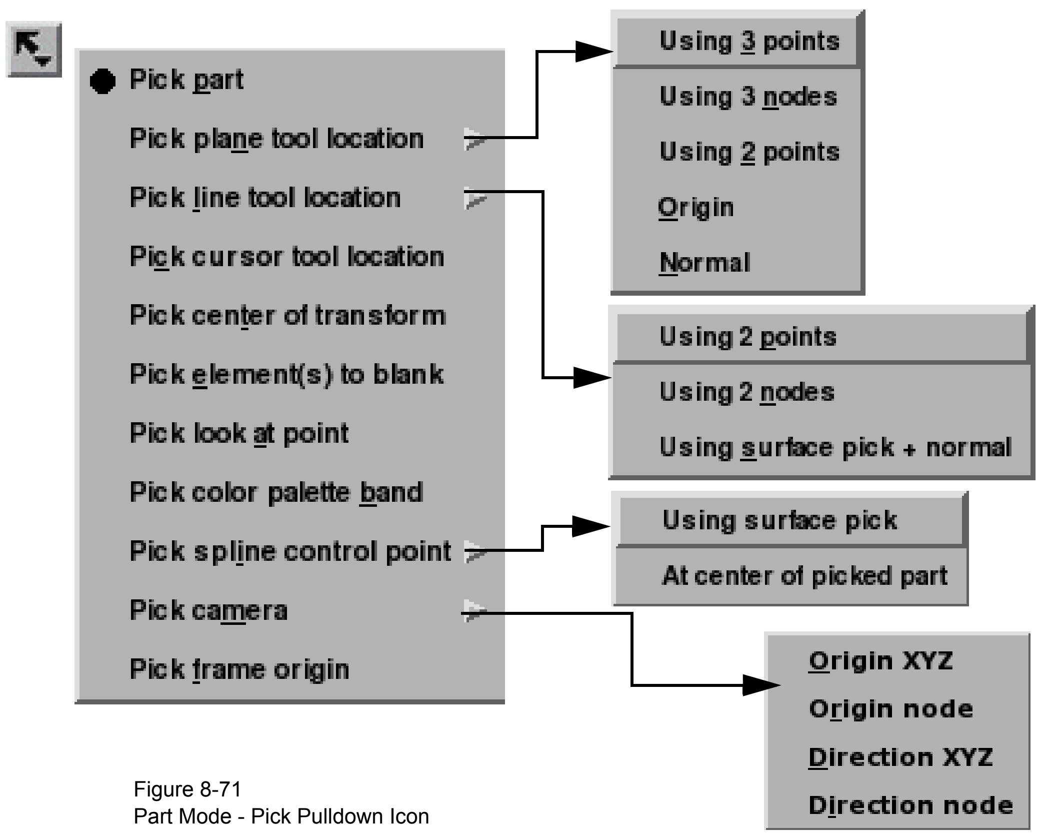


Figure 8-71
Part Mode - Pick Pulldown Icon

Pick Part	When the Pick operation is performed (by default, pressing the “P” key), the Part directly under the mouse cursor is selected. To select multiple Parts, hold down the Control Key during the Pick operation. It is usually helpful to open and use the Selected Parts Window while Picking Parts. This is done from Main Menu: View > Show Selected Part(s)...
Pick Plane Tool Location - Using 3 points	When the Pick operation is performed (by default, pressing the “P” key), the Plane Tool will be positioned at the Picked points. Three points must be Picked to position the Plane Tool.
Pick Plane Tool Location - Using 3 nodes	When the Pick operation is performed (by default, pressing the “P” key), the Plane Tool will be positioned at the nodes nearest the pick points. Three nodes must be Picked to position the Plane Tool. The plane will continue to be tied to these three nodes.
Pick Plane Tool Location - Using 2 points	When the Pick operation is performed (by default, pressing the “P” key), you can click and drag the mouse to define a line. The Plane Tool will be positioned parallel to your current viewing angle through the defined line. Consider using this option together with the F5, F6, F7, and F8 keys which will transform the view to a standard orientation. (see Section 9.1, Global Transform)
Pick Plane Tool Location - Using Origin	When the Pick operation is performed (by default, pressing the “P” key), the location of origin of the plane tool is chosen. Note that the orientation of the plane tool is unchanged with this option.
Pick Plane Tool Location - Using Normal	When the Pick operation is performed (by default, pressing the “P” key), the origin of the plane tool remains unchanged, but the normal goes from the origin to the picked point.
Pick Line Tool Location - Using 2 points	When the Pick operation is performed (by default, pressing the “P” key), the ends of the Line Tool will be placed at the Picked points. Two points must be picked to position the Line Tool.
Pick Line Tool Location - Using 2 nodes	When the Pick operation is performed (by default, pressing the “P” key), the ends of the Line Tool will be placed at the nodes closest to the picked points. Two nodes must be Picked to position the Line Tool. The line tool will continue to be tied to these two nodes.
Pick Cursor Tool Location	When the Pick operation is performed (by default, pressing the “P” key), the Cursor Tool will be positioned at the Picked point.
Pick Center of Transformation	When the Pick operation is performed (by default, pressing the “P” key), the center of global transformation is positioned at the Picked point.
Pick Elements to blank	When the Pick operation is performed (by default, pressing the “P” key), the Element that is chosen is visually removed from the graphics screen. The element still remains in the model and is used for all calculations, just is not rendered on the graphics screen. Use the Element Blanking Button in Part Mode to enable / disable and restore visibility. Element Blanking Button 
Pick Look At Point	When the Pick operation is performed (by default, pressing the “P” key), the Look At Point is positioned at the Picked point. The Look From Point is also adjusted to preserve the distance (between the two Points) and vector that existed prior to the Pick operation. (see Section 9.7, Look At/Look From)
Pick color palette band	When the Pick operation is performed (by default, pressing the "P" key) on geometry colored by a variable or a color palette visible in the graphics window, a color band will appear. (see Section 6.2, Edit Menu Functions))
Pick spline control point	When the Pick operation is performed (by default, pressing the "P" key) on visible geometry a control point will be added to the currently selected spline at the insertion point. If no spline is currently selected or one does not exist a new spline will be created.

(see Section 6.5, Tools Menu Functions)

Pick camera Origin XYZ	When the Pick operation is performed (by default, pressing the "P" key) on visible geometry the selected camera origin will be updated to the picked position.
Pick camera Origin node	When the Pick operation is performed (by default, pressing the "P" key) on visible geometry the selected camera origin will be updated to the picked node id and it's origin property will be updated to contrain itself to the node's position.
Pick camera Direction XYZ	When the Pick operation is performed (by default, pressing the "P" key) on visible geometry the selected camera view direction will be udpated to point to the picked coordinate location and it's direction property will be updated to contrain itself to point to the given location.
Pick camera Direction node	When the Pick operation is performed (by default, pressing the "P" key) on visible geometry the selected camera view direction will be udpated to point to the picked node id location and it's direction property will be updated to contrain itself to the node's position.

Access: Part Mode : Pick Pull-down Icon

Global Shaded Toggle Icon

Toggles on/off global Shaded (default is off) which displays all Parts in a more realistic manner by making hidden surfaces invisible while shading visible surfaces according to specified lighting parameters. Performs the same function as Main Menu > View > Shaded toggle button and the Desktop > Shaded button.

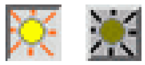


Figure 8-72
View Mode - Shaded Toggle Icon ON / OFF

When toggled-off, all visible Parts are shown as line drawings. Shaded may be turned off for individual Parts using the Shaded toggle in the Parts Mode Icon Bar or the Feature Detail Editor for each type of Part. It can also be turned off for a Particular viewport in the Viewport Special Attributes Icon under VPort Mode.

Shaded require more time to redraw than a line-mode display (the default), so you may wish to first set up the Graphics Window as you want it, then turn on Shaded to see the final result. It is possible to improve graphics performance when Shaded is on by also toggling on Static Lighting (Main Menu > View > Static Lighting). To shade surfaces, a Part's representation on the Client must include surfaces - (2D elements). Any 1D elements of Parts displayed with Shaded on will continue to be drawn as lines. Lighting parameters for brightness and reflectivity are specified independently in the Feature Detail Editor for each type of Part.

Access: View Mode Icon Bar: Shaded Toggle Icon
or: EnSight dialog > View > Shaded
or: Desktop > Shaded

(see Section 6.4, View Menu Functions and How To Set Drawing Style)

Troubleshooting Hidden Surfaces

Problem	Probable Causes	Solutions
Graphics Window shows line drawing after toggling on Shaded.	Shaded is toggled off for some or all individual Parts.	Toggle Shaded on for individual Parts with the Shaded Icon in Part Mode or in the Feature Detail Editor dialog.
	There are no surfaces to shade—all Parts have only lines.	If Parts are currently in Feature Angle representation, change the representation. If model only has lines, you can not display shaded images.
	Element Visibility has been toggled off for some or all Parts.	Toggle Element Visibility on for individual Parts in the Feature Detail Editor dialog.

Global Hidden Line Toggle Icon

Toggles on/off global Hidden Line (default is off) which simplifies a line-drawing display by making hidden lines—lines behind surfaces—invisible while continuing to display other lines. Performs the same function as Main Menu > View > Hidden Line Toggle and the Desktop > Hidden Line button.

Figure 8-73
View Mode - Global Hidden Line Toggle Icon ON / OFF

Hidden Line can be combined with Shaded to display both shaded surfaces and the edges of the visible surface elements. Hidden Line applies to all Parts displayed in the Graphics Window but it can be toggled-on/off for individual Parts using the Feature Detail Editor or the Part Mode: Hidden Line Toggle button.

To have lines hidden behind surfaces, you must have surfaces (2D elements). If the representation of the in-front Parts consists of 1D elements, the display is the same whether or not you have Hidden Lines mode toggled-on.

During interactive transformations, the display reverts to displaying all lines. When you release the mouse button, the Main View display automatically resumes Hidden Line mode (assuming it is toggled on at that time).

The Hidden line option will not be active during playback of flipbook objects animations.

Hidden Line Overlay

If you toggle Hidden Line on while Shaded is already on, the lines overlay the surfaces. EnSight will prompt you to specify a color for the displayed lines (you do not want to use the same color as the surfaces since they then will be indistinguishable from the surfaces). The default is the Part-color of each Part, which may be appropriate if the surfaces are colored by a color palette instead of their Part-color.

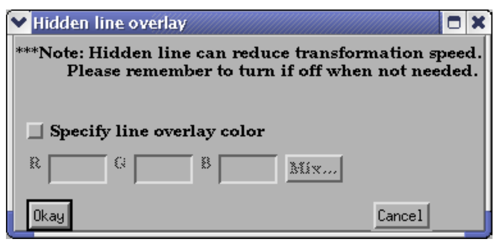


Figure 6-74
Hidden Line Overlay dialog

Specify Line Overlay Toggle

Toggle-on if you want to specify an overlay color. If off, the overlay line color will be the same as the Part color.

R, G, B

The red, green, and blue components of the hidden line overlay. These fields will not be accessible unless the Specify Overlay option is on.

Mix...

Click to interactively specify the constant color used for the hidden line overlay using the Color Selector dialog. (See Section 8.1, Color Selector) and How To Change Color)

Access: View Mode Icon Bar: Global Hidden Line Toggle Icon
or: Main Menu > View > Hidden Line
or: Desktop > Hidden Line

(See How To Set Attributes)

Highlight Selected .Parts Toggle

Toggles on/off the hightlighting in the graphics window of selected parts. Parts selected in the part list are indicated in the graphics window.

Figure 8-75
View Mode - Hightlight ON / OFF

Selection Tool Visibility Toggle

Toggles on/off global visibility of the Selection Tool (default is off)

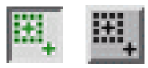


Figure 8-76
View Mode - Global Selection Tool Visibility Toggle Icon ON / OFF

: Main Menu > Tools > Region Selector

Cursor Tool Visibility Toggle

Toggles on/off global visibility of the Cursor Tool (default is off)

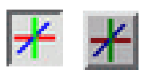


Figure 8-77
View Mode - Global Cursor Tool Visibility Toggle Icon ON / OFF

: Main Menu > Tools > Cursor

Line Tool Visibility Toggle

Toggles on/off global visibility of the Line Tool (default is off)

Figure 8-78
View Mode - Global Line Tool Visibility Toggle Icon ON / OFF

: Main Menu > Tools > Line

Plane Tool Visibility Toggle

Toggles on/off global visibility of the Plane Tool (default is off)

Figure 8-79
View Mode - Global Plane Tool Visibility Toggle Icon ON / OFF

: Main Menu > Tools > Plane

Global Axis Triad

Toggles on/off the visibility (default is off) of the Global Axis triad.

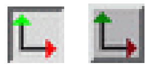


Figure 8-80
View Mode - Global Axis Triad Visibility Toggle Icon

The Global Axis triad shows the point and axes around which Global rotations occur.

: Main Menu > View > Axis Triad Visibility

Bounding Box Visibility Toggle Icon

This toggles on and off a 2D or 3D bounding box grid describing the upper and lower coordinates in each of the X-, Y-, and Z-directions.

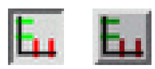


Figure 8-81
View Mode - Global Bounding Box Visibility Toggle Icon

The Global Axis triad shows the point and axes around which Global rotations occur.

: Main Menu > View > Bounds Visibility

Fast Display Toggle Icon

This toggles on and off the Fast Display feature. This feature reduces a model to a simple representation prior to doing a transformation such as rotate or translate in order to speed up the rendering. Simple representations include Box, Points, Reduced Polygon, Invisible, and Sparse model (if running in immediate mode) which can be chosen from the Fast Display Representation icon in Part Mode.

Fast Display Representation Icon

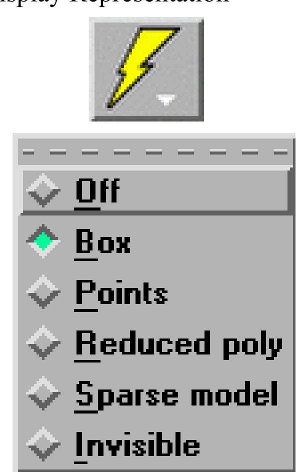


Figure 8-82
View Mode - Global Fast Display Toggle Icon ON / OFF and button and pulldown

Fast Display allows faster model translation or rotation for large models.

: Main Menu > View > Fast Display

Record Animation

Button

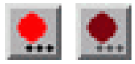


Figure 8-83
View Mode - Global Record Animation Toggle Icon ON / OFF

The Record Active Animation Toggle will only turn ON if you have an active animation (flipbook, or streaming transient data via the clock icon, or animated particle traces, or spin mode where you press F4 and drag the cursor to spin the model, or clip in auto or auto cycle interactive mode, or an isosurface in auto or auto cycle interactive mode). This button does not record keyframing, which has it's own button in the Keyframe Quick Interaction Area.

Legend Visibility Toggle Icon

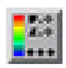


Figure 8-84
View Mode - Legend Visibility Button

The Legend Visibility Button will bring up the Legend Visibility Dialog which allows control of legend visibility for all active variables. The Global Legend Visibility Flag must be ON to see the legend, see Main Menu > View > Legend.

Display Information Icon

Click on this icon to see a message related to the most recent EnSight task that you've completed. This icon is red (error message), yellow (caution message) or green (OK or no message) depending upon the result of your most recent EnSight task.

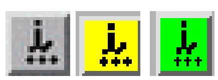


Figure 8-85
View Mode - Global Display Information Toggle Icon States

9 Transformation Control

Included in this chapter:

General Description

An essential feature of postprocessing is the reorientation of the visualized model in order to see it from a number of different vantage points. Basic transformations include *rotating* (about an axis or axis origin point), *translating* (up, down, left, right), and *zooming* (moving the model toward or away from you). When EnSight reads in a geometry file, it assigns all model parts to the same Frame of reference: Frame 0. Frame 0 corresponds to the model coordinate system (defined when the model was created).

Two methods exist to transform a scene. In Global transform mode, the geometry is transformed. In Camera transform mode, the scene does not move but instead a camera is moved. A viewport can either use a global transform mode or can be viewed through a camera.

Using the Frame Mode, it is possible to create additional frames and reassign parts to them. In fact, when you copy a part, a new Frame is automatically created and the part copy is assigned to the new Frame. (See Section 8.6 Frame Mode for further discussion).

Just after all parts of your model have been read in, EnSight centers the model in the Graphics Window by placing the geometric center of the model at the *Look At Point* which is always located in the center of the Graphics Window. Initially - before any Global translations are made -the origin for the *Global Axis* is located at the Look At Point.

There are nine Editor Functions available within the Transformation Editor, Global Transform, Camera, Frame, Tools, Center of Transform, Z-Clip, Look At/ Look From, Copy Transformation State, and Paste Transformation State. (The Transformation Editor dialog is opened by clicking the Transf Edit... button)

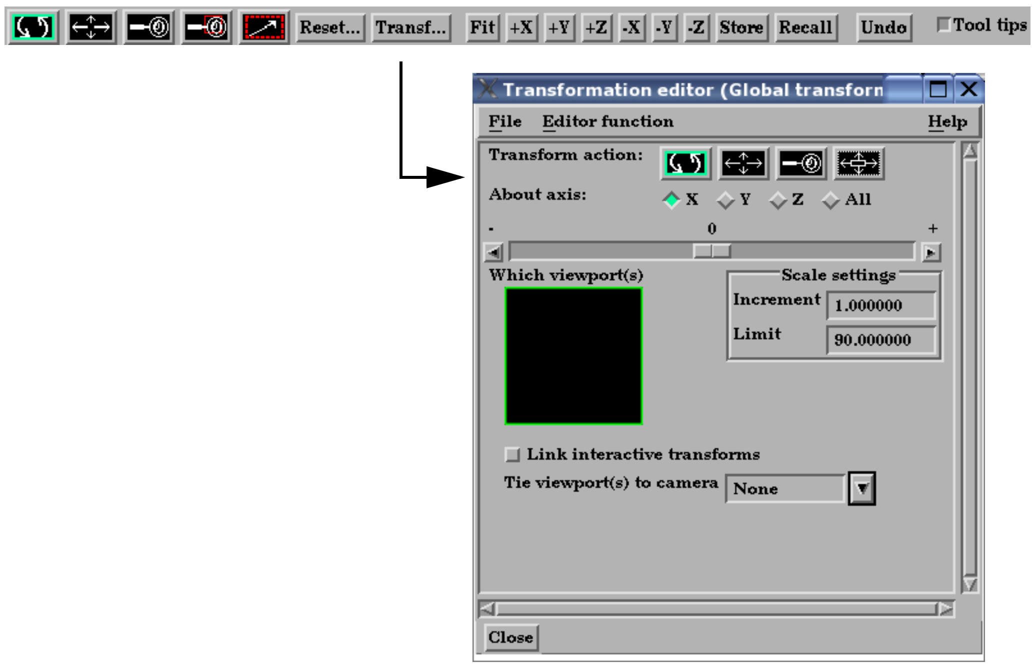


Figure 9-1
Transf Edit... button and Transformation Editor dialog

Transformations performed within the Editor affect the selected viewports and/or frames. The transforms from one viewport can be copied to another by selecting the viewport to be copied, selecting *Copy Transformation State*, selecting the viewport(s) to be modified, and selecting *Paste Transformation State*.

File button Pull-down Menu

File > Save View — This opens the Save View dialog which allows you to save in a file the view (orientation) of the model you have created in the Graphics Window and any Viewports by selecting Save View and then entering the name of the file.

File > Restore View — Opens the Restore View dialog which allows you to specify the name of a file in which you previously stored a view. Clicking Okay in this dialog restores the view only in the selected Viewports.

9.1 Global Transform

Transformations you make while in Part Mode (rotations, translations, zoom, scale) are performed globally. Global transformations affect the *entire* model as a unit and move all Frames, parts, and *visible* tools relative to the Global Axis. If the viewport is being viewed through a camera, the transformations move the camera. If the viewport is not viewed through a camera, the geometry is transformed. You can make the Global Axis triad (which pinpoints the Global Axis Origin) visible by selecting Axis Visibility > Axis - Global from View in the Main Menu or by clicking the Global Axis Visibility Toggle Icon on the desktop.

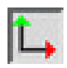


Figure 9-2
Global Axis Visibility Toggle Icon and Global Axis triad

You can also show the global frame orientation by toggling it on from Desktop > Axis.

Most Global transformations you will make will be done interactively. Interactive Transformations normally affect only the single, selected viewport (the one which the mouse pointer is in when you click the left mouse button). The exception to this is if when you toggle on *Link Interactive Transforms*, causing the selected viewports in the Transformation Editor dialog to all transform together. You choose the type of transformation you wish to perform from among the Transformation Control Icons.

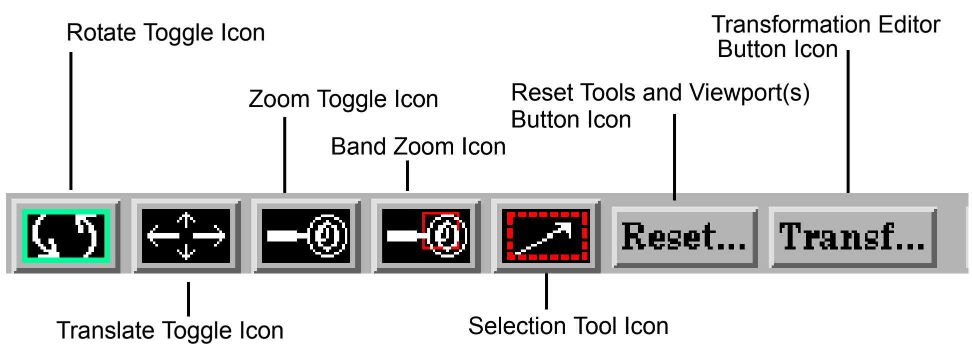


Figure 9-3
Transformation Control Area in View or Part Mode

Rotate Toggle

Interactive Rotation

When this toggle is on, clicking the left mouse button and dragging horizontally will rotate the scene (including any tools that are visible) about the Global Y axis.

Clicking the left mouse button and dragging vertically will rotate the scene (including any tools that are visible) about the Global X axis.

Holding the Control Key down and then clicking the left mouse button and dragging will rotate the scene (including any tools that are visible) about the Global Z Axis.

Rotation Using Function Keys

Pressing the F1, F2, or F3 function keys will rotate the scene 45 degrees about the X, Y, or Z axis respectively. Holding the Control Key down while pressing these keys will rotate the scene by -45 degrees. The mouse must be located in the graphics window for these keys to work.

Precise Rotation

When the Transformation Editor is open under Global Transform and the Rotate toggle is selected, the dialog will be configured to permit precise Rotation.

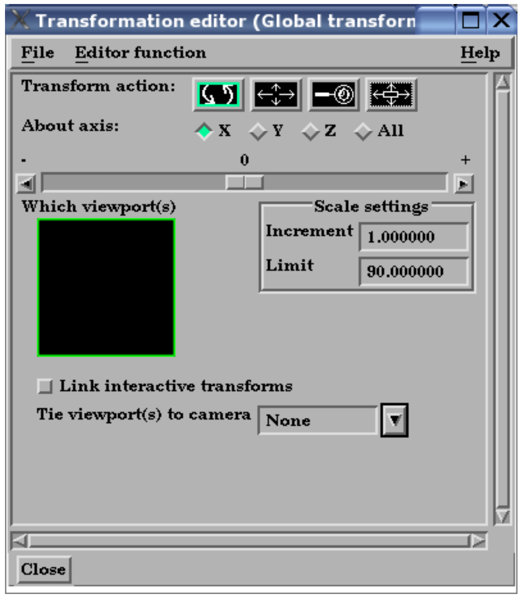


Figure 9-4
Transformation Editor for Exact Global Rotation

You may rotate the entire scene or camera (including any tools that are visible) precisely about the X, Y, Z, or All axes by clicking on the appropriate axis of rotation toggle and:

entering the desired rotation in (+ or -) degrees in the Increment field and pressing Return,

clicking the stepper buttons at each end of the slider bar (each click will rotate the model by the number of degrees specified in the Increment field), or

dragging the slider in the positive or negative direction to the desired number of degrees you wish to rotate the model (the Limit Field specifies the maximum number of degrees of rotation performed when the slider is pulled to either end of the slider bar).

Translate Toggle

Interactive Translation

When this toggle is on, you can transform objects interactively in the Global X-Y plane (or by holding down the Control key, in Z). Clicking the left mouse button and dragging will translate the scene (including any tools that are visible) up, down, left or right (or forward or backward).

Precise Translation

When the Transformation Editor is open under Global Transform and the Translate toggle is selected, the dialog will be configured to permit precise Translation.

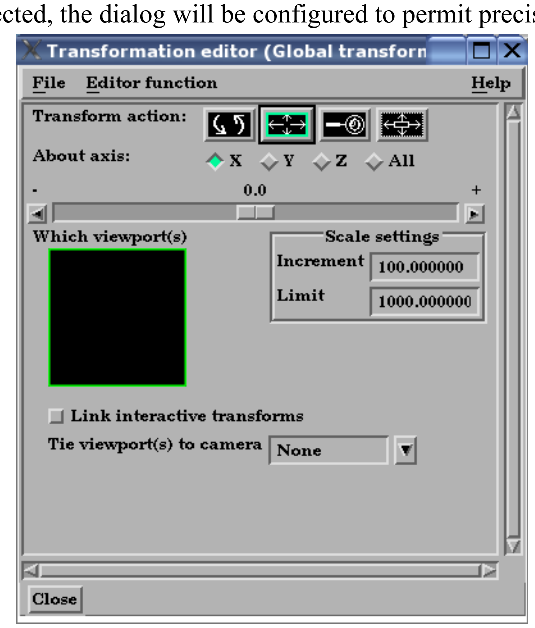


Figure 9-5
Transformation Editor for Exact Global Translation

You may translate the entire scene or camera (including any tools that are visible) precisely along the X, Y, Z, or All axes by clicking on the appropriate direction toggle and:

1. entering the desired translation in (+ or -) model coordinate units in the Increment field and pressing Return, then
2. clicking the stepper buttons at each end of the slider bar (each click will translate the model by the number of model coordinate units specified in the Increment field), or
3. dragging the slider in the positive or negative direction to the desired number of model coordinate units you wish to translate the model and then releasing the slider (the Limit Field specifies the maximum number of model coordinate units that the model is translated when the slider is pulled to either end of the slider bar).

Zoom Toggle

A Zoom transform while in Global Tranform Mode is really an adjustment of the Look From Point, which you might also think of as the Camera Position. When this toggle is on, clicking and dragging to the left or down will zoom-in, that is it will move the Look From Point closer to the Look At Point. Clicking and dragging to the right or up will zoom-out, that is it will move the Look From Point farther away from the Look At Point. If you hold down the Control key while interactively zooming, you will "pan", i.e. move both the Look At and Look From Points in the direction of the mouse movement.
(see Section 9.7, Look At/Look From)

A Zoom transform while in Camera Mode is a movement of the camera in the camera Z direction.

As you Zoom in or out, be aware that you may clip the model with the Front or Back Z-Clip planes since they move in relationship to the Look From Point, always maintaining the distance from the Look From Point specified in the Transformation Editor dialog: Editor Function > Z-Clip.
(see Section 9.6, Z-Clip)

Precise Zooming

When the Transformation Editor is open under Global Transform and the Zoom toggle is selected, the dialog will be configured to permit precise Zoom.

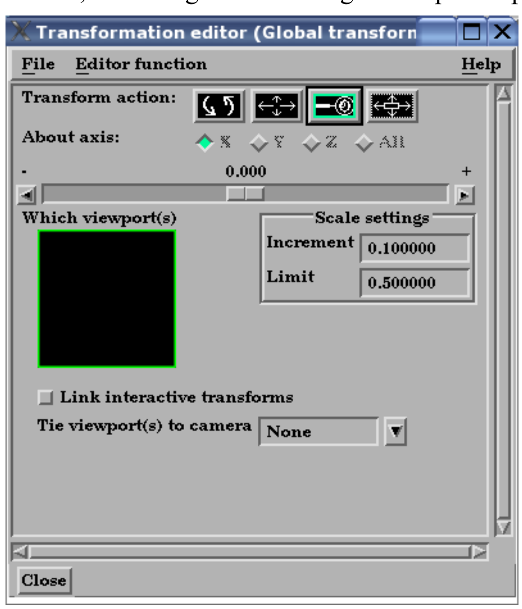


Figure 9-6
Transformation Editor for Exact Global Zoom

You may precisely adjust the position of the Look From Point (with respect to the Look At Point) or move the camera in the camera Z-direction by:

1. entering in the Increment Field the desired modification (+ or -) in the

distance between the two Points (a value of .5 will increase the distance to be equal to 1.5 the current distance, a value of 1.0 will double the current distance), then

2. clicking the stepper buttons at each end of the slider bar (each click will increase or decrease the distance between the two Points by the factor specified in the Increment field), or
3. dragging the slider in the positive or negative direction to the desired modification factor and then releasing the slider (the Limit Field specifies the maximum modification factor for the distance between the two Points when the slider is pulled to either end of the slider bar).

Band Zoom Button

You specify the area of interest by clicking and dragging the white rectangle (rubber band) around the area you wish to zoom in on. Immediately after you perform the Band Zoom operation however, EnSight will switch to the regular Zoom Transformation. So, each time you click on the Band Zoom button, EnSight allows you to perform one Band Zoom operation and subsequent clicking/dragging actions you make in the Graphics Window perform regular Zoom transformations.

Band Zoom combines the functionality of a zoom-in transform as described above with a panning operation. The effect of performing a Band Zoom is that the area of interest that you specify will be centered in and will fill the selected viewport. EnSight adjusts the transformation center to be in the center of the area you specified.

The Transformation Editor is inactive for the Band Zoom Operation.

Scale Toggle

Interactive modifications to scale are not permitted. When the Transformation Editor is open under Global Transform and the Scale toggle is selected, the dialog will be configured to permit precise adjustments to the scale of the scene. If in Global Transform Mode the scene will be scaled. If in Camera Transform Mode the size of the camera glyph will be changed.

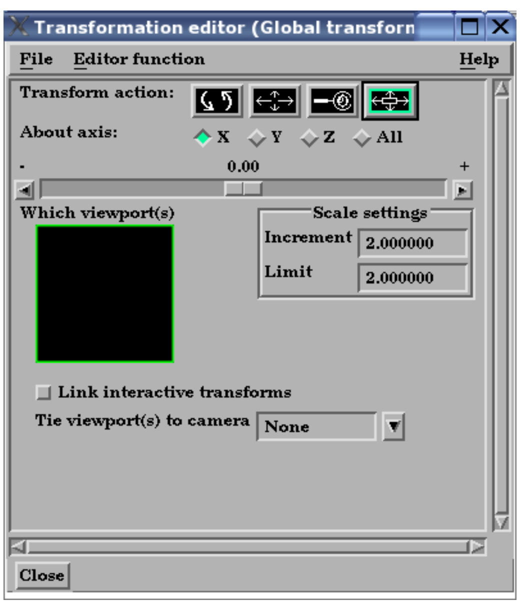


Figure 9-7
Transformation Editor for Exact Global Scaling

You may precisely rescale the scene or camera in the X, Y, Z, or All axes by clicking on the appropriate scaling direction and:

1. entering in the Increment Field the desired rescale factor and pressing Return (A value of .5 will reduce the scale of the scene in the chosen axis by half. A value of 2 will double the scale in the chosen axis. *Be aware that entering a negative number will invert the model coordinates in the chosen axis.*), then

2. clicking the stepper buttons at each end of the slider bar (Clicking the left stepper button will apply 1/Increment value to the scale. Clicking the right stepper button will apply the entire Increment value to the scale), or

3. dragging the slider in the positive or negative direction to the desired scale factor and then releasing the slider. (Dragging the slider to the leftmost position will apply 1/ Limit value to the scale. Dragging the slider to the rightmost position will apply the entire Limit value to the scale.).

Link Interactive Transforms

If you have multiple viewports you may link them together such that interactive (those performed in the graphics window) transformation that occurs in one of the linked viewports occurs in the other linked viewports. To link the viewports select all of the viewports to be linked in the Which viewport(s) window and turn on the Link interactive transforms button. An "L" will occur in the viewport outlines in the Which viewport(s) window indicating which viewports are linked.

Tie viewport(s) to camera

A viewport may either be in Global Transform Mode or in Camera Mode. In camera mode the viewport is viewed through a camera. This pulldown makes this choice. (see How to View a Viewport Through a Camera)

Reset Tools & Viewports Button

Clicking the Reset Tools and Viewport(s) button in the Transformation Control Area will open the Reset Tools and Viewport(s) dialog

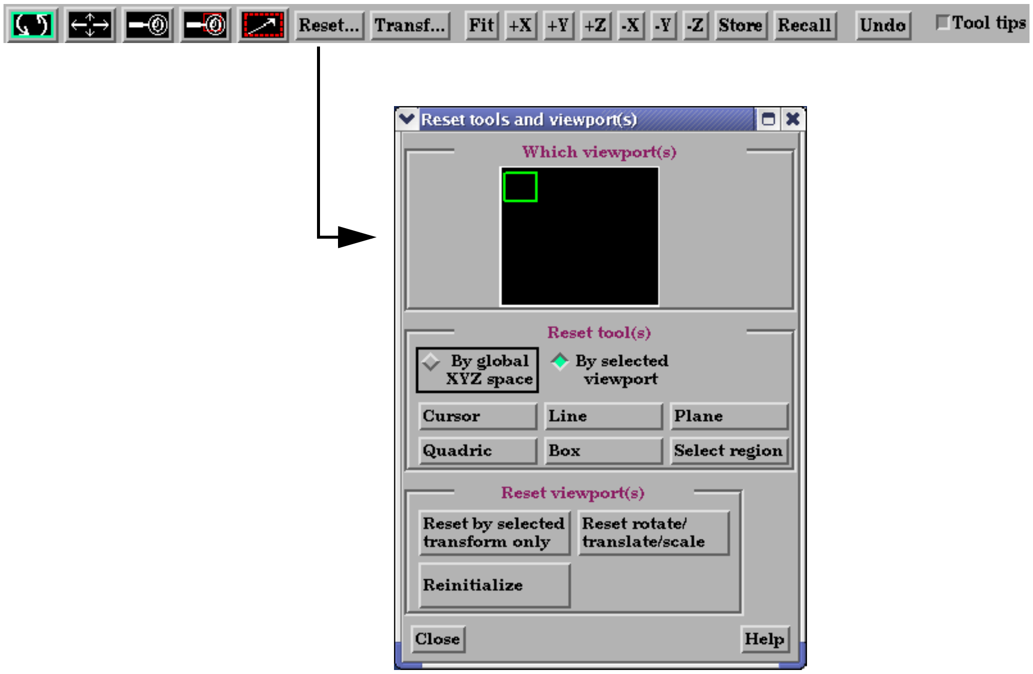


Figure 9-8
Reset Tools and Viewport(s) dialog

By Global XYZ Space Toggle — When enabled, clicking a Reset button will cause the Cursor, Line, Plane, or Quadric Tool to reset to its initial default position.

By Selected Viewport Toggle — When enabled, clicking a Reset button will cause the Cursor, Line, Plane, or Quadric Tool to be repositioned in the center of the geometry for the selected viewport.

Reset Cursor — Clicking this button will cause the Cursor Tool to reset according to the “By” toggle.

Reset Line — Clicking this button will cause the Line Tool to reset according to the “By” toggle.

Reset Plane — Clicking this button will cause the Plane Tool to reset according to the “By” toggle.

Reset Quadric — Clicking this button will cause the currently selected Quadric Tool to reset according to the “By” toggle.

Reset Box — Clicking this button will cause the Box Tool to reset according to the “By” toggle.

Reset Select region — Clicking this button will cause the selection Tool to reset according to the “By” toggle.

Reset By Selected Transform Only — Clicking this button will cause the transformation selected in the Transformation Control Area to reset for the viewports selected in the dialog’s Viewport(s) area.

Reset Rotate/ Translate/Scale — Clicking this button will cause the rotate, translate, and scale transformations to reset for the viewports selected in the dialog’s Viewport(s) area.

Reinitialize — Clicking this button will cause the viewports selected in the dialog’s Viewport(s) area to reset and recenter on the Parts which are visible in the Viewport(s).

Reset using Function Keys

Pressing the F5 button will change the scene in the current viewport to the standard “right side” view. Similarly, pressing F6 will show a “top” view and F7 a “front” view. Pressing F8 will restore the view to the one which existed before F5, F6, or F7 were pressed. If the Control Key is pressed at the same time as F5, F6, or F7, then the current view will be stored to the selected button.

9.2 Frame Definition

When Frame Definition has been chosen from the Transform/Definition button Pulldown menu or from the Editor Function menu in the Transformation Editor dialog, then actions you make will affect only the definition (origin and orientation) of the selected Frame(s). Frame 0's definition however, can not be changed.

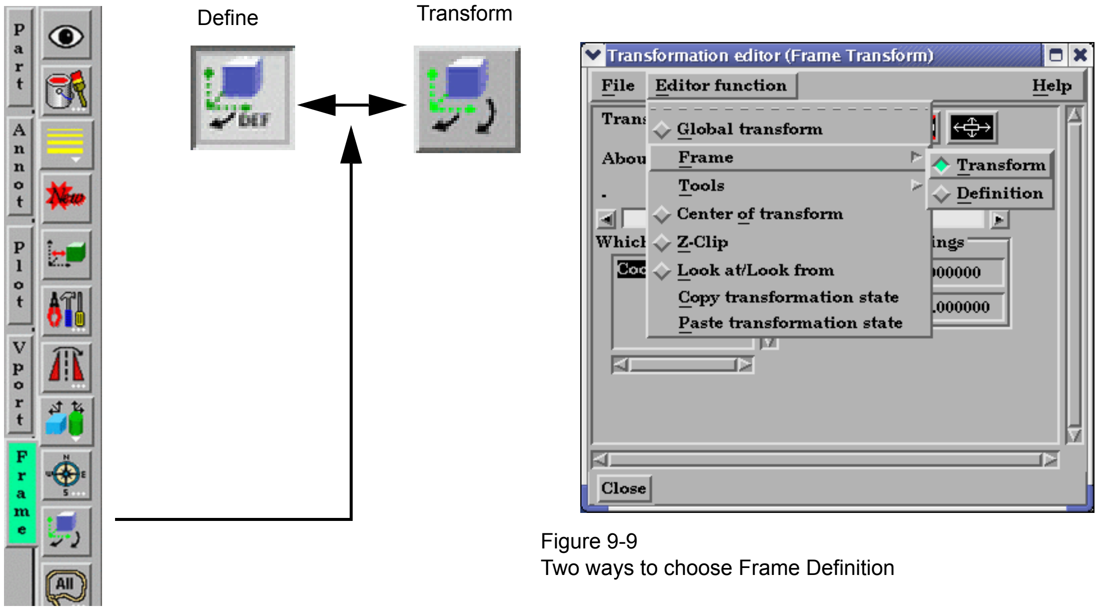


Figure 9-9
Two ways to choose Frame Definition

A Frame's definition should be adjusted before it is transformed under Frame Transform (as described in the previous pages). Transformations under Frame Transform are always about the Frame's origin and orientation. Failure to define the proper origin position and orientation of a Frame will result in unexpected transformation behavior.

You choose the type of transformation you wish to perform (rotate or translate) from the Transformation Control Icons. Note that you cannot perform zoom, scale, or reset operations under Frame Definition.

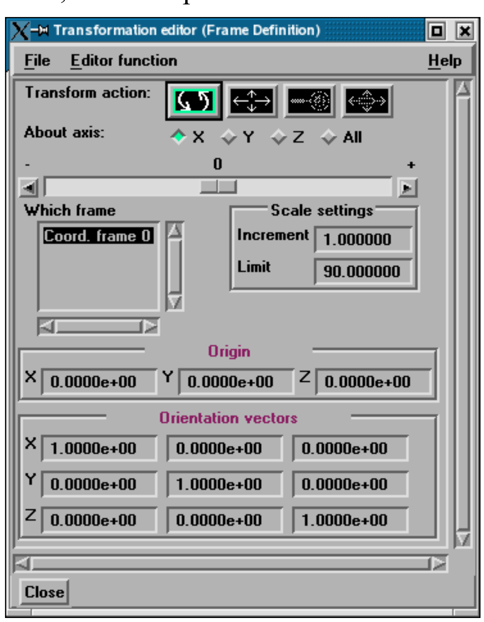


Figure 9-10
Transformation Control Area in Frame Mode
under Frame Definition

Rotate Toggle

Interactive Modification of Orientation

When this toggle is on, clicking the left mouse button and dragging modifies the orientation of the selected Frame(s). Clicking on the end of the X axis will limit the rotation to be about the Y axis. Similarly, clicking on the end of the Y axis will limit the rotation to be about the X axis.

Precise Modification of Orientation

When the Transformation Editor is opened under Frame Definition and the Rotate toggle is selected, the dialog will be configured to permit precise rotation (modification of the orientation) of the selected Frame(s).

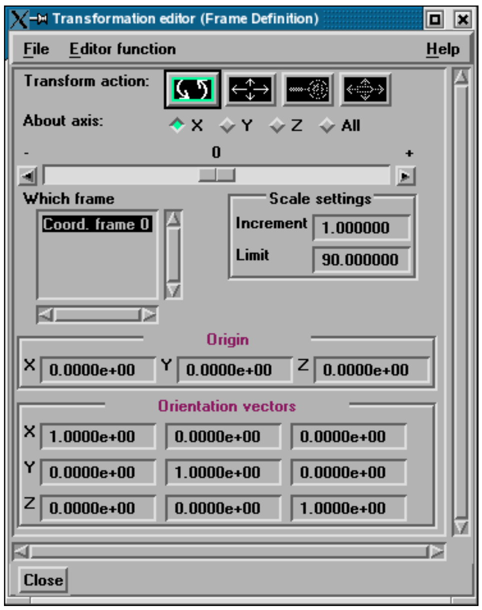


Figure 9-11
Transformation Editor for Exact Rotation for Selected Frame(s) Only

You may rotate the selected Frame(s) precisely about their X, Y, Z, or All axes by clicking on the desired axis and:

1. entering the desired rotation in (+ or -) degrees in the Increment field and pressing Return, then
2. clicking the stepper buttons at each end of the slider bar (each click will rotate the selected Frame(s) by the number of degrees specified in the Increment field), or
3. dragging the slider in the positive or negative direction to the desired number of degrees you wish to rotate the selected Frame(s) (the Limit Field specifies the maximum number of degrees of rotation performed when the slider is pulled to either end of the slider bar).

Origin XYZ
Orientation XYZ

You may precisely position both the origin and the axis of a selected Frame by entering in the desired coordinates in the Origin and Orientation Vector X Y Z fields and then pressing Return. These fields can be used regardless of whether the Rotate or the Translate toggle is selected.

Translate Toggle

Interactive Translation of Origin Position

When this toggle is on, clicking the left mouse button and dragging will translate the selected Frame(s) (other than Frame 0) up, down, left, or right within the viewport. Holding down the Control key while dragging will translate the selected Frame(s) forward or backward.

Precise Translation of Origin Position

When the Transformation Editor is open under Frame Definition and the Translate toggle is selected, the dialog will be configured to permit precise Translation (modification of the origin position) of the selected Frame(s).

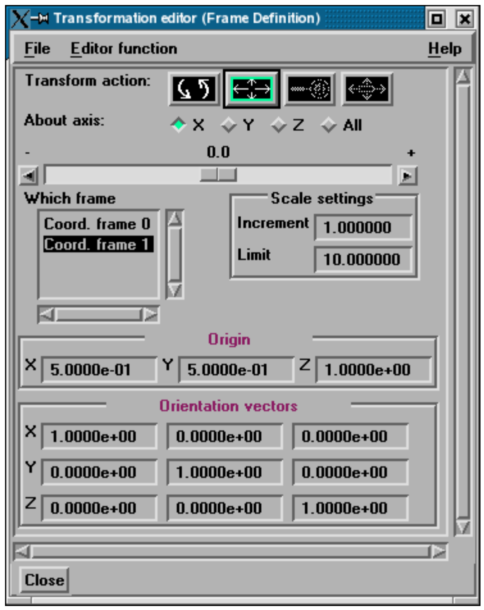


Figure 9-12
Transformation Editor for Exact Translation of Selected Frames

You may translate the selected Frame(s) precisely along the X, Y, Z, or All axes by clicking on the desired axis direction and:

1. entering the desired translation in (+ or -) model coordinate units in the Increment field and pressing Return, then
2. clicking the stepper buttons at each end of the slider bar (each click will translate the selected Frame(s) by the number of model coordinate units specified in the Increment field), or
3. dragging the slider in the positive or negative direction to the desired number of model coordinate units you wish to translate the selected Frame(s) and then releasing the slider (the Limit Field specifies the maximum number of model coordinate units that the Frame is translated when the slider is pulled to either end of the slider bar).

9.3 Frame Transform

When Frame Transform has been chosen from the Transform/Definition button Pulldown menu or from the Editor Function menu in the Transformation Editor dialog, transformations you make will affect the selected Frame(s) and the Parts assigned to them.

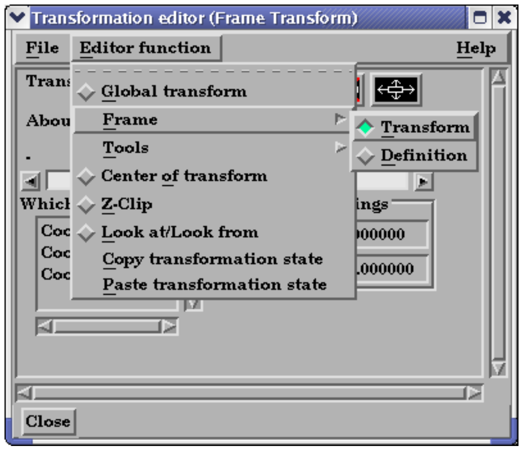


Figure 9-13
Two ways to choose Frame Transform

Note: *Before any transformations are performed on a Frame, its definition should be modified (if necessary) as described later in this section. Transformations always occur about a Frame's origin and orientation. Failure to define the proper position and orientation of the Frame will result in unexpected transform behavior. Thus, the order of dealing with things should be 1) define the frame, 2) assign parts to the frame, 3) transform according to the frame.*

You choose the type of transformation you wish to perform from among the Transformation Control Icons in the Transformation Editor dialog. Note that under Frame Transform, you cannot perform the zoom operation.

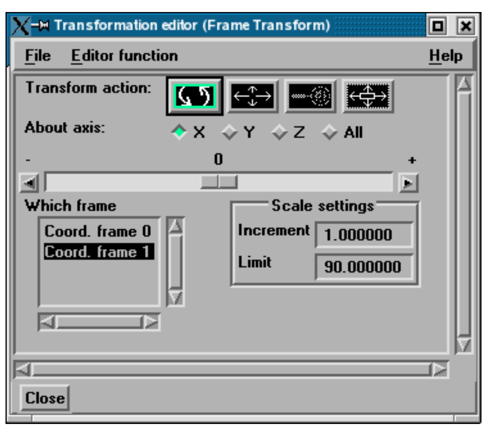
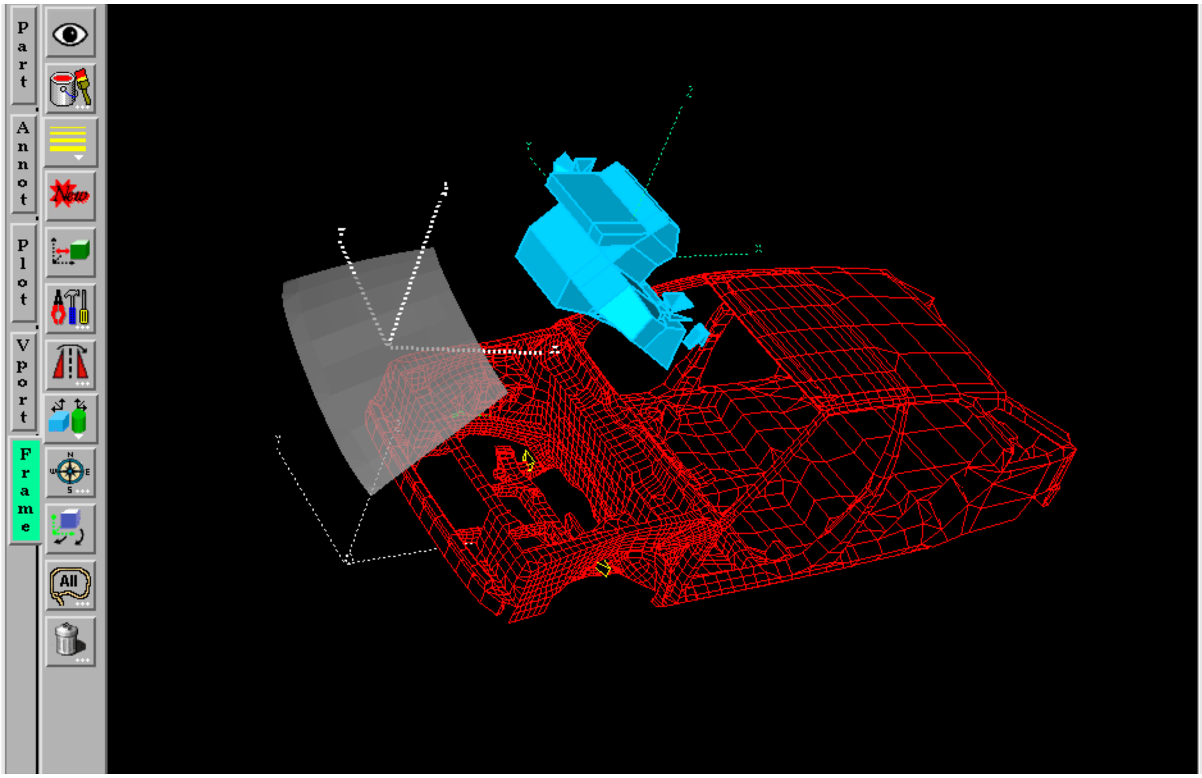


Figure 9-14
Transformation Control Area in Frame Mode under Frame Transform



Rotate Toggle

Interactive Rotation — When this toggle is on, clicking the left mouse button and dragging causes the selected Frame(s) and all Parts assigned to the Frame(s) to rotate about the Origins of each Frame Axis. Holding down the Control key while dragging will rotate the selected Frame(s) and all assigned Parts about a Z axis perpendicular to the screen.

Precise Rotation — When the Transformation Editor is open under Frame Transform and the Rotate toggle is selected, the dialog will be configured to permit precise Rotation.

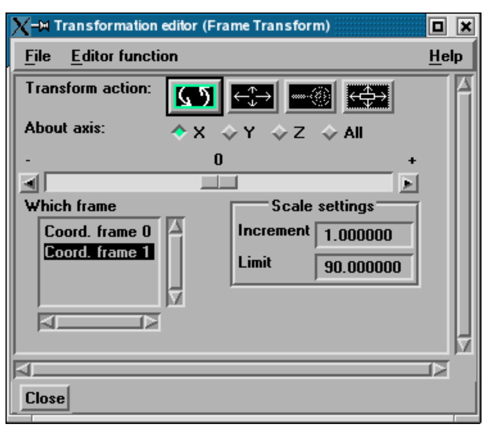


Figure 9-16
Transformation Editor for Precise Rotation under Frame Transform

You may rotate the selected Frame(s) and assigned Part(s) precisely about the X, Y, Z, or All axes, as the orientation of the axes were defined when the Frame was first created by:

1. entering the desired rotation in (+ or -) degrees in the Increment field and pressing Return, then
2. clicking the stepper buttons at each end of the slider bar (each click will rotate the selected Frame(s) and assigned Part(s) by the number of degrees specified in the Increment field), or
3. dragging the slider in the positive or negative direction to the desired number of degrees you wish to rotate the selected Frame(s) and assigned Part(s) (the Limit Field specifies the maximum number of degrees of rotation performed when the slider is pulled to either end of the slider bar).

Translate Toggle

Interactive Translation

When this toggle is on, you can transform objects interactively in the X-Y plane (or by holding down the Control key, in Z). Clicking the left mouse button and dragging will translate the selected Frame(s) and all assigned Part(s) up, down, left or right (or forward or backward) within the selected viewport.

Precise Translation

When the Transformation Editor is open under Frame Transform and the Translate toggle is selected, the dialog will be configured to permit precise Translation.

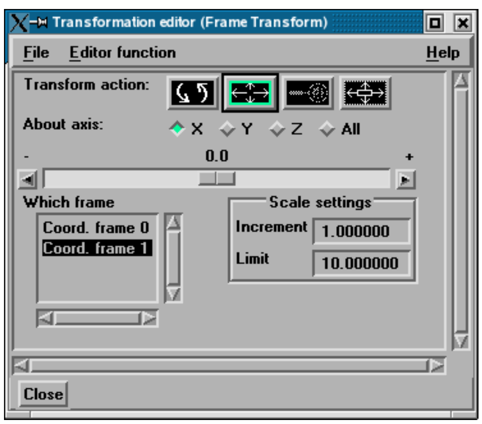


Figure 9-17
Transformation Editor for Precise Translation under Frame Transform

You may translate the selected Frame(s) and all Parts assigned to them precisely along the X, Y, Z, or All axes by:

1. entering the desired translation in (+ or -) model coordinate units in the Increment field and pressing Return, then
2. clicking the stepper buttons at each end of the slider bar (each click will translate the selected Frame(s) and assigned Part(s) by the number of model coordinate units specified in the Increment field), or
3. dragging the slider in the positive or negative direction to the desired number of model coordinate units you wish to translate the selected Frame(s) and assigned Part(s) and then releasing the slider (the Limit Field specifies the maximum number of model coordinate units that the model is translated when the slider is pulled to either end of the slider bar).

Scale Toggle

When the Transformation Editor is open under Frame Transform and the Scale toggle is selected, the dialog will be configured to permit precise scale.

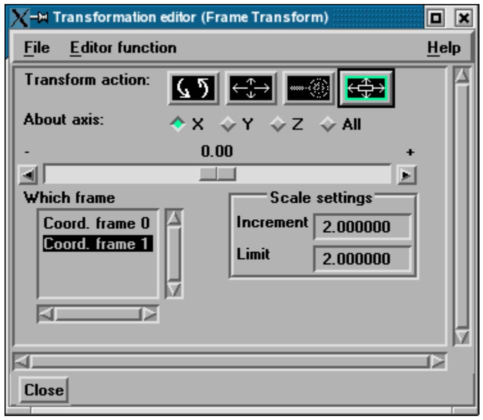


Figure 9-18
Transformation Editor for Exact Scaling under Frame Transform

You may precisely rescale the selected Frame(s) and assigned Part(s) in the X, Y, Z, or All axes by:

1. entering in the Increment Field the desired rescale factor and pressing Return (A value of .5 will reduce the scale of the selected Frame(s) and assigned Part(s) in the chosen axis by half. A value of 2 will double the scale in the chosen axis. *Be aware that entering a negative number will invert the model coordinates in the chosen axis.*), then
2. clicking the stepper buttons at each end of the slider bar (Clicking the left stepper button will apply 1/Increment value to the scale. Clicking the right stepper button will apply the entire Increment value to the scale), or
3. dragging the slider in the positive or negative direction to the desired scale factor and then releasing the slider. (Dragging the slider to the leftmost position will apply 1/Limit value to the scale. Dragging the slider to the rightmost position will apply the entire Limit value to the scale.)

9.4 Tool Transform

Transformation of the Cursor, Line, Plane, Box, Selection, and Quadric (cylinder, sphere, cone, and revolution) Tools is covered in depth in Chapter 6.

(see *Tool Positions* in Section 6.5, Tools Menu Functions)

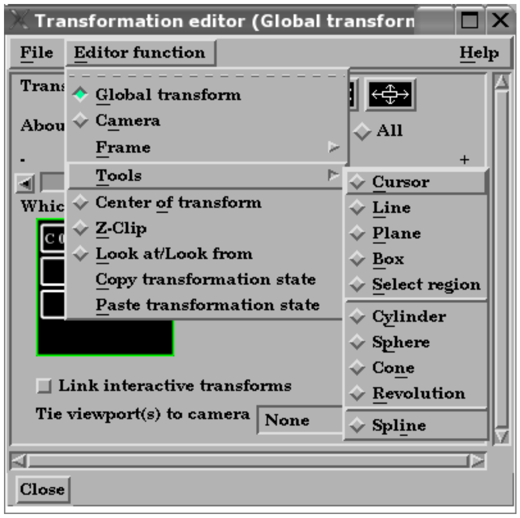


Figure 9-19
Transformation Editor Tools Selections

9.5 Center Of Transform

The point about which global transformations will occur can be specified exactly if desired. Simply enter the model coordinates for the location of this point.

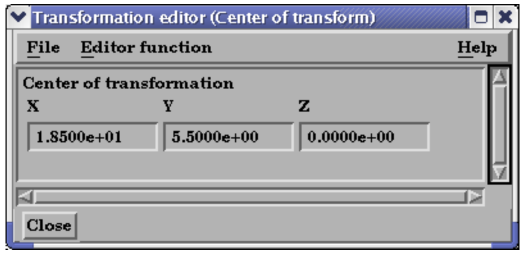


Figure 9-20
Transformation Editor Center of Transform dialog

Alternatively you can click on the pick icon and pull down to 'Pick center of transform', and then move the cursor over the part location (make sure the EnSight graphics window is active) and pick using the 'p' key.

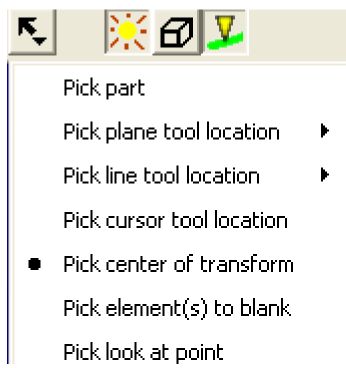


Figure 9-21
Pick Center of Transform pulldown

Alternatively, you can click on the Fit button which will reset the center of tranform to the geometric center of the visible parts.

9.6 Z-Clip

EnSight displays the scene in a three-dimensional, rectangular workspace that has finite boundaries on all sides. Even if you rotate the model, you are always looking into the workspace from the front side. The top-to-bottom and side-to-side boundaries of the workspace are analogous to looking out a real window—the window frame limits your view. In addition, since the memory of your computer is finite, your workspace also has limits in the front-and-back direction.

The front boundary is the *Front Clipping Plane* (or the *Near Plane*) and the rear boundary is the *Back Clipping Plane* (or the *Far Plane*). Only the portion of the scene *between* these two planes is visible—the rest of the model (if any) is *clipped* and therefore invisible. By convention, the front-to-back direction of the workspace is the Z direction. Hence, the front and back clipping planes are together called the *Z-Clip Planes*. Note that the Z-direction in the workspace is always in-and-out of the screen and is completely independent of the Z-direction of the model Frame (Frame 0).

Z-Clip Positions

The position of the Z-Clip planes is specified in terms of their *distance from the Look From Point* in the distance units implied by the model-geometry data. By default, the planes automatically move as the model moves.

Initially, EnSight positions the Z-Clip Planes based on the dimensions of the model parts read to the Client, with some extra space for you to perform transformations. You can reposition the planes when doing so becomes necessary or desirable.

Each viewport has its own independently adjustable set of Z-Clip Planes.

Using Z-Clip Planes

You can use Z-Clip planes to *deliberately* clip-away portions of the model you are not interested in, or which are getting in the way of what is of interest. For example, you can clip-away both a front-portion and a back-portion of a model to reduce the number of node and element labels displayed. *Z-Clip Planes and*

Hidden Surfaces

EnSight uses your workstation's graphics hardware to perform all graphics manipulations, including the display of solid surfaces. The appearance of a solid model is created by *not* displaying *hidden surfaces*—surfaces hidden behind nearer surfaces. The algorithm used by the graphics hardware to do this task— *Z-buffering*—is a simple algorithm which compares Z-values to calculate which surfaces are closest to you and thus visible. Z-buffering is normally performed in integer arithmetic, and on most graphics systems is confined to 24 bits of resolution. Hence, the coordinates in Z must be mapped into this 24-bit space. To achieve the maximum resolution possible in the 24 bits available, the graphics hardware maps the Z-distance between the Front and Back Clipping Planes into the 24 bits available. Hence, the larger the distance between the Z-Clip Planes, the lower the Z resolution, which can affect image quality for solid images. If you see problems with your solid images, move the front and back clipping planes in as close as possible.

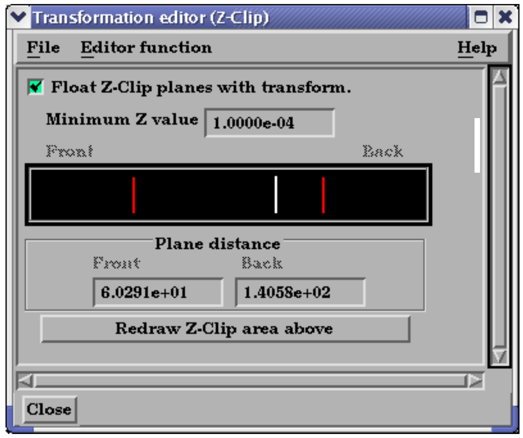


Figure 9-22
Transformation Editor for Z-Clip Plane Positions

The Transformation Editor (Z-Clip) is used to adjust the distances of the Front and Back Clipping Planes from the Look-From Point.

Float Z Clip Planes With Transform
When on, will automatically adjust the front and back Z-Clip planes away from the model.

Minimum Z Value
Minimum distance the Front Clipping Plane is allowed to float to from the Look From Point (model coordinates). Used only if Float Z Clip Planes with Transform toggle is on.

Z-Clip Area Display
Displays position of Z-Clip planes relative to model-part Z-range (shown as a rectangle) and allows interactive positioning (by clicking and dragging) of the Z-Clip planes. If lines are inside model rectangle, that part of model is clipped from the display. Values update in data fields as you move sliders. Active viewports of the Main View update automatically as you move sliders.

Plane Distance

Front
Distance of the Front Clipping Plane from Look From Point in model coordinates. Precisely specify by typing in desired distance and pressing Return. Not used if the Float Z Clip Planes With Transform toggle is on.

Back
Distance of the Back Clipping Plane from the Look From Point in model coordinates. Precisely specify by typing in desired distance and pressing Return. Not used if the Float Z Clip Planes With Transform toggle is on.

Redraw
The Plane Position Display does not automatically update if you perform transformations in the active viewport. Click this button to update the Plane Position Display.

Troubleshooting Z-Clip Planes

Problem	Probable Causes	Solutions
Main View is empty	No parts located between Front and Back Z-Clip Planes.	Adjust Z-Clip plane locations
Model degenerates to irregular polygons or the front Z-Clip line is locked in the model extent box	You have moved the front Z-Clip plane too close to (or on) the Look From Point.	Move the front Z-Clip plane away from the Look From Point.

9.7 Look At/Look From

Using the Transformation Editor with Editor Function > Look At/Look From chosen, you can reposition the point from which you are observing the model (the Look From Point) and the point at which you are looking (the Look At Point). Both the Look-From and Look-At points are specified in the coordinates of the Model Frame (Frame 0).

Initially, the Look At Point is at the geometric center of the initial model parts read by the EnSight Client. The Look From Point is on the positive Z-axis at a distance appropriate to display the model in the Main View window.

If you increased only the X position of the Look From Point, in the Graphics Window (or selected Viewport), it would appear that the model had rotated about the Global Y axis. In fact, the model has not rotated at all, which is shown by the visible Global Axis triad in the figure below. What has happened is that you are now viewing the model from a position farther to the right than previously.

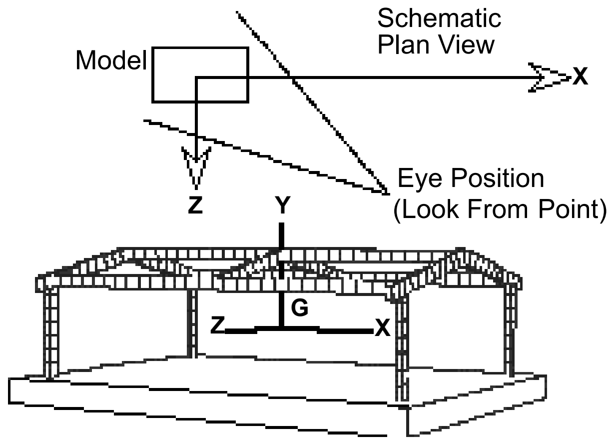


Figure 9-23
Image showing view of model from negative X axis towards positive X axis

If the Y and Z coordinates of the Look From point were made to be the same as those of the Look At point, but the X coordinate of Look From point was specified as a much smaller value than that of the Look At point, it would appear in the Graphics Window (or selected Viewport) that the model had rotated 90 degrees about the Global Y axis. As before, the model has actually not rotated at all, which is shown by the visible Global Axis triad in the figure below. What has happened is that you are now viewing the model from a position on the negative Global X axis looking in the positive X direction.

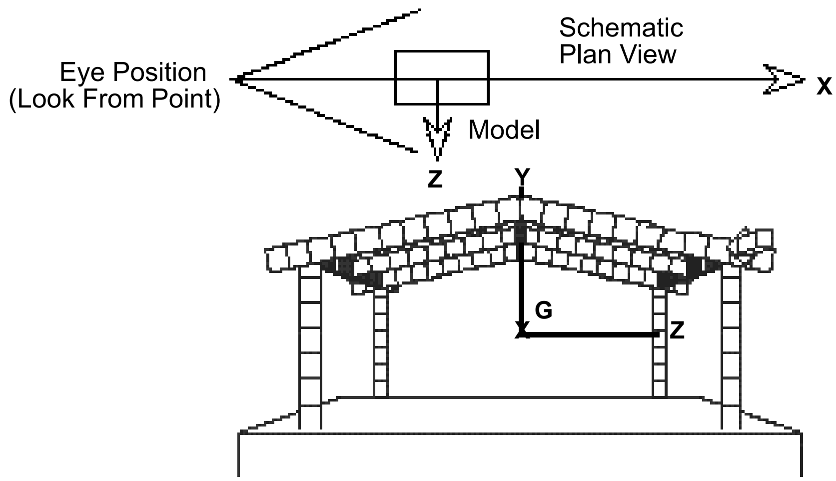


Figure 9-24
Image showing view of model from negative X axis towards positive X axis

The position of the Look-At and Look-From points can be interactively or precisely specified using the Transformation Editor dialog with Editor Function > Look At/Look From.

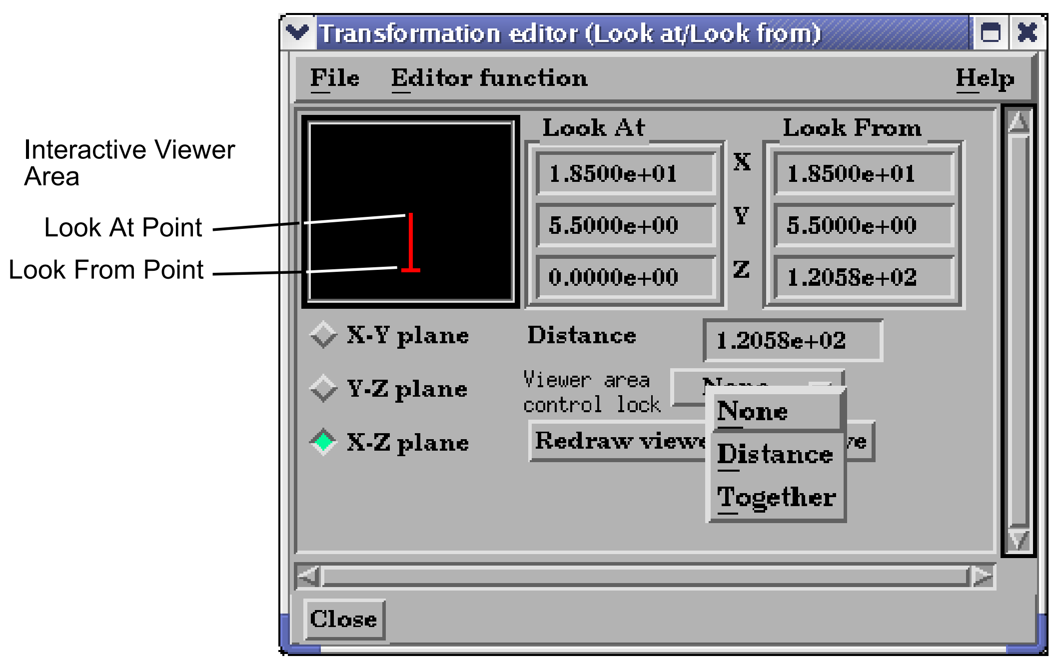


Figure 9-25
Transformation Editor for Look At/Look From

Interactive — The position of the Look At and Look From Points may be positioned interactively in the Interactive Viewer Area by grabbing the Look At or Look From Point and dragging it to the desired location. These interactive modifications can be made in the X-Z Plane, the X-Y Plane, or the Y-Z Plane, depending upon which of the three toggles are selected. The Graphics Window as well as the Look At and Look From coordinate fields updates as you drag either Point to a new location.

Precise — The position of the Look At and Look From Points may be positioned precisely by specifying the desired coordinate values in the X Y Z fields and pressing Return.

Distance — The distance in model coordinates may be precisely specified by entering the desired value in this field and pressing Return.

Viewer Area Control Lock — Opens a pop-up menu for the selection of how interactive actions taken in the Viewer Area will be limited. Choices are:

None	No locks are applied
Distance	The distance between the two Points is locked
Together	The distance and direction vector between the two Points is locked

Redraw Viewer Area Above — This button redraws the Viewer Area. This button should be clicked after a transformation is performed in the selected viewport while this dialog is active.

9.8 Copy/Paste Transformation State

This transformation option can be used to apply the transformation state of one viewport to other viewports. Useful if you want multiple viewports to have the model oriented the same, and you did not link the viewports for transformations before applying any transformations.

The use of this option consists of:

1. Selecting the viewport (one only) containing the transformation state desired.

   (You can do this in Vport mode in the graphics screen, or under Editor Function -> Global Transform in the Transformation Editor Dialog.)

2. Selecting *Copy Transformation State* under Editor Function in the Transformation Editor dialog.

3. Selecting the one or more viewports to receive this transformation state.

   (As in 1. above)

4. Selecting *Paste Transformation State* under Editor Function in the Transformation Editor dialog.

9.9 Camera

If a Viewport is in Camera Mode (see "Tie Viewports to Camera" in Section 9.1, Global Transform) the viewport will be viewed through the camera chosen. Any global transforms that have been created for the viewport are ignored and any transforms performed will be for the camera selected.

(also see How to View a Viewport Through a Camera)

Rotate Toggle

Interactive Rotation: When this toggle is on, clicking the left mouse button and dragging horizontally in a viewport that is being viewed through a camera will rotate the camera about the Camera Y axis. Holding the Control Key down and then clicking the left mouse button and dragging will rotate the camera about the Camera Z axis.

Precise Rotation: When the Transformation Editor is open under Camera and the Rotate toggle is selected, the dialog will be configured to permit precise Rotation of the selected camera(s).

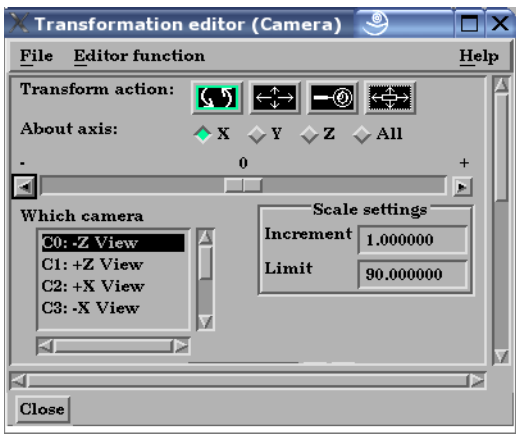


Figure 9-26
Transformation Editor for Camera Rotation

You may rotate the selected camera(s) selected precisely about the X, Y, Z, or All camera axis by clicking on the appropriate axis of rotation toggle and entering the desired rotation in (+ or -) degrees in the Increment field and pressing Return, or by clicking the stepper buttons at each end of the slider bar (each click will rotate the camera by the number of degrees specified in the Increment field), or by dragging the slider in the positive or negative drection to the desired number of degrees you wish to rotate (note: the Limit Field specifies the maximum number of degrees of rotation performed when the slider is pulled to either end of the slider bar).

Translate Toggle

Interactive Translation: When this toggle is on, clicking the left mouse button and dragging in a viewport that is being viewed through a camera will translate the camera left/right, or up/down (or by holding the Control key, in/out).

Precise Translation: When the Transformation Editor is open under Camera and the Translate toggle is selected, the dialog will be configured to permit precise Translation of the selected camera(s).

Figure 9-27
Transformation Editor for Camera Translation

You may translate the camera(s) selected precisely about the X, Y, Z, or All camera axis by clicking on the appropriate axis of translation toggle and entering the desired translation in (+ or -) model coordinate units in the Increment field and pressing Return, or by clicking the stepper buttons at each end of the slider bar (each click will translate the camera by the number of units specified in the Increment field), or by dragging the slider in the positive or negative direction to the desired number of units you wish to translate (note: the Limit Field specifies the maximum number of units in model coordinates performed when the slider is pulled to either end of the slider bar).

Zoom Toggle

Interactive Zooming

When this toggle is on, clicking the left mouse button and dragging in a viewport that is being viewed throgh a camera will move the camera in the camera z-direction. The result is exactly the same as if you translated the camera in the z-direction using the Translate Toggle.

Precise Zooming

When the Transformation Editor is open under the Camera and the Zoom toggle is selected, the dialog will be configured to permit precise Zoom operations of the selected camera(s).

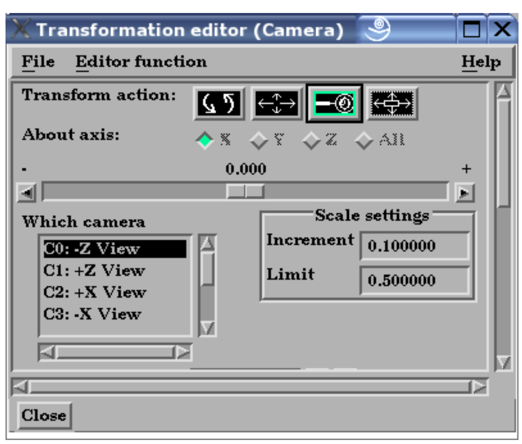


Figure 9-28
Transformation Editor for Camera Zoom

You may zoom the camera(s) (moving the camera in the camera z-direction) selected precisely by entering the desired camera z-axis movement in (+ or -) model coordinate units in the Increment field and pressing Return, or by clicking the stepper buttons at each end of the slider bar (each click will translate the camera in the z camera axis direction by the number of units specified in the Increment field), or by dragging the slider in the postitive or negative direction to the desired number of units you wish to move (the Limit Field specifies the maximum number of units in model coordinates performed when the slider is pulled to either end of the slider bar).

Scale Toggle

When the Transformation Editor is open under Camera, and the Scale toggle is selected, the dialog will be configured to permit Scale operations of the selected camera(s).

Figure 9-29
Transformation Editor for Camera Scale

A Scale operation of a camera does not affect the transformations that are occuring in the viewport - they affect the size of the camera glyph. You may modify the size of the camera glyph by selecting the X, Y, Z, or All axis (they will all perform the same camera resize operation) toggles and then entering the desired camera glyph scale factor in the Increment field and pressing Return (values < 1 will make the camera glyph smaller while values > 1. will make it larger), by clicking the stepper buttons at each end of the slider bar (each click will scale the camera glyph up/down by the factor specified in the Limit field), or dragging the slider in the postitive or negative direction to scale the camera up or down (the Limit Field specifes the maximum scale factor performed when the slider is pulled to either end of the slider bar).

Which camera

Selects the camera(s) being modified

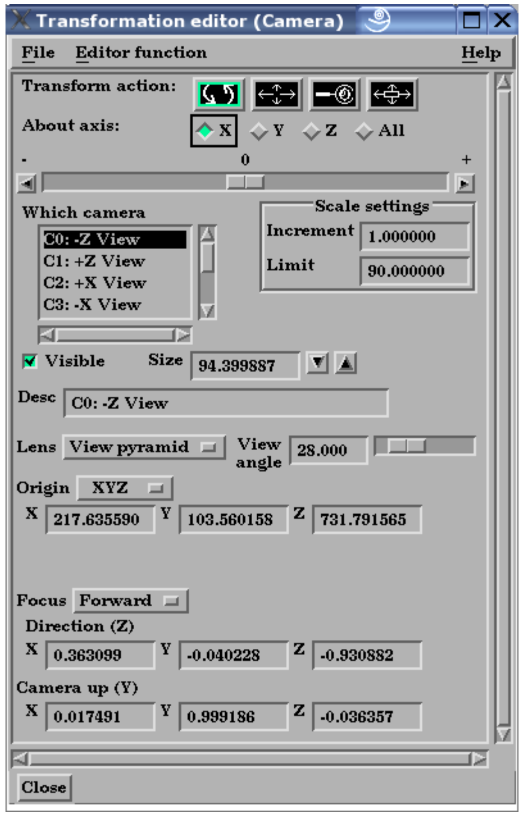


Figure 9-30
Transformation Editor for Camera Scale

File Restore Camera Position
This opens the Restore camera dialog which allows you to specify the name of a file in which a camera location and orientation is specified (see Chapter 11.20, Camera Orientation File Format).

Visible
Sets the visiblity of the camera(s) selected

Size
Sets the size of the camera glyph for the selected camera(s). Size is in model coordinates.

Desc
Description of the camera

Lens
View pyramid will show the view contraints if the camera were used to view the viewport. Classic shows a classical "lens" on the camera with no hint of the view volume.

View Angle
The view angle in degrees that will be used with the camera. Small values decrease the view angle and simulate a telephoto lens while large values increase the view angle and simulate a wide angle lens. Current limitiation is 5 < view angle < 120

Origin
Sets the camera origin

XYZ
The model coordinates of the camera

Node
Sets the Camera origin at a specific part node id number

Spline
Set the Camera origin to lie on a defined spline

Offset
If the Origin is set to Node or Spline it is possible to offset the origin from the node or spline by a XYZ value

Focus
Defines the orientation of the camera

Forward
If the origin is Spline then focus is "forward" on the spline. If the origin is not Spline then the focus is defined by the Direction(Z) fields.

Node
Set the focus to a specified part and node id

XYZ
Set the focus to a specific location

Camera up (Y) Sets the vector defining the "tilt" of the camera. Will be adjusted if the Focus is set to Node or XYZ to form a right handed orthogonal coordinate system.

10 Preference and Setup File Formats

This chapter provides information about the various file formats associated with different preference options within EnSight.

Section 10.1, Window Position File Format describes the format of the file which contains your saved window positions and sizes.

Section 10.2, Connection Information File Format describes the format of the file which contains your auto-connection information.

Section 10.3, Palette File Formats describes the format of the color selector palette, saved function palettes, and the default false color function palette.

Section 10.4, Default Part Colors File Format describes the file format for the default colors for parts.

Section 10.5, Data Reader Preferences File Format describes the format for the data reader preferences file.

Section 10.6, Data Format Extension Map File Format describes the format of the ensight_reader_extension.map file.

Section 10.7, Parallel Rendering Configuration File points to the location where the format of the parallel rendering configuration file is described.

Section 10.8, Resource File Format describes the format of the ensight resources file and points to the location where samples of its use are given.

10.1 Window Position File Format

To save a window position file, click Edit >Preferences... from the Main Menu and select the "General User Interface" option. Then select the Save Size and Position of Main Windows button. When you select this button, the current position of major dialog windows is saved to the `ensight9.winpos.default.XRESxYRES` file. In general, this file contains dialog position and size information, along with information about the states of the expandable sections of dialogs.

This file is normally saved to and read from EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac). If the file is in the Client working directory it will be read from and saved to that directory instead.

Only major dialogs are affected; the miscellaneous pop-up dialogs are not specified. You do not have to include every dialog and every section listed. EnSight will process the ones provided.

Window File Format

The format of the EnSight window position file is as follows:

- Line 1: Font Size

  Integer specifying font size for dialog labels.

- Lines 2 to N: Dialog Title, Size, & Location

  `String: [IntegerXInteger+]Integer+Integer` specifying
  Dialog title: Width x Height + Xloc + Yloc. The dialog title of each window can be shortened using the * as a meta character. For example, the string title Transformation Editor: 0+815 can be shortened to *Transform*: 0+815. Be careful that your abbreviated name does not match any other names, or the position of all those names will be changed.

- Line N+1: List Separator String

  Character string `–Section Expansion Information–` to separate dialog size and location information from section-open information.

- Lines N+2 to End: Section Expansion Toggles

  `Dialog->Section[->Section]: open|closed` character strings indicating whether corresponding dialog section is open or closed.

The following is an example window position file:

```
fontsize: 13
EnSight: 910x984+369+31
Transformation Editor: 390x381+180+368
Command: 300x0+0+682
Connect Server: 137+0
Query Dataset: 0+0
```

10.2 Connection Information File Format

EnSight saves a file on the Client host system, called ensight_comm_settings which contains the possible pre-configured server and sos connection settings. The file is saved from the Case->Connection settings... dialog. One of the connections is always the default for both normal server and sos connections.

The complete ASCII text file contains the following Server and Sos connection information - there is no limit to the number of possible connections that can be described but only one of the connections (one each for server and sos) can be designated the default.

```
# EnSight Connection Settings Version 1.1
#
ENTRY
SERVER: YES
DEFAULT: YES
USE_RSH: YES
CONN_NAME: "localhost"
HOST_NAME: "localhost"
RSH_CMD: ""
LOGIN_NAME: ""
EXECUTABLE: "ensight92.server"
WORKING_DIR: ""

ENTRY
SERVER: NO
DEFAULT: YES
USE_RSH: YES
CONN_NAME: "localhost"
HOST_NAME: "localhost"
RSH_CMD: ""
LOGIN_NAME: ""
EXECUTABLE: "ensight92.sos"
WORKING_DIR: ""

ENTRY
SERVER: YES
DEFAULT: NO
USE_RSH: YES
CONN_NAME: "mydebug"
HOST_NAME: "localhost"
RSH_CMD: ""
LOGIN_NAME: ""
EXECUTABLE: "/home/metoo/ang/ensight_dev/ensight_92/server/ensight9.server"
WORKING_DIR: ""
```

Each new connection is designated by the ENTRY key word and is followed by keywords with appropriate system information. The system variables are shown in the table below.

Keyword	Description
SERVER	YES if this connection is a standard server. NO if the connection is for SOS.

Keyword	Description
DEFAULT	YES if this is the default connection. NO otherwise.
USE_RSH	YES if the server will be started with the standard rsh command. NO otherwise. (see RSH_CMD below)
RSH_CMD	If USE_RSH is NO, then this describes which command to use in it's stead.
CONN_NAME	The name of the connection as it will appear in the Connection Settings dialog list.
HOST_NAME	The id or hostname of the system where the server/sos will be executed. "localhost" is defined to mean the same machine that the client is running on.
LOGIN_NAME	Your alternate login id on the server/sos host system. This default to your Client host system login id. (This option is only applicable to distributed connections).
EXECUTABLE	The complete path to the executable program to start for the server/sos. This default to ensight92.server or ensight92.sos (which must be in your defined UNIX search path). This path is normally defined in your .cshrc or .login file in your home directory (for C shell users).
WORKING_DIR	The directory where the server/sos will be started (the file browser will see the files in this directory without having to modify the path).

10.3 Palette File Formats

The following palette formats are discussed in this section:

Color Selector Palette File Format

Function Palette File Format

Predefined Function Palette

Default False Color Map File Format

Color Selector Palette File Format

This file defines the colors that are used with the EnSight Color Selector. If EnSight does not find a definition file it uses a default palette. If, however, it does find a file (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac) at start-up it will read your colors and show them in the Color Selector.

The format of the `ensight.colpal.default` file is as follows:

- Line 1: “Version 6.0” (Note, this need not match EnSight’s version number.)
- Line 2 through Line 37

  Three integers, one for each color (red, green, blue), ranging from

  0 (no intensity) to 255 (full intensity).

Function Palette File Format

A function palette file is saved using the Function Editor when you save (one or more) function color palettes. The following is an example function palette file:

```
palette ‘velocity’
variable_type vector
variable ‘velocity’
type continuous
limit_fringes no
scale linear
number_of_levels 5
colors
0.000000 0.000000 1.000000
0.000000 1.000000 1.000000
0.000000 1.000000 0.000000
1.000000 1.000000 0.000000
1.000000 0.000000 0.000000
values
0.100341
0.301022
0.501704
0.702385
0.903067
```

Many lines of the file consists of a descriptive keyword followed by an appropriate value. In other areas the keyword is used to start a block of information. The values are all free format real or integer numbers or string constants. The palette name must have single quotes around each name. The string keywords and constant values must match exactly.

Keyword	Description
`palette`	Name of the palette when one name is present. Name of the subpalette when two names are present (ex. palette ‘velocity’ ’xcomp’)
`variable`	Name of the variable used with the palette.
`variable_type`	Type of the variable, scalar or vector.
`type`	Type of the palette, continuous or banded.
`limit_fringes`	Indicates if the palette is set up for limiting fringe. If it is, the options are `by_Part` or `by_invisible`.
`scale`	Indicates whether the palette scale is linear, logarithmic, or quadratic.
number_of_levels	Indicates the number of levels defined for the palette.
colors	Indicates the start of a block of RGB triplets, 1 triplet per line. There will be the same number of lines as there are levels.
values	Indicates the start of a block of level values. There will be the same number of values as there are levels.

Predefined Function Palette

When EnSight starts, it looks for user defined function color palettes located under `$CEI_HOME/ensight92/site_preferences/palettes` and in the `palettes` subdirectory under the user’s EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\username\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac). These files must be named `palette_name.cpal`, where the palette_name is the name of the color palette in the Simple Interface area of the function dialog.

The format of the `.cpal` file is as follows:

- Line 1: The string “`number_of_levels x`”, where x is an integer.
- Line 2: The string “`colors`”
- Line 3 through x + 2: Three float values in range 0.0 to 1.0, indicating red, green, and blue color components.

An example color palette file:

```
number_of_levels 5
colors
```

```
.008 0. 0.
.5 0. 0.
1. 0. 0.
1. 1. 0.
1. 0. 1.
```

Default False Color Map File Format

This file defines the default false-color map color range that is assigned by EnSight to each palette when variables are activated. If EnSight does not find a definition file, it uses an internal default list. If, however, EnSight does find a file at start-up, EnSight will read your colors as the default color palette colors. Note the file must be called `ensight.false_color.default` and be located in the EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac) or be located in `$CEI_HOME/ensight92/site_preferences`).

The format of the `ensight.false_color.default` file is as follows:

- Line 1: "Version 6.0" (Note, this need not match EnSight's version number.)
- Line 2: One integer, the number default false color map colors
- Line 3 on: three floats (each ranging between 0. and 1.), the (red, green, blue) color triplet of each color, each listed on separate lines.

An example default file can be found in:

```
$CEI_HOME/ensight92/site_preferences/ensight.false_color.default
```

on your client system.

The following is an example default false color map file with 5 colors; blue, cyan, green, yellow, and red:

```
Version 6.0
5
0. 0. 1.
0. 1. 1.
0. 1. 0.
1. 1. 0.
1. 0. 0.
```

10.4 Default Part Colors File Format

This file defines default Constant Colors that are assigned (and cycled through) by EnSight when parts are built. If EnSight does not find a definition file it uses an internal default list. If, however, EnSight does find a file (the file must be called `ensight.part.colors.default` and be located in the EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac) or be located in `$CEI_HOME/ensight92/site_preferences`) at start-up, EnSight will read your colors as the default Constant Colors.

The format of the `ensight.part.colors.default` file is as follows:

- Line 1: "Version 6.0" (Note, this need not match EnSight's version number.)
- Line 2: One integer, the number of default part colors
- Line 3 on: three floats (each ranging between 0. and 1.), the (red, green, blue) color triplet of each color, each listed on separate lines.

An example default file can be found in:

```
$CEI_HOME/ensight92/site_preferences/ensight.part.colors.default
```

on your client system.

The following is an example default part colors file with 6 colors (blue, cyan, green, yellow, red, and magenta):

```
Version 6.0
6
0. 0. 1.
0. 1. 1.
0. 1. 0.
1. 1. 0.
1. 0. 0.
1. 0. 1.
```

10.5 Data Reader Preferences File Format

This is an optional file that will be created when the user saves preferences under Main Menu > Edit > Preferences... Data. It can contain two basic things: 1) the reader name desired to be the default Format in the Data Reader dialog, and/or 2) any reader names that the user does NOT want to appear in the Format list. The default data Format will be "Case" unless this file exists and overrides it. Also, by default, all readers will appear in the list of available reader Formats unless specifically set to be removed in this file. The file must be called `ensight_reader_prefs.def` and be located in the EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac) or be located in `$CEI_HOME/ensight92/site_preferences.`

The format of the `ensight_readers_prefs.def` file is as follows:

- Line 1: "Version 7.1" (Note, this need not match EnSight's version number.)
- Line 2: "select *readername*" Where *readername* is the name of the reader that will be used as the default
- Line 3 on: "remove *readername*" Where *readername* is the name of a reader that will NOT be shown in the data reader Format list.

The following is an example data reader preferences file which sets EnSight 5 as the default Format, and causes the Movie, MPGS 4.1, and the SCRYU readers to NOT be available in the list.

```
Version 7.1
select Ensight 5
remove Movie
remove MPGS 4.1
remove SCRYU
```

10.6 Data Format Extension Map File Format

The ensight_reader_extension.map file is used to associate file suffixes with a reader within EnSight. This association allows EnSight to use the Quick load method of reading data. See Section 2.1, Reader Basics or How To Read Data

A default version of this file is provided with the EnSight distribution and is placed in the `$CEI_HOME/ensight92/site_preferences` directory. A user can override this file by placing his own version of the file in his EnSight Defaults Directory (located at %HOMEDRIVE%%HOMEPATH%\(username)\.ensight92 commonly located at C:\Users\username\.ensight92 on Vista and Win7, C:\Documents and Settings\yourusername\.ensight92 on older Windows, and ~/.ensight92 on Linux, and in ~/Library/Application Support/EnSight92 on the Mac).

A description of the format is contained in the file itself. A portion of the file is given below (it contains only a few file formats, but it should be easy to see how this file is formatted.

```
EnSight file extension to format association file
Version 1.0
#
# Comment lines start with a #
#
#
# The format of this file is as follows:
#
# READER_NAME: reader name as it appears in the Format chooser in the
#               EnSight Data Reader dialog
#
# NUM_FILE_1: the number of file_1_ext lines to follow
#
# FILE_1_EXT: the extension that follows a file name minus the ".", i.e.,
#              "geo", "case", etc.
#              There should be one definition after the :.  Multiple
#              FILE_1_EXT lines may exist
#
# NUM_FILE_2: the number of file_2_ext lines to follow
#
# FILE_2_EXT: the extension of a second file that will act as the result
#              file.  This is only used for formats that require two file
#              names.  As with FILE_1_EXT, there may be multiple
#              FILE_2_EXT lines.
#
# ELEMENT_REP: A key word that describes how the parts will be loaded
#               (all parts will be loaded the same way).
#               One of the following:
#                                 "3D border, 2D full"
#                                 "3D feature, 2D full"
#                                 "3D nonvisual, 2D full"
#                                 "Border"
#                                 "Feature angle"
#                                 "Bounding Box"
#                                 "Full"
#                                 "Non Visual"
```

```
#               If option is not set then 3D border, 2D full is used
#
# READ_BEFORE: (optional) The name of a command file to play before reading
#               the file(s)
#
# READ_AFTER: (optional) The name of a command file to read after loading
#              the parts

# Definition for Case files

READER_NAME: Case
NUM_FILE_1: 2
FILE_1_EXT: case
FILE_1_EXT: encas
ELEMENT_REP: 3D feature, 2D full

# Definition for EnSight5 files

READER_NAME: EnSight 5
NUM_FILE_1: 2
FILE_1_EXT: geo
FILE_1_EXT: GEOM
NUM_FILE_2: 2
FILE_2_EXT: res
FILE_2_EXT: RESULTS
ELEMENT_REP: 3D feature, 2D full

# Definition for STL files

READER_NAME: STL
NUM_FILE_1: 4
FILE_1_EXT: stl
FILE_1_EXT: STL
FILE_1_EXT: xct
FILE_1_EXT: XCT
ELEMENT_REP: 3D feature, 2D full

# Definition for Dytran files

READER_NAME: MSC/Dytran
NUM_FILE_1: 2
FILE_1_EXT: dat
FILE_1_EXT: ARC
ELEMENT_REP: 3D border, 2D full
READ_AFTER: ~/mylinuxpath/read_after_dytran.enc
#READ_AFTER: C:\mywindowspath\read_after_dytran.enc
```

10.7 Parallel Rendering Configuration File

The format of the configuration file for parallel rendering is described in detail in Section 13, Parallel and Distributed Rendering.

10.8 Resource File Format

Resources are used to specify which computers are used for running the various EnSight components, specifically the Server (ensight92.server), the SOS (ensight92.sos), the CollabHub (ensight92.collabhub), and the distributed renderers (ensight92.client). If you are running a single client and server on a single computer, you may skip this document.

Resources are an alternative way to specify these computers compared to SOS case files, PRDIST files, Connection Settings, and command line options. While these other ways are still valid and take precedence for backwards compatibility, resources greatly simplify specifying computers in a dynamic network environment. For example, SOS Case files and PRDIST files no longer need to be edited to reflect the current node allocation from cluster batch schedulers. Resources coupled with native reader support in the SOS even make SOS Case files unnecessary.

Resources can be specified via command line arguments and environment variables. Resources can be specified multiple times; precedence rules determine which resources ultimately get used. This allows sites to specify defaults while allowing those to be overridden.

Resources can be specified via a resource file. Here is an example of a resource file:

```
#!CEIResourceFile 1.0
SOS:
        host: localhost
SERVER:
        prologue: "setup_job"
        epilogue: "cleanup_job"
        host: server1
        host: server2
        host: server3
        host: server4

COLLABHUB:
        host: pc0

RENDERER:
        prologue: "setenv DISPLAY :0.0"
#       epilogue:
        host: pc1
        host: pc2
        host: pc2
```

Resource files must begin with the '`#!CEIResourceFile 1.0`' line. Afterwards, they may have up to four optional sections: `SOS`, `SERVER`, `COLLABHUB`, and `RENDERER`. Each of the four sections contains one or more '`host:  hostname`' lines. These lines specify which computers to use for the corresponding section. '`hostname`' must be an Internet/intranet routable host name or IP address. A given host name may appear on multiple lines within a section or in different sections. If it appears multiple times within a section, then that host will run multiple instances of the corresponding EnSight component if needed.

Additionally, each section may have an optional ‘`prologue: cmd`’ line and/or an optional ‘`epilogue: cmd`’ line. These specify a command to execute on each host before and after the corresponding EnSight component. Note that the cmd string must be quoted, and may include appropriate job backgrounding symbols (e.g. ‘&’).

For examples, see How To Use Resource Management.

11 EnSight Data Formats

This section describes the format for all readable and writable files in EnSight which you may need access to. The formats described are only for those files that are specific to EnSight. We do not describe data formats not developed by CEI (for example, data formats for various analysis codes). For information about these formats, consult the applicable creator.

Note: *If you are using this documentation to produce your own data translator, please make sure that you follow the instructions exactly as specified. In many cases, EnSight reads data in blocks to improve performance. If the format is not followed, the calculations of how much to read for a block will be thrown off. EnSight does little in the way of error checking data files when they are read. In this respect, EnSight sacrifices robustness for performance.*

EnSight Formats

EnSight has three evolutionary file formats listed below from oldest to most recent:

EnSight 5

- legacy format
- supported unstructured meshes only
- used a global nodal array
- used per node variables only

EnSight 6

- support for case file
- support for both unstructured and structured meshes
- uses a global nodal array
- use per node or per element variables

EnSight Case Gold (recommended format)

- is much faster than EnSight 6 and is more memory efficient (noticeable if you have a large number of parts or for larger models)
- uses connectivity which can be separate from the node ids
- uses a part basis rather than a global array

Format illustration

Ensight Case Gold	EnSight 6
part 1 node coordinates element connectivity by local node index **part 2** node coordinates element connectivity by local node index ... **part n** node coordinates element connectivity by local node index	global nodal ids & coordinates **part 1** element connectivity by global node ids **part 2** element connectivity by global node ids ... **part n** element connectivity by global node ids

Jump to Detailed Description of Formats

Saving Gold from EnSight

EnSight can export your model into Case Gold format (either ASCII or Binary). Activate the variables of interest, select the parts in the main part window and then go to the File menu: File->Save->Geometric Entities.

Tool to Check EnSight Format

There is another advantage to using Case Gold format in that there is a debugging tool called ens_checker that can help you find mistakes in files as you write a translator or exporter. Just type ens_checker file.case and the code will echo it's progress and the problems it finds to the console. This tool will also check EnSight 6 format.

Maximums to be Aware of:

There are some maximums that you should be aware of when producing EnSight data:

Maximum number of parts allowed	65000
Maximum number of variables allowed	10000
Maximum file name length	1024
Maximum part name length visible in the GUI	49
Maximum variable name length visible in the GUI	49

Detailed Description of Formats

Section 11.1, EnSight Gold Casefile Format describes in detail the EnSight Gold case, geometry, and variable file formats. ***This format provides the best performance for getting data into EnSight. It is readable by both EnSight and EnSight Gold.***

Section 11.2, EnSight6 Casefile Format describes in detail the EnSight6 case, geometry, and variable file formats. *This format is still supported, but is a legacy format. It has most, but not all, of the options of the gold format - and is somewhat lower in performance.*

Section 11.3, EnSight5 Format describes in detail the EnSight5 geometry and variable file formats. *This format is still supported, but is a legacy format. It is for unstructured data only.*

Section 11.4, FAST UNSTRUCTURED Results File Format describes the "executive" .res file that can be used with FAST unstructured solution and function files.

Section 11.5, FLUENT UNIVERSAL Results File Format describes the "executive" .res file used with FLUENT Universal files for transient models.

Section 11.6, Movie.BYU Results File Format describes the "executive" .res file that can be used with Movie.BYU files.

Section 11.7, PLOT3D Results File Format describes the "executive" .res file that can be used with PLOT3D solution and function files.

Section 11.8, Server-of-Server Casefile Format describes the format of the casefile used with the server-of-server capability of EnSight.

Section 11.9, Periodic Matchfile Format describes the file format used to explicitly specify which nodes match from one periodic instance to the next.

Section 11.10, XY Plot Data Format describes the XY plot dat file format.

Section 11.11, EnSight Boundary File Format describes the format of the file which can define unstructured boundaries of structured data.

Section 11.12, EnSight Particle Emitter File Format describes the format of the optional file containing particle trace emitter time and location information.

Section 11.13, EnSight Rigid Body File Format describes the format of the optional executive file that relates EnSight part names or numbers to rigid body transformations.

Section 11.14, Euler Parameter File Format describes the format of a file that contains translations and euler parameters for rigid body transformations.

Section 11.15, Vector Glyph File Format describes the format of the optional file containing vector information such as forces or moments.

Section 11.16, Constant Variables File Format describes the format of the constant variable file

Section 11.17, Point Part File Format describes the point part format.

Section 11.18, Spline Control Point File Format describes the spline file format.

Section 11.19, EnSight Embedded Python (EEP) File Format describes a portable delivery mechanism for data and scripts.

Section 11.20, Camera Orientation File Format describes the file format for positioning and orienting the Camera.

11.1 EnSight Gold Casefile Format

Include in this section:

EnSight Gold General Description

EnSight Gold data consists of the following files:

- Case (required) (points to all other needed files including model geometry, variables, and possibly measured geometry and variables - as well as optionally a periodic match file, a structured boundary file, and a rigid body file)

EnSight makes no assumptions regarding the physical significance of the scalar, vector, 2nd order symmetric tensor, and complex variables. These files can be from any discipline. For example, the scalar file can include such things as pressure, temperature, and stress. The vector file can be velocity, displacement, or any other vector data, etc.

In addition, EnSight Gold format handles "undefined" as well as "partial" variable values. (See appropriate subsections later in this chapter for details.)

All variable results for EnSight Gold format are contained in disk files—one variable per file. Additionally, if there are multiple time steps, there must either be a set of disk files for each time step (transient multiple-file format), or all time steps of a particular variable or geometry in one disk file each (transient single-file format).

Sources of EnSight Gold format data include the following:

- Data that can be translated to conform to the EnSight Gold data format (including being written from EnSight itself using the Save Geometric Entities option under File->Save)
- Data that originates from one of the translators supplied with the EnSight application

The EnSight Gold format supports an unstructured defined element set as shown in the figure on a following page. Unstructured data must be defined in this element set. Elements that do not conform to this set must either be subdivided or discarded.

The EnSight Gold format also supports the same structured block data format as EnSight6, which is very similar to the PLOT3D format. *Note that for this format, the standard order of nodes is such that I's advance quickest, followed by J's, and then K's.*

A given EnSight Gold model may have either unstructured data, structured data, or a mixture of both.

This format is somewhat similar to the EnSight6 format, but differs enough to allow for more efficient reading of the data. It is intended for **3D, binary, big** data models.

Note: While an ASCII format is available, it is not intended for use with large models and is in fact subject to limitations such as integer lengths of 10 digits. Use the binary format if your model will exceed 10 digits for node or element numbers or labels.

Starting with version 7, EnSight writes out all model and variable files in EnSight Gold format. Thus, it can be read by all version 7 or 8 EnSight licenses (i.e. standard, gold, and custom licenses).

ens_checker

A program is supplied with EnSight which attempts to verify the integrity of the *format* of EnSight 6 and EnSight Gold files. If you are producing EnSight formatted data, this program can be very helpful, especially in your development stage, in making sure that you are adhering to the published format. It makes no attempt to verify the validity of floating point values, such as coordinates, variable values, etc. This program takes a casefile as input. Thus, it will check the format of the casefile, and all associated geometry and variable files referenced in the casefile. See How To Use ens_checker.

Supported EnSight Gold Elements

The elements that are supported by the EnSight Gold format are:

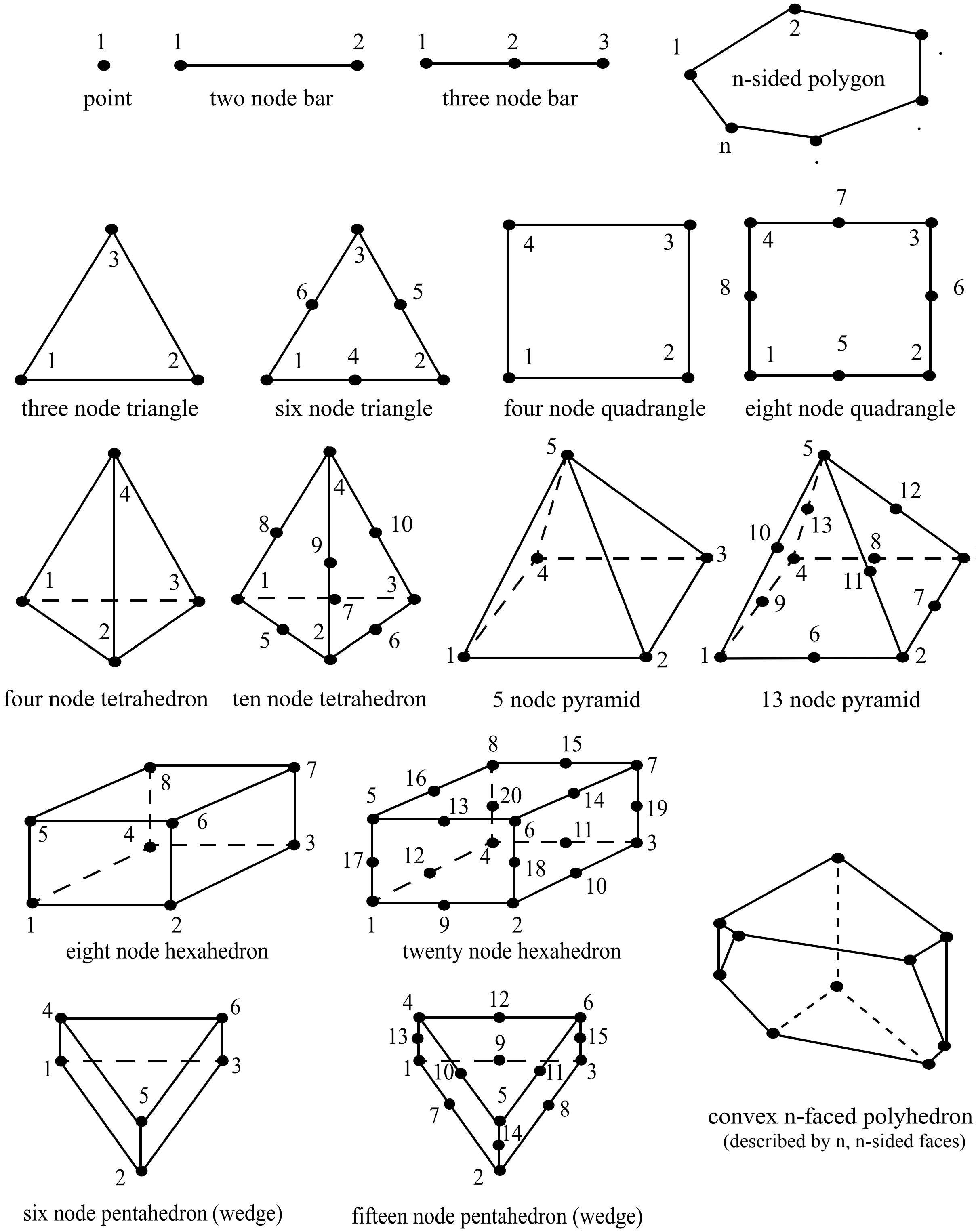


Figure 11-1
Supported EnSight Gold Elements

EnSight Gold Case File Format

The Case file is an ASCII free format file that contains all the file and name information for accessing model (and measured) geometry, variable, and time information. It is comprised of five sections (`FORMAT, GEOMETRY, VARIABLE, TIME, FILE`) as described below:

Notes: *If the case file name contains spaces it should load fine in the browser, but from the command line it must be in quotes (e.g. ensight92 -case "file name.case") or on linux or mac can also use the backslash character ("\") as follows: ensight92 -case file\ name.case).*

All lines in the Case file are limited to 1024 characters.

The titles of each section must be in all capital letters.

Anything preceded by a "#" denotes a comment and is ignored. Comments may append information lines or be placed on their own lines.

Information following ":" may be separated by white spaces or tabs.

Specifications encased in "[]" are optional, as indicated.

Format Section

This is a required section which specifies the type of data to be read.

Usage:

```
FORMAT
type:     ensight gold
```

Geometry Section

This is a required section which specifies the geometry information for the model (as well as measured geometry if present, periodic match file (see Section 11.9, Periodic Matchfile Format) if present, boundary file (see Section 11.11, EnSight Boundary File Format) if present, rigid body file (see Section 11.13, EnSight Rigid Body File Format) if present, and vector glyphs file (see Section 11.15, Vector Glyph File Format) if present).

Usage:

```
GEOMETRY
model:       [ts]  [fs]        filename    [change_coords_only [cstep]]
measured:    [ts]  [fs]        filename    [change_coords_only]
match:                         filename    [add_ghosts]
boundary:                      filename
rigid_body:                    filename
Vector_glyphs:                 filename
```

where: `ts` = time set number as specified in TIME section. This is optional.

`fs` = corresponding file set number as specified in FILE section below.

(Note, if you specify `fs`, then `ts` is no longer optional and must also be specified.)

`filename` = The filename of the appropriate file.

-> Model or measured filenames for a static geometry case (or single file format), as well as match, boundary, and rigid_body filenames will <u>not</u> contain "*" wildcards.

-> Model or measured filenames for a changing geometry case (unless single file format) will contain "*" wildcards.

-> Model filenames for the structured block continuation option will contain "%" wildcards.

-> filenames with spaces in them need to be enclosed in quotes.

`change_coords_only` = The option to indicate that the changing geometry (as indicated by wildcards in the filename) is coords only. Otherwise, changing geometry connectivity will be assumed.

cstep = the zero-based time step which contains the connectivity - only used for change_coords_only option. This is an optional parameter. If all time steps have the connectivity, then this is not needed and can be omitted. But if used, the other time steps do not need to contain the connectivity - the parts need only contain the coordinates, which can save considerably on file size.

add_ghosts EnSight Case Gold allows the optional [add_ghosts] parameter after the filename which will produce ghost cells across the match file boundary which will provide continuity for variable calculations across the boundary, and for computational symmetry/mirroring, etc. Only use the add_ghosts option if you can afford the penalty of the additonal ghost cells and you need the computational continuity that they provide.

Note: *It is possible to use EnSight 5 measured data with a casefile. This is done by using the Measured: line in the GEOMETRY section without any of the optional portions, and with filename being an EnSight 5 measured results file (which typically has a .mea extension). Also, since such information is contained in the .mea file, do not use any measured variable lines in the VARIABLE section.*

Variable Section

This is an optional section which specifies the files and names of the variables (max number of variables is 300). Constant variable values can also be set in this section.

Usage:

```
VARIABLE
constant per case:            [ts]        description  const_value(s)
constant per case file:       [ts]        description  cvfilename
scalar per node:              [ts] [fs]   description  filename
vector per node:              [ts] [fs]   description  filename
tensor symm per node:         [ts] [fs]   description  filename
tensor asym per node:         [ts] [fs]   description  filename
scalar per element:           [ts] [fs]   description  filename
vector per element:           [ts] [fs]   description  filename
tensor symm per element:      [ts] [fs]   description  filename
tensor asym per element:      [ts] [fs]   description  filename
scalar per measured node:     [ts] [fs]   description  filename
vector per measured node:     [ts] [fs]   description  filename
complex scalar per node:      [ts] [fs]   description  Re_fn   Im_fn   freq
complex vector per node:      [ts] [fs]   description  Re_fn   Im_fn   freq

complex scalar per element:     [ts] [fs] description  Re_fn   Im_fn   freq
complex vector per element:     [ts] [fs] description  Re_fn   Im_fn   freq
```

where:

`ts` = The corresponding time set number (or index) as specified in TIME section below. This is only required for transient constants and variables.

`fs` = The corresponding file set number (or index) as specified in FILE section below.
(Note, if you specify `fs`, then `ts` is no longer optional and must also be specified.)

`description` = The variable (GUI) name (ex. Pressure, Velocity, etc.)If the variable name contains a space, it must be in quotes, and will be renamed in EnSight, replacing the spaces with an underscore.

`const_value(s)` = The constant value. If constants change over time, then `ns` (see TIME section below) constant values of `ts`.

`cvfilename`	=	The filename containing the constant values, one value per time step.
`filename`	=	The filename of the variable file. Note: only transient filenames contain "*" wildcards. If the filename contains a space, it must be in quotes.
`Re_fn`	=	The filename for the file containing the real values of the complex variable.
`Im_fn`	=	The filename for the file containing the imaginary values of the complex variable.
`freq`	=	The corresponding harmonic frequency of the complex variable. For complex variables where harmonic frequency is undefined, simply use the text string: UNDEFINED.

Note: *As many variable description lines as needed may be used.*

Note: *Variable descriptions have the following restrictions:*

The maximum variable name length is documented at the beginning of this chapter.

Duplicate variable descriptions are not allowed.

Leading and trailing white space will be eliminated.

Variable descriptions must not start with a numeric digit.

Variable descriptions must not contain any of the following reserved characters:

(	[	+	@	!	*	$	:	,
)	]	-	space	#	^	/	.	

Note: *scalar or vector per measured node is necessary for EnSight Gold or EnSight 6 measured format data. For EnSight 5 format measured data, only the results file (typically suffix .mea) is necessary in the geometry section because the EnSight 5 results file describes the geometry and variable files.*

Time Section

This is an optional section for steady state cases, but is required for transient cases. It contains time set information. Shown below is information for one time set. Multiple time sets (up to 16) may be specified for measured data as shown in Case File Example 3 below.

Usage:

```
TIME
time set:             ts  [description]
number of steps:      ns
filename start number:fs
filename increment:   fi
time values:          time_1  time_2  .... time_ns
```

or

```
TIME
time set:             ts  [description]
number of steps:      ns
filename numbers:     fn
time values:          time_1  time_2  .... time_ns
```

or

```
TIME
time set:             ts  [description]
number of steps:      ns
```

```
filename numbers file:fnfilename
time values file:     tvfilename
```

where: `ts` = timeset number. This is the number referenced in the GEOMETRY and VARIABLE sections.
`description` = optional timeset description which will be shown in user interface.
`ns` = number of transient steps
`fs` = the number to replace the "*" wildcards in the filenames, for the first step
`fi` = the increment to fs for subsequent steps
`time` = the actual time values for each step, each of which must be separated by a white space and which may continue on the next line if needed
`fn` = a list of numbers or indices, to replace the "*" wildcards in the filenames.
`fnfilename` = name of file containing ns filename numbers (fn).
`tvfilename` = name of file containing the time values(time_1 ... time_ns).

File Section

This section is optional for expressing a transient case with single-file formats. This section contains single-file set information. This information specifies the number of time steps in each file of each *data entity*, i.e. each geometry and each variable (model and/or measured). Each data entity's corresponding file set might have multiple *continuation* files due to system file size limit, i.e. ~2 GB for 32-bit and ~4 TB for 64-bit architectures. **Each file set corresponds to one and only one time set, but a time set may be referenced by many file sets.** The following information may be specified in each file set. For file sets where all of the time set data exceeds the maximum file size limit of the system, both `filename index` and `number of steps` are repeated within the file set definition for each continuation file required. Otherwise `filename index` may be omitted if there is only one file. File set information is shown in Case File Example 4 below.

Usage:

```
FILE
file set:             fs
filename index:       fi  # Note: only used when data continues in other files
number of steps:      ns
```

where: `fs` = file set number. This is the number referenced in the GEOMETRY and VARIABLE sections above.
`ns` = number of transient steps
`fi` = file index number in the file name (replaces "*" in the filenames)

Material Section

This is an optional section for material set information in the material interface part case. For more details see the description in the Material Parts Create/Update Feature Detail Editor, or the MatSpecies calculator function in 4.3. Shown below is the format for one material set. ***(Note, currently only one material set is supported.)*** An example of this material set information is appended below as EnSight Gold Material Files Format.

Usage:

```
MATERIAL
material set number:          ms [description]
material id count:            nm
material id numbers:          matno_1  matno_2  ... matno_nm
material id names:            matdesc_1 mat_2 ... mat_nm
```

 # Either sparse file specifications:

```
material id per element       [ts] [fs] filename
material mixed ids:           [ts] [fs] filename
material mixed values:        [ts] [fs] filename
# optional species parameters with sparse file specifications only
```

species id count:ns

```
species id numbers:           spno_1 spno_2 … spno_ns
species id names:             spdesc_1 spdesc_2 … spdesc_ns
species per material counts:  spm_1 spm_2 … spm_nm
species per material lists:   matno_1_sp_1 matno_1_sp_2 … matno_1_sp_spm_1
                              matno_2_sp_1 matno_2_sp_2 … matno_2_sp_spm_2
                              …
                              matno_nm_sp_1 matno_nm_sp2 … matno_nm_sp_spm_nm
                              (Note: above concatenated lists do not have to be on
                              separate lines.)
species element values:       [ts] [fs] sp_filename
```

 # Or materials defined by per element scalar variables:

material scalars per element: desc_esv_1 desc_esv_2 ... desc_esv_nm

where:

ts = The corresponding time set number (or index) as specified in TIME section above. This is only required for transient materials.

fs = The corresponding file set number (or index) as specified in FILE section above. (Note, if you specify fs, then ts is no longer optional and must also be specified.)

ms = Material set number. (Note, currently there is only one, and it must be a positive number.)

description = Optional material set description which will be reflected in the file names of exported material files.

nm = Number of materials for this set.

matno = Material number used in the material and mixed-material id files. There should be nm of these. Non-positive numbers are grouped as the “null material”. See EnSight Gold Material Files Format

matdesc = GUI material description corresponding to the nm matno’s.

filename = The filename of the appropriate material file. Note, only transient filenames contain “*” wildcards. The three required files are the material id per element, the mixed-material ids, and the mixed-material values files.

ns = Number of species for this set.

spno = Specie number used in the "species per material lists:" specification. There should be ns of these positive integers.

spdesc = GUI specie description corresponding to the ns spno's.

spm = Number of species per material number (matno). Enter 0 if no species exist for a material.

matno_#_sp = Specie id number (spno) list member for this material number id (matno). If no species for this material, then proceed to next material that has species.

sp_filename = The filename of the appropriate “species element values:” file. Note, only transient filenames contain "*" wildcards.

desc_esv = The description of each per element scalar variable to be a material. The description listed must match the description listed under the VARIABLE

```
section for the per element scalar variable.
```

Note: *Material and species descriptions are limited to 19 characters in the current release. Material and species descriptions and file names must not start with a numeric digit and must not contain any of the following reserved characters:*

```
( [ + @ ! * $
) ] - # ^ / space
```

Block Continuation Section

This section is optional for grouping partitioned structured blocks together. The files containing blocks must conform to some restrictions in order for this to be possible. Namely, the blocks in the files must be a continuation (in one of the directions) from the blocks in the previous file. This purpose for this capability is to be able to read N number of files using M number of cluster nodes, where N > M. The filenames for the geometry and variables must contain "%" wildcards if this option is used.

Usage:

```
BLOCK_CONTINUATION
number of sets:       ns
filename start number:fs
filename increment:   fi
```

where: `ns` = The number of contiguous partitioned structured files to use
`fs` = the number to replace the "%" wildcards in the geometry and variable filenames, for the first set
`fi` = the increment to `fs` for subsequent sets

Scripts Section

This is an optional section which specifies the name of a Python script file or an XML metadata file. The Python file is read when the dataset is loaded. The script file contents are transferred to the client where they are executed in the running client Python interpreter. The geometry pathname will be prepended to the Python script filename before it is read if a fully qualified pathname is not provided.

The XML metadata file is used internal to EnSight.
Usage:

```
SCRIPTS
python:   filename.py
metadata: filename.xml
```

Case File Example 1

The following is a minimal EnSight Gold case file for a steady state model with some results.

Note: this (`engold.case`*) file, as well as all of its referenced geometry and variable files (along with a couple of command files) can be found under your installation directory (path:* `$CEI_HOME/ensight92/data/user_manual`*). The EnSight Gold Geometry File Example and the Variable File Examples are the contents of these files.*

```
FORMAT
type:  ensight gold

GEOMETRY
model:  engold.geo

VARIABLE
```

```
constant per case:             Cden   .8

scalar per element:            Esca   engold.Esca
scalar per node:               Nsca   engold.Nsca

vector per element:            Evec   engold.Evec
vector per node:               Nvec   engold.Nvec

tensor symm per element:       Eten   engold.Eten
tensor symm per node:          Nten   engold.Nten

complex scalar per element:    Ecmp   engold.Ecmp_rengold.Ecmp_i2.
complex scalar per node:       Ncmp   engold.Ncmp_rengold.Ncmp_i4.
```

Case File Example 2

The following is a Case file for a transient model. The connectivity of the geometry is also changing.

```
FORMAT
type:  ensight gold

GEOMETRY
model:                   1                      exgold2.geo**

VARIABLE
scalar per node:         1     Stress           exgold2.scl**
vector per node:         1     Displacement     exgold2.dis**

TIME
time set:                      1
number of steps:               3
filename start number:         0
filename increment:            1
time values:       1.0    2.0  3.0
```

The following files would be needed for Example 2:

```
exgold2.geo00        exgold2.scl00        exgold2.dis00
exgold2.geo01        exgold2.scl01        exgold2.dis01
exgold2.geo02        exgold2.scl02        exgold2.dis02
```

Case File Example 3

The following is a Case file for a transient model with measured data.

This example has pressure given per element.

```
FORMAT
type:  ensight gold


GEOMETRY
model:                  1                  exgold3.geo*
measured:               2                  exgold3.mgeo**


VARIABLE
constant per case:                         Gamma        1.4
constant per case:                   1     Density      .9 .9 .7 .6 .6
scalar per element                   1     Pressure           exgold3.pre*
vector per node:                     1     Velocity           exgold3.vel*
scalar per measured node:            2     Temperature        exgold3.mtem**
vector per measured node:            2     Velocity           exgold3.mvel**

TIME
time set:                      1
number of steps:               5
filename start number:         1
filename increment:            2
time values:                   .1 .2 .3           # This example shows that time
                               .4 .5              # values can be on multiple lines
time set:                      2
number of steps:               6
filename start number:         0
filename increment:            2
time values:
.05 .15 .25 .34 .45 .55
```

The following files would be needed for Example 3:

```
exgold3.geo1          exgold3.pre1          exgold3.vel1
exgold3.geo3          exgold3.pre3          exgold3.vel3
exgold3.geo5          exgold3.pre5          exgold3.vel5
exgold3.geo7          exgold3.pre7          exgold3.vel7
exgold3.geo9          exgold3.pre9          exgold3.vel9

exgold3.mgeo00        exgold3.mtem00        exgold3.mvel00
exgold3.mgeo02        exgold3.mtem02        exgold3.mvel02
exgold3.mgeo04        exgold3.mtem04        exgold3.mvel04
exgold3.mgeo06        exgold3.mtem06        exgold3.mvel06
exgold3.mgeo08        exgold3.mtem08        exgold3.mvel08
exgold3.mgeo10        exgold3.mtem10        exgold3.mvel10
```

Case File Example 4

The following is two Case files for a simple static Block Continuation structured model containing 5 files (sets). The first uses the first two sets, the second uses the last three sets.

```
FORMAT
type:  ensight gold
GEOMETRY
model:                                          ex_bc_%.geo
VARIABLE
scalar per node:                temperature     ex_bc_%.scl
BLOCK_CONTINUATION
number of sets:                 2
filename start number:          1
filename increment:             1
```

```
FORMAT
type:  ensight gold
GEOMETRY
model:                                          ex_bc_%.geo
VARIABLE
scalar per node:                temperature     ex_bc_%.scl
BLOCK_CONTINUATION
number of sets:                 3
filename start number:          3
filename increment:             1
```

The following files would be needed for Example 4:

```
ex_bc_1.geo        ex_bc_1.scl     used by first case
ex_bc_2.geo        ex_bc_2.scl     used by first case
ex_bc_3.geo        ex_bc_3.scl     used by second case
ex_bc_4.geo        ex_bc_4.scl     used by second case
ex_bc_5.geo        ex_bc_5.scl     used by second case
```

Case File Example 5

The following is Case File Example 3 expressed in transient single-file formats.

In this example, the transient data for the measured velocity data entity happens to be greater than the maximum file size limit. Therefore, the first four time steps fit and are contained in the first file, and the last two time steps are 'continued' in a second file.

```
FORMAT
type: ensight gold

GEOMETRY
model:                  1     1     exgold5.geo
measured:               2     2     exgold5.mgeo

VARIABLE
constant per case:                        Density          .5
scalar per element:           1     1     Pressure         exgold5.pre
vector per node:              1     1     Velocity         exgold5.vel
scalar per measured node:     2     2     Temperature      exgold5.mtem
vector per measured node:     2     3     Velocity         exgold5.mvel*

TIME
time set:               1     Model
number of steps:        5
time values:            .1 .2 .3 .4 .5

time set:               2     Measured
number of steps:        6
time values:            .05 .15 .25 .34 .45 .55

FILE
file set:               1
```

```
number of steps:        5

file set:               2
number of steps:        6

file set:               3
filename index:         1
number of steps:        4
filename index:         2
number of steps:        2
```

The following files would be needed for Example 5:

```
exgold5.geo     exgold5.pre     exgold5.vel
exgold5.mgeo    exgold5.mtem    exgold5.mvel1
                                exgold5.mvel2
```

Case File Example 6

The following is a Case file for a transient model. The connectivity of the geometry is not changing, but the coordinates are. The connectivity is only present in step 1, but is not present in steps 0, 2, 3, 4, or 5).

```
FORMAT
type: ensight gold
GEOMETRY
model:                  1     aaa_coords.geo**  change_coords_only  1
TIME
time set:                     1
number of steps:              6
filename start number:        0
filename increment:           1
time values:                  0.1 1.2 2.3 3.4 4.5 5.6
```

The following files would be needed for Example 6:

```
aaa_coords.geo00
aaa_coords.geo01    (contains the connectivity)
aaa_coords.geo02
aaa_coords.geo03
aaa_coords.geo04
aaa_coords.geo05
```

Contents of Transient Single Files

Each file contains transient data that corresponds to the specified number of time steps. The data for each time step sequentially corresponds to the simulation time values (`time values`) found listed in the `TIME` section. In transient single-file format, the data for each time step essentially corresponds to a standard EnSight gold geometry or variable file (model or measured) as expressed in multiple file format. The data for each time step is enclosed between two *wrapper* records, i.e. preceded by a `BEGIN TIME STEP` record and followed by an `END TIME STEP` record. Time step data is not split between files. If there is not enough room to append the data from a time step to the file without exceeding the maximum file limit of a particular system, then a continuation file must be created for the time step data and any subsequent time step. Any type of user comments may be included before and/or after each transient step wrapper.

*Note 1: If transient single file format is used, EnSight expects **all** files of a dataset to be specified in transient single file format. Thus, even static files must be enclosed between a* `BEGIN TIME STEP` *and an* `END TIME STEP` *wrapper. This includes the condition where you have transient variables with static geometry. The static geometry file must have the wrapper.*

1. *Note 2: For binary geometry files, the first* `BEGIN TIME STEP` *wrapper must follow the <C Binary/Fortran Binary> line. Both* `BEGIN TIME STEP` *and* `END TIME STEP` *wrappers are written according to type (1) in binary.*

Namely: This is a write of 80 characters to the file:

in C:
```
char buffer[80];
strcpy(buffer,"BEGIN TIME STEP");
fwrite(buffer,sizeof(char),80,file_ptr);
```

in FORTRAN:
```
character*80 buffer
buffer = "BEGIN TIME STEP"
```

*Note 3: Efficient reading of each file (especially binary) is facilitated by appending each file with a **file index**. A file index contains appropriate information to access the file byte positions of each time step in the file. (EnSight automatically appends a file index to each file when exporting in transient single file format.) If used, the file index must follow the last* `END TIME STEP` *wrapper in each file.*

File Index Usage:

ASCII	Binary	Item	Description
"%20d\n"	sizeof(int)	n	Total number of data time steps in the file.
"%20d\n"	sizeof(long)	fb_1	File byte loc for contents of 1st time step*
"%20d\n"	sizeof(long)	fb_2	File byte loc for contents of 2nd time step*
. . .	. . .	. . .	. . .
"%20d\n"	sizeof(long)	fb_n	File byte loc for contents of nth time step*
"%20d\n"	sizeof(int)	flag	Miscellaneous flag (= 0 for now)
"%20d\n"	sizeof(long)	fb of item n	File byte loc for Item n above
"%s\n"	sizeof(char)*80	"FILE_INDEX"	File index keyword

* *Each file byte location is the first byte that follows the* `BEGIN TIME STEP` *record.*

Shown below are the contents of each of the above files, using the data files from Case File Example 3 for reference (without FILE_INDEX for simplicity).

Contents of file `exgold4.geo_1`:

BEGIN TIME STEP
Contents of file `exgold3.geo1`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.geo3`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.geo5`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.geo7`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.geo9`
END TIME STEP

Contents of file `exgold4.pre_1`:

BEGIN TIME STEP
Contents of file `exgold3.pre1`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.pre3`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.pre5`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.pre7`
END TIME STEP
BEGIN TIME STEP
Contents of file `exgold3.pre9`
END TIME STEP

Contents of file `exgold4.vel_1`:

BEGIN TIME STEP
Contents of file `exgold3.vel1`

```
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.vel3
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.vel5
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.vel7
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.vel9
END TIME STEP
```

Contents of file `exgold4.mgeo_1`:

```
BEGIN TIME STEP
Contents of file exgold3.mgeo00
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mgeo02
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mgeo04
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mgeo06
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mgeo08
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mgeo10
END TIME STEP
```

Contents of file `exgold4.mtem_1:`

```
BEGIN TIME STEP
Contents of file exgold3.mtem00
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mtem02
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mtem04
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mtem06
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mtem08
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mtem10
END TIME STEP
```

Contents of file `exgold4.mvel1_1:`

```
BEGIN TIME STEP
Contents of file exgold3.mvel00
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mvel02
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mvel04
END TIME STEP
BEGIN TIME STEP
Contents of file exgold3.mvel06
END TIME STEP
```

Contents of file `exgold4.mvel2_1:`

```
Comments can precede the beginning wrapper here.
BEGIN TIME STEP
Contents of file exgold3.mvel08
END TIME STEP
Comments can go between time step wrappers here.
BEGIN TIME STEP
Contents of file exgold3.mvel10
END TIME STEP
Comments can follow the ending time step wrapper.
```

Note: Each of these files could (and should for efficiency reasons) have the FILE_INDEX information following the last END TIMESTEP. See the previous discussion for its usage.

EnSight Gold Wild Card Name Specification

If multiple time steps are involved, the file names must conform to the EnSight wild-card specification. This specification is as follows:

- File names must include numbers that are in ascending order from beginning to end.
- Numbers in the files names must be zero filled if there is more than one significant digit.
- Numbers can be anywhere in the file name.
- When the file name is specified in the EnSight case file, you must replace the numbers in the file with an asterisk(*). The number of asterisks specified is the number of significant digits. The asterisk must occupy the same place as the numbers in the file names.

EnSight Gold Geometry File Format

The EnSight Gold format is part based for both unstructured and structured data. There is no global coordinate array that each part references, but instead - each part contains its own local coordinate array. Thus, the node numbers in element connectivities refer to the coordinate array index, not a node id or label. *This is different than the EnSight6 format!*

The EnSight Gold format consists of keywords followed by information. The following items are important when working with EnSight Gold geometry files:

1. Node ids are optional. In this format they are strictly labels and are not used in the connectivity definition. The element connectivities are based on the local implied node number of the coordinate array in each part, which is sequential starting at one. If you let EnSight assign node IDs, this implied internal numbering is used. If node IDs are set to off, they are numbered internally, however, you will not be able to display or query on them. If you have node IDs given in your data, you can have EnSight ignore them by specifying "node id ignore." Using this option may reduce some of the memory taken up by the Client and Server, but display and query on the nodes will not be available. Note, prior to EnSight 7.4, node ids could only be specified for unstructured parts. This restriction has been removed and user specified node ids are now possible for structured parts.
2. Element ids are optional. If you specify element IDs, or you let EnSight assign them, you can show them on the screen. If they are set to off, you will not be able to show or query on them. If you have element IDs given in your data you can have EnSight ignore them by specifying "element id ignore." Using this option will reduce some of the memory taken up by the Client and Server. This may or may not be a significant amount, and remember that display and query on the elements will not be available. Note, prior to EnSight 7.4, element ids could only be specified for unstructured parts. This restriction has been removed and user specified element ids are now possible for structured parts.
3. Model extents can be defined in the file so EnSight will not have to determine

these while reading in data. If they are not included, EnSight will compute them, but will not actually do so until a dataset query is performed the first time.

4. The format of integers and real numbers **must be followed** (See the Geometry Example below).

5. ASCII Integers are written out using the following integer format:

   From C: `10d` format

   From FORTRAN: `i10` format

   Note: this size of integer format limits the number of nodes as well as node and element labels to 2^{31} (2 GB) per part which is the same as a 32-bit integer.

   ASCII Real numbers are written out using the following floating-point format:

   From C: `12.5e` format

   From FORTRAN: `e12.5` format

   The number of integers or reals per line must also be followed!

6. By default, a Part is processed to show the outside boundaries. This representation is loaded to the Client host system when the geometry file is read (unless other attributes have been set on the workstation, such as feature angle).

7. Coordinates for unstructured data must be defined within each part. This is normally done before any elements are defined within a part, but does not have to be. The different elements can be defined in any order (that is, you can define a hexa8 before a bar2).

8. A Part containing structured data cannot contain any unstructured element types or more than one block. **Each structured Part is limited to a single block (or some subset of that block).** A structured block is indicated by following the Part description line with a 'block' line. By default, a block will be curvilinear, non-iblanked, non-ghost, complete range. However, by suppling one or more of the following options on the 'block' line, rectilinear or uniform blocks can be specified, nodal iblanking for the block can be used, cells within the block can be flagged as ghosts (used for computations, but not displayed), subset ranges can be specified (useful for partitioned data). The options include:

Only one of these can be used on the 'block' line	
`curvilinear`	Indicates that coordinates of all ijk locations of the block will be specified (default)
`rectilinear`	Indicates that i,j,k delta vectors for a regular block with possible non-regular spacing will be specified
`uniform`	Indicates that i,j,k delta values for a regular block with regular spacing will be specified
Any, none, or all of these can be used	

<table>
<tr><td rowspan="5"><code>iblanked</code></td><td colspan="2">An “iblanked” block must contain an additional integer array of values at each node, traditionally called the iblank array. Valid iblank values for the EnSight Gold format are:</td></tr>
<tr><td>0</td><td>for nodes which are exterior to the model, sometimes called blanked-out nodes</td></tr>
<tr><td>1</td><td>for nodes which are interior to the model, thus in the free stream and to be used</td></tr>
<tr><td><0 or >1</td><td>for any kind of boundary nodes</td></tr>
<tr><td colspan="2">In EnSight’s structured Part building dialog, the iblank option selected will control which portion of the structured block is “created”. Thus, from the same structured block, the interior flow field part as well as a symmetry boundary part could be “created”.

Note: By default EnSight does not do any “partial” cell iblank processing. Namely, only complete cells containing no “exterior” nodes are created. It is possible to obtain partial cell processing by issuing the “test:partial_cells_on” command in the Command Dialog before reading the file.</td></tr>
<tr><td><code>with_ghost</code></td><td colspan="2">A block with ghosts must contain an additional integer array of flags for each cell. A flag value of zero indicates a non-ghost cell. A flag value of non-zero indicates a ghost cell.</td></tr>
</table>

`range`	A block with ranges will contain an extra line, following the ijk line, which gives min and max planes for each of the ijk directions.
	
	Thus, normally a 6 x 5 x 1 block part would start something like:
	
	part
	        1
	description
	block
	        6 5 1
	 0.00000e+00
	...
	(The coordinate information for the 30 nodes of the block must follow.)
	
	But if only the top 6 x 3 x 1portion was to be represented in the file, you can use “range” like:
	
	part
	        1
	description for top only
	block range
	        6 5 1
	        1 6 3 5 1 1
	 0.00000e+00
	...
	(The coordinate information for the 18 nodes of the top portion of the block must follow. Note that the ijk line following the block line contains the size of the original block - which is needed to properly deal with node and element numbering. The next line contains the imin, imax, jmin, jmax, kmin, kmax defining the subset ranges. The actual size of the block being defined is thus computed from these ranges:
	size_i = imax - imin + 1
	size_j = jmax - jmin + 1
	size_k = kmax - kmin + 1

Note that for structured data, the standard order of nodes is such that I’s advance quickest, followed by J’s, and then K’s.

9. Maximum number of parts is 65000
   Maximum number of variables is 10000
   Maximum file name length is 1024
   Maximum part name length is 79, but the GUI will only display 49
   Maximum variable name length is 79, but the GUI will only display 49

Generic Format

Usage Notes:

In general an unstructured part can contain several different element types.

element type can be any of:

point	g_point
bar2	g_bar2
bar3	g_bar3
tria3	g_tria3
tria6	g_tria6
quad4	g_quad4
quad8	g_quad8
tetra4	g_tetra4
tetra10	g_tetra10
pyramid5	g_pyramid5
pyramid13	g_pyramid13
penta6	g_penta6
penta15	g_penta15
hexa8	g_hexa8
hexa20	g_hexa20
nsided	g_nsided
nfaced	g_nfaced

# = a part number (maximum is 65000)
nn = total number of nodes in a part
ne = number of elements of a given type
np = number of nodes per element for a given element type
nf = number of faces per nfaced element
id_* = node or element id number
x_* = x component
y_* = y component
z_* = z component
n*_e* = node number for an element
f*_e* = face number for an nfaced element
ib_* = iblanking value
gf_e*= ghost flag for a structured cell

[ ] contain optional portions
< > contain choices
` indicates the beginning of an unformatted sequential FORTRAN binary write
' indicates the end of an unformatted sequential FORTRAN binary write

C Binary form:

```
C Binary                                                  80 chars
description line 1                                        80 chars
description line 2                                        80 chars
node id <off/given/assign/ignore>                         80 chars
element id <off/given/assign/ignore>                      80 chars
[extents                                                  80 chars
xmin xmax ymin ymax zmin zmax]                             6 floats
part                                                      80 chars
#                                                          1 int
description line                                          80 chars
coordinates                                               80 chars
nn                                                         1 int
[id_n1 id_n2 ... id_nn]                                   nn ints
x_n1 x_n2 ... x_nn                                        nn floats
y_n1 y_n2 ... y_nn                                        nn floats
z_n1 z_n2 ... z_nn                                        nn floats
element type                                              80 chars
ne                                                         1 int
[id_e1 id_e2 ... id_ne]                                   ne ints
n1_e1 n2_e1 ... np_e1
n1_e2 n2_e2 ... np_e2
 .
 .
n1_ne n2_ne ... np_ne                                     ne*np ints
element type                                              80 chars
 .
 .
part                                                      80 chars
 .
 .
part                                                      80 chars
#                                                          1 int
description line                                          80 chars
block [iblanked] [with_ghost] [range]                     80 chars
i j k             # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)     3 ints
[imin imax jmin jmax kmin kmax]  # if range used:          6 ints
                                  nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                                       ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
x_n1 x_n2 ... x_nn                                        nn floats
y_n1 y_n2 ... y_nn                                        nn floats
z_n1 z_n2 ... z_nn                                        nn floats
[ib_n1 ib_n2 ... ib_nn]                                   nn ints
[ghost_flags]                                             80 chars
[gf_e1 gf_e2 ... gf_ne]                                   ne ints
[node_ids]                                                80 chars
[id_n1 id_n2 ... id_nn]                                   nn ints
[element_ids]                                             80 chars
[id_e1 id_e2 ... id_ne]                                   ne ints
part                                                      80 chars
#                                                          1 int
description line                                          80 chars
block rectilinear [iblanked] [with_ghost] [range]         80 chars
i j k             # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)    3 ints
[imin imax jmin jmax kmin kmax]  # if range used:          6 ints
                                  nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                                       ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
x_1 x_2 ... x_i                                            i floats
y_1 y_2 ... y_j                                            j floats
```

```
z_1 z_2 ... z_k                                           k floats
[ib_n1 ib_n2 ... ib_nn]                                  nn ints
[ghost_flags]                                            80 chars
[gf_e1 gf_e2 ... gf_ne]                                  ne ints
[node_ids]                                               80 chars
[id_n1 id_n2 ... id_nn]                                  nn ints
[element_ids]                                            80 chars
[id_e1 id_e2 ... id_ne]                                  ne ints
part                                                     80 chars
#                                                         1 int
description line                                         80 chars
block uniform [iblanked] [with_ghost] [range]            80 chars
i j k          # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)       3 ints
[imin imax jmin jmax kmin kmax]  # if range used:         6 ints
                        nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                              ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
x_origin y_origin z_origin                                3 floats
x_delta y_delta z_delta                                   3 floats
[ib_n1 ib_n2 ... ib_nn]                                  nn ints
[ghost_flags]                                            80 chars
[gf_e1 gf_e2 ... gf_ne]                                  ne ints
[node_ids]                                               80 chars
[id_n1 id_n2 ... id_nn]                                  nn ints
[element_ids]                                            80 chars
[id_e1 id_e2 ... id_ne]                                  ne ints
```

Fortran Binary form:

```
'Fortran Binary'                                         80 chars
'description line 1'                                     80 chars
'description line 2'                                     80 chars
'node id <off/given/assign/ignore>'                      80 chars
'element id <off/given/assign/ignore>'                   80 chars
['extents'                                               80 chars
'xmin xmax ymin ymax zmin zmax']                          6 floats
'part'                                                   80 chars
'#'                                                       1 int
'description line'                                       80 chars
'coordinates'                                            80 chars
'nn'                                                      1 int
['id_n1 id_n2 ... id_nn']                                nn ints
'x_n1 x_n2 ... x_nn'                                     nn floats
'y_n1 y_n2 ... y_nn'                                     nn floats
'z_n1 z_n2 ... z_nn'                                     nn floats
'element type'                                           80 chars
'ne'                                                      1 int
['id_e1 id_e2 ... id_ne']                                ne ints
'n1_e1 n2_e1 ... np_e1
 n1_e2 n2_e2 ... np_e2
 .
 .
 n1_ne n2_ne ... np_ne'                                  ne*np ints
'element type'                                           80 chars
 .
 .
'part'                                                   80 chars
 .
 .
'part'                                                   80 chars
```

```
'#'                                                          1 int
'description line'                                          80 chars
'block [iblanked] [with_ghost] [range]'                     80 chars
'i j k'          # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)         3 ints
['imin imax jmin jmax kmin kmax']  # if range used:          6 ints
                         nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                              ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
'x_n1 x_n2 ... x_nn'                                        nn floats
'y_n1 y_n2 ... y_nn'                                        nn floats
'z_n1 z_n2 ... z_nn'                                        nn floats
['ib_n1 ib_n2 ... ib_nn']                                   nn ints
['ghost_flags']                                             80 chars
['gf_e1 gf_e2 ... gf_ne']                                   ne ints
['node_ids']                                                80 chars
['id_n1 id_n2 ... id_nn']                                   nn ints
['element_ids']                                             80 chars
['id_e1 id_e2 ... id_ne']                                   ne ints
'part'                                                      80 chars
'#'                                                          1 int
'description line'                                          80 chars
'block rectilinear [iblanked] [with_ghost] [range]'         80 chars
'i j k'           # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)        3 ints
['imin imax jmin jmax kmin kmax']  # if range used:          6 ints
                         nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                              ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
'x_1 x_2 ... x_i'                                            i floats
'y_1 y_2 ... y_j'                                            j floats
'z_1 z_2 ... z_k'                                            k floats
['ib_n1 ib_n2 ... ib_nn']                                   nn ints
['ghost_flags']                                             80 chars
['gf_e1 gf_e2 ... gf_ne']                                   ne ints
['node_ids']                                                80 chars
['id_n1 id_n2 ... id_nn']                                   nn ints
['element_ids']                                             80 chars
['id_e1 id_e2 ... id_ne']                                   ne ints
'part'                                                      80 chars
'#'                                                          1 int
'description line'                                          80 chars
'block uniform [iblanked] [with_ghost] [range]'             80 chars
'i j k'            # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)       3 ints
['imin imax jmin jmax kmin kmax']  # if range used:          6 ints
                         nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                              ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
'x_origin y_origin z_origin                                  3 floats
 x_delta y_delta z_delta'                                    3 floats
['ib_n1 ib_n2 ... ib_nn']                                   nn ints
['ghost_flags']                                             80 chars
['gf_e1 gf_e2 ... gf_ne']                                   ne ints
['node_ids']                                                80 chars
['id_n1 id_n2 ... id_nn']                                   nn ints
['element_ids']                                             80 chars
['id_e1 id_e2 ... id_ne']                                   ne ints
```

ASCII form:

```
description line 1                                          A (max of 79 typ)
description line 2                                          A
node id <off/given/assign/ignore>                           A
element id <off/given/assign/ignore>                        A
```

```
[extents                                              A
xmin xmax                                             2E12.5
ymin ymax                                             2E12.5
zmin zmax]                                            2E12.5
part                                                  A
#                                                     I10
description line                                      A
coordinates                                           A
nn                                                    I10
[id_n1                                                I10    1/line (nn)
 id_n2
  .
  .
 id_nn]
x_n1                                                  E12.5  1/line (nn)
x_n2
 .
 .
x_nn
y_n1                                                  E12.5  1/line (nn)
y_n2
 .
 .
y_nn
z_n1                                                  E12.5  1/line (nn)
z_n2
 .
 .
z_nn
element type                                          A
ne                                                    I10
[id_e1                                                I10    1/line (ne)
 id_e2
  .
  .
 id_ne]
n1_e1 n2_e1 ... np_e1                                 I10    np/line
n1_e2 n2_e2 ... np_e2                                          (ne lines)
 .
 .
n1_ne n2_ne ... np_ne
element type                                          A
 .
 .
part                                                  A
 .
 .
part                                                  A
#                                                     I10
description line                                      A
block [iblanked] [with_ghost] [range]                 A
i j k           # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)  3I10
[imin imax jmin jmax kmin kmax]  # if range used:     6I10
                              nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                                   ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
x_n1                                                  E12.5  1/line (nn)
x_n2
 .
 .
x_nn
y_n1                                                  E12.5  1/line (nn)
```

```
y_n2
 .
 .
y_nn
z_n1                                                       E12.5  1/line (nn)
z_n2
 .
 .
z_nn
[ib_n1                                                     I10    1/line (nn)
 ib_n2
  .
  .
 ib_nn]
[ghost_flags]                                              80 chars
[gf_e1                                                     I10    1/line (ne)
 gf_e2
  .
  .
 gf_ne]
[node_ids]                                                 80 chars
[id_n1                                                     I10    1/line (nn)
 id_n2
  .
  .
 id_nn]
[element_ids]                                              80 chars
[id_e1                                                     I10    1/line (ne)
 id_e2
  .
  .
 id_ne]
part                                                       A
#                                                          I10
description line                                           A
block rectilinear [iblanked] [with_ghost] [range]          A
i j k            # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)      3I10
[imin imax jmin jmax kmin kmax]  # if range used:          6I10
                                  nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                                        ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
x_1                                                        E12.5  1/line (i)
x_2
 .
 .
x_i
y_1                                                        E12.5  1/line (j)
y_2
 .
 .
y_j
z_1                                                        E12.5  1/line (k)
z_2
 .
 .
z_k
[ib_n1                                                     I10    1/line (nn)
 ib_n2
  .
  .
 ib_nn]
[ghost_flags]                                              80 chars
```

```
[gf_e1                                              I10    1/line (ne)
 gf_e2
  .
  .
 gf_ne]
[node_ids]                                          80 chars
[id_n1                                              I10    1/line (nn)
 id_n2
  .
  .
 id_nn]
[element_ids]                                       80 chars
[id_e1                                              I10    1/line (ne)
 id_e2
  .
  .
 id_ne]
part                                                A
#                                                   I10
description line                                    A
block uniform [iblanked] [with_ghost] [range]       A
i j k            # nn = i*j*k, ne = (i-1)*(j-1)*(k-1)       3I10
[imin imax jmin jmax kmin kmax]  # if range used:   6I10
                              nn = (imax-imin+1)* (jmax-jmin+1)* (kmax-kmin+1)
                                    ne = (imax-imin)*(jmax-jmin)*(kmax-kmin)
x_origin                                            E12/5
y_origin                                            E12/5
z_origin                                            E12/5
x_delta                                             E12.5
y_delta                                             E12.5
z_delta                                             E12.5
[ib_n1                                              I10    1/line (nn)
 ib_n2
  .
  .
 ib_nn]
[ghost_flags]                                       80 chars
[gf_e1                                              I10    1/line (ne)
 gf_e2
  .
  .
 gf_ne]
[node_ids]                                          80 chars
[id_n1                                              I10    1/line (nn)
 id_n2
  .
  .
 id_nn]
[element_ids]                                       80 chars
[id_e1                                              I10    1/line (ne)
 id_e2
  .
  .
 id_ne]
```

Notes:

- If `node id` is `given` or `ignore`, the [id] section must be there for each part.
- If `element id` is `given` or `ignore`, the [id] section must be there for each `element type` of each part
- If `iblanked` is there, the [ib] section must be there for the block.
- x, y, and z coordinates are mandatory, even if a 2D problem.
- If `block rectilinear`, then the x, y, z coordinates change to the x, y, and z delta vectors.
- If `block uniform`, then the x, y, z coordinates change to the x, y, z coordinates of the origin and the x, y, and z delta values.
- If `block range`, the ijk min/max range line must follow the ijk line. And the number of nodes and elements is based on the ranges. The ijk line indicates the size of the original block.
- If `with_ghost` is on the `block` line, then the `ghost_flag` section must be there
- Ids are just labels, the coordinate (or element) order is implied.
- The minimum needed for unstructured empty parts is the three lines:

```
part
         #              (use the actual part number)
description
```

- The minimum needed for structured empty parts is the five lines:

```
part
         #              (use the actual part number)
description
block
         0         0         0
```

- Element blocks for `nsided` elements contain an additional section - **the number of nodes in each element.** See below.

C Binary form of element block, if nsided:

```
nsided                                                        80 chars
ne                                                             1 int
[id_n1 id_n2 ... id_ne]                                       ne ints
np1 np2 ... npne           This data is needed                ne ints
e1_n1 e1_n2 ... e1_np1
e2_n1 e2_n2 ... e2_np2
 .
 .
ne_n1 ne_n2 ... ne_npne                             np1+np2+...+npne ints
```

Fortran Binary form of element block, if nsided:

```
'nsided'                                                      80 chars
'ne'                                                           1 int
['id_n1 id_n2 ... id_ne']                                     ne ints
'np1 np2 ... npne'         This data is needed                ne ints
'e1_n1 e1_n2 ... e1_np1
```

```
 e2_n1 e2_n2 ... e2_np2
 .
 .
 ne_n1 ne_n2 ... ne_npne'                             np1+np2+...+npne ints
```

Ascii form of element block, if nsided:

```
nsided                                                  A
ne                                                      I10
[id_n1                                                  I10    1/line (ne)
 id_n2
  .
 id_ne]
np1                       This data is needed           I10    1/line (ne)
np2                        .
  .                        .
npne                       .
e1_n1 e1_n2 ... e1_np1                                  I10    np*/line
e2_n1 e2_n2 ... e2_np2                               (ne lines)
 .
ne_n1 ne_n2 ... ne_npne
```

- Element blocks for `nfaced` elements are more involved since they are described by their nsided faces. Thus, there is the optional section for ids (id_e*), a section for the number of faces per element (nf_e*), a section for number of nodes per face per element (np(f*_e*)), and a section for the connectivity of each nsided face of each element (n*(f*_e*)). See below.

C Binary form of element block, if nfaced:

```
nfaced                                                  80 chars
ne                                                       1 int
[id_e1 id_e2 ... id_ne]                                 ne ints
nf_e1 nf_e2 ... nf_ne                                   ne ints
np(f1_e1) np(f2_e1) ... np(nf_e1)
np(f1_e2) np(f2_e2) ... np(nf_e2)
.
.
np(f1_ne) np(f2_ne) ... np(nf_ne)                  nf_e1+nf_e2+...+nf_ne ints
n1(f1_e1) n2(f1_e1) ... n(np(f1_e1))
n1(f2_e1) n2(f2_e1) ... n(np(f2_e1))
.
n1(nf_e1) n2(nf_e1) ... n(np(nf_e1))
n1(f1_e2) n2(f1_e2) ... n(np(f1_e2))
n1(f2_e2) n2(f2_e2) ... n(np(f2_e2))
.
n1(nf_e2) n2(nf_e2) ... n(np(nf_e2))
.
.
n1(f1_ne) n2(f1_ne) ... n(np(f1_ne))
n1(f2_ne) n2(f2_ne) ... n(np(f2_ne))
.
n1(nf_ne) n2(nf_ne) ... n(np(nf_ne))         np(f1_e1)+np(f2_e1)+...+np(nf_ne) ints
```

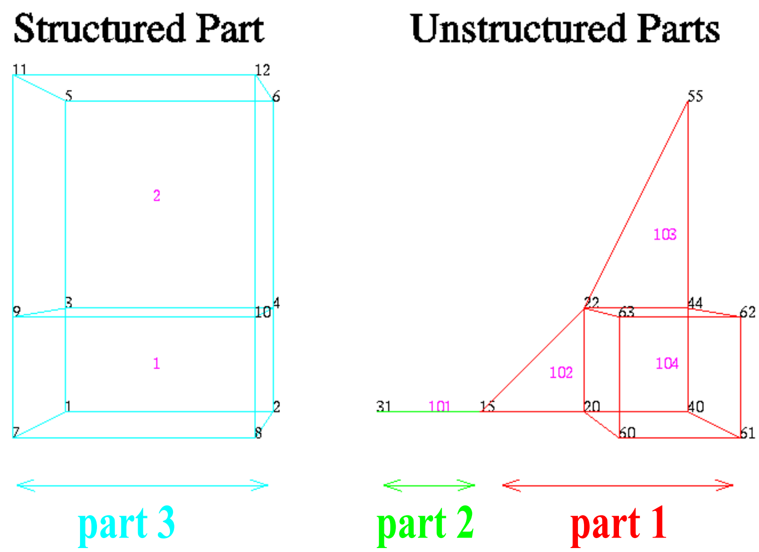
Fortran Binary form of element block, if nfaced:

```
'nfaced'                                                80 chars
'ne'                                                     1 int
['id_e1 id_e2 ... id_ne']                               ne ints
'nf_e1 nf_e2 ... nf_ne'                                 ne ints
```

```
'np(f1_e1) np(f2_e1) ... np(nf_e1)
np(f1_e2) np(f2_e2) ... np(nf_e2)
.
.
np(f1_ne) np(f2_ne) ... np(nf_ne)'                nf_e1+nf_e2+...+nf_ne ints
'n1(f1_e1) n2(f1_e1) ... n(np(f1_e1))
n1(f2_e1) n2(f2_e1) ... n(np(f2_e1))
.
n1(nf_e1) n2(nf_e1) ... n(np(nf_e1))
n1(f1_e2) n2(f1_e2) ... n(np(f1_e2))
n1(f2_e2) n2(f2_e2) ... n(np(f2_e2))
.
n1(nf_e2) n2(nf_e2) ... n(np(nf_e2))
.
.
n1(f1_ne) n2(f1_ne) ... n(np(f1_ne))
n1(f2_ne) n2(f2_ne) ... n(np(f2_ne))
.
n1(nf_ne) n2(nf_ne) ... n(np(nf_ne))'        np(f1_e1)+np(f2_e1)+...+np(nf_ne) ints
```

Ascii form of element block, if nfaced:

```
nfaced                                                  A
ne                                                      I10
[id_e1                                                  I10    1/line
 id_e2                                                         (ne lines)
  .
 id_ne]
nf_e1                                                   I10    1/line
nf_e2                                                          (ne lines)
.
nf_ne
np(f1_e1)                                               I10    1/line
np(f2_e1)                                         (nf_e1+nf_e2+...+nf_ne lines)
.
np(nf_e1)
np(f1_e2)
np(f2_e2)
.
np(nf_e2)
.
.
np(f1_ne)
np(f2_ne)
.
np(nf_ne)
n1(f1_e1) n2(f1_e1) ... n(np(f1_e1))                    I10  np*/line
n1(f2_e1) n2(f2_e1) ... n(np(f2_e1))              (nf_e1+nf_e2+...+nf_ne lines)
.
n1(nf_e1) n2(nf_e1) ... n(np(nf_e1))
n1(f1_e2) n2(f1_e2) ... n(np(f1_e2))
n1(f2_e2) n2(f2_e2) ... n(np(f2_e2)).
.
n1(nf_e2) n2(nf_e2) ... n(np(nf_e2))
.
.
n1(f1_ne) n2(f1_ne) ... n(np(f1_ne))
n1(f2_ne) n2(f2_ne) ... n(np(f2_ne))
.
n1(nf_ne) n2(nf_ne) ... n(np(nf_ne))
```



EnSight Gold Geometry File Example

The following is an example of an ASCII EnSight Gold geometry file: This is the same example model as given in the EnSight6 geometry file section (only in Gold format) with 11 defined unstructured nodes from which 2 unstructured parts are defined, and a 2x3x2 structured part as depicted in the above diagram.

Note: *The example file below (*`engold.geo`*) and all example variable files in the gold section (also prefixed with* `engold`*) may be found under your EnSight installation directory (path:* `$CEI_HOME/ensight92/data/user_manual`*).*

Note: *The appended "#" comment lines are for your reference only, and are not valid format lines within a geometry file as appended below.* ***Do NOT put these # comments in your file!!!***

```
This is the 1st description line of the EnSight Gold geometry example
This is the 2nd description line of the EnSight Gold geometry example
node id given
element id given
extents
 0.00000e+00 6.00000e+00
 0.00000e+00 3.00000e+00
 0.00000e+00 2.00000e+00
part
         1
2D uns-elements (description line for part 1)
coordinates
        10                 # nn             Do NOT put these # comments in your file!!
        15                 # node ids
        20
        40
        22
        44
        55
        60
        61
        62
        63
 4.00000e+00               # x components
 5.00000e+00
 6.00000e+00
```

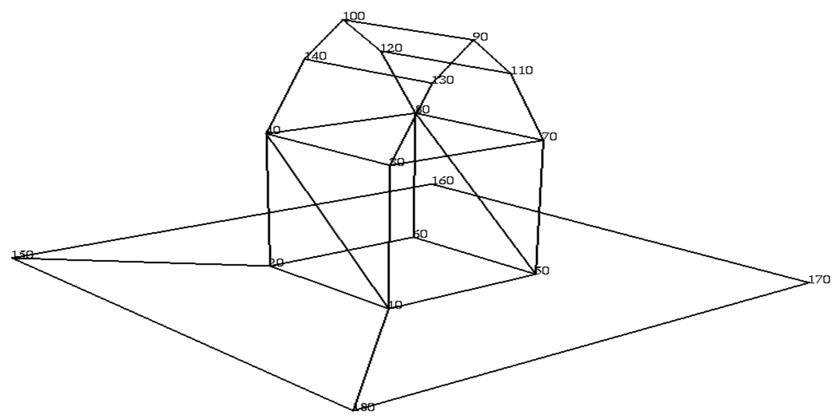
```
 5.00000e+00
 6.00000e+00
 6.00000e+00
 5.00000e+00
 6.00000e+00
 6.00000e+00
 5.00000e+00
 0.00000e+00                # y components
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 3.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 0.00000e+00                # z components
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
tria3                       # element type
         2                  # ne
       102                  # element ids
       103
         1         2         4
         4         5         6
hexa8
         1
       104
         2         3         5         4        7        8        9       10
part
         2
1D uns-elements (description line for part 2)
coordinates
         2
        15
        31
 4.00000e+00
 3.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
bar2
         1
       101
         2         1
part
         3
3D struct-part (description line fro part 3)
block iblanked
         2         3         2
 0.00000e+00                # i components
 2.00000e+00
 0.00000e+00
 2.00000e+00
 0.00000e+00
 2.00000e+00
 0.00000e+00
 2.00000e+00
```

```
0.00000e+00
2.00000e+00
0.00000e+00
2.00000e+00
0.00000e+00                # j components
0.00000e+00
1.00000e+00
1.00000e+00
3.00000e+00
3.00000e+00
0.00000e+00
0.00000e+00
1.00000e+00
1.00000e+00
3.00000e+00
3.00000e+00
0.00000e+00                # k components
0.00000e+00
0.00000e+00
0.00000e+00
0.00000e+00
0.00000e+00
2.00000e+00
2.00000e+00
2.00000e+00
2.00000e+00
2.00000e+00
2.00000e+00
        1                  # iblanking
        1
        1
        1
        1
        1
        1
        1
        1
        1
        1
        1
```

Simple example using nsided/ nfaced elements

The following is an example of an ASCII EnSight Gold geometry file with nsided and nfaced data. It is a non-realistic, simple model which is intended only to illustrate the format. Two nsided elements and three nfaced elements are used, even though the model could have been represented with a single nsided and single nfaced element.

Note: *The appended "#" comment lines are for your reference only, and are not valid format lines within a geometry file as appended below.* ***Do NOT put these # comments in your file!!!***




```
simple example for nsided/nfaced
element types in EnSight Gold Format
node id given
element id given
extents
-2.00000e+00 4.00000e+00
 0.00000e+00 3.50000e+00
-2.00000e+00 4.00000e+00
part
         1
barn
coordinates
        18                # nn             Do NOT put these # comments in your file!!
        10                # node ids
        20
        30
        40
        50
        60
        70
        80
        90
       100
       110
       120
       130
       140
       150
       160
       170
       180
 0.00000e+00               # x components
 2.00000e+00
 0.00000e+00
 2.00000e+00
```

```
 0.00000e+00
 2.00000e+00
 0.00000e+00
 2.00000e+00
 0.00000e+00
 2.00000e+00
 0.00000e+00
 2.00000e+00
 0.00000e+00
 2.00000e+00
 4.00000e+00
 4.00000e+00
-2.00000e+00
-2.00000e+00
 0.00000e+00                # y components
 0.00000e+00
 2.00000e+00
 2.00000e+00
 0.00000e+00
 0.00000e+00
 2.00000e+00
 2.00000e+00
 3.50000e+00
 3.50000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00                # z components
 0.00000e+00
 0.00000e+00
 0.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 1.00000e+00
 1.00000e+00
 1.50000e+00
 1.50000e+00
 0.50000e+00
 0.50000e+00
-2.00000e+00
 4.00000e+00
 4.00000e+00
-2.00000e+00
nsided
         2                                                                     # 2 nsided elements
       101                                                                     # element ids
       202
         4                                                                   # 4 nodes in first element
         8                                                                  # 8 nodes in second element
         2       15       18        1                                       # connectivity of element 1
         1       18       17       16       15        2        6        5   # connectivity of element 2
nfaced
         3                                                                 # 3 nfaced polyhedra elements
      1001                                                                     # element ids
      1002
      1003
         5                                         # number of faces in element 1
         5                                         # number of faces in element 2
         7                                         # number of faces in element 3
         3                                         # number of nodes in face 1 of element 1
```

```
  3                                                 #                    face 2 of element 1
  4                                                 #                    face 3 of element 1
  4                                                 #                    face 4 of element 1
  4                                                 #                    face 5 of element 1
  3                                                 # number of nodes in face 1 of element 2
  3                                                 #                    face 2 of element 2
  4                                                 #                    face 3 of element 2
  4                                                 #                    face 4 of element 2
  4                                                 #                    face 5 of element 2
  5                                                 # number of nodes in face 1 of element 3
  5                                                 #                    face 2 of element 3
  4                                                 #                    face 3 of element 3
  4                                                 #                    face 4 of element 3
  4                                                 #                    face 5 of element 3
  4                                                 #                    face 6 of element 3
  4                                                 #                    face 7 of element 3
  5         6         8                             # connectivity of face 1 of element 1
  2         1         4                             #                 face 2 of element 1
  6         2         4         8                   #                 face 3 of element 1
  8         4         1         5                   #                 face 4 of element 1
  1         2         6         5                   #                 face 5 of element 1
  5         8         7                             # connectivity of face 1 of element 2
  1         3         4                             #                 face 2 of element 2
  7         8         4         3                   #                 face 3 of element 2
  7         3         1         5                   #                 face 4 of element 2
  5         1         4         8                   #                 face 5 of element 2
  8         4        14        10        12         # connectivity of face 1 of element 3
  7        11         9        13         3         #                 face 2 of element 3
  7         8        12        11                   #                 face 3 of element 3
 11        12        10         9                   #                 face 4 of element 3
  9        10        14        13                   #                 face 5 of element 3
 13        14         4         3                   #                 face 6 of element 3
  7         3         4         8                   #                 face 7 of element 3
```

Important note concerning use of nsided and nfaced element representations.

It is important to know that the execution time and memory use for nsided or nfaced elements is ***significantly*** increased. It is not advisable to simpy use the nsided/nfaced representations for all elements, even those that could be represented with the simple basic elements. EnSight has no problem with different element types begin used within a part.

If, for example, you have a model with triangles, quads and some nsided elements. You will be far ahead to represent them in the EnSight data format as triangles, quads and the few nsided. Do not represent them all as nsided.

Simple examples using ghost cells

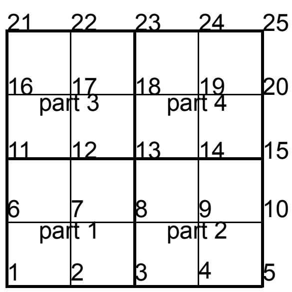
The following two ASCII EnSight Gold geometry file examples show use of ghost cells in unstructured and structured models. First the geometry file for the total model, composed of four parts, is given without any ghost cells. Then, two of four separate geometry files – each containing just one of the original parts (and appropriate ghost cells) will be given. This is supposed to simulate a decomposed model, such as you might provide for EnSight's Server-of-Servers.

Note:

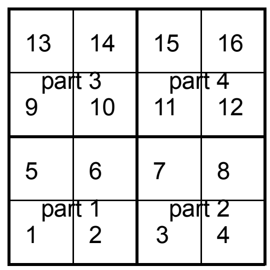
For unstructured models, ghost cells are simply a new element type. For structured models, ghost cells are an "iblank"-like flag.

Unstructured model

Nodes



Elements



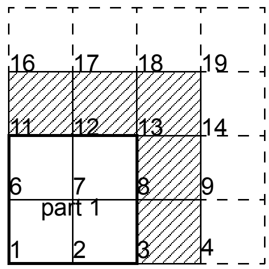
Total Unstructured Model Geometry File:

```
EnSight Model Geometry File
EnSight 7.1.0
node id given
element id given
extents
 0.00000e+00 4.00000e+00
 0.00000e+00 4.00000e+00
 0.00000e+00 0.00000e+00
part
         1
bottom left
coordinates
         9
         1
         2
         3
         6
         7
         8
        11
        12
        13
 0.00000e+00
 1.00000e+00
 2.00000e+00
 0.00000e+00
 1.00000e+00
 2.00000e+00
 0.00000e+00
 1.00000e+00
 2.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 1.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
quad4
         4
         1
         2
         5
         6
         1         2         5         4
         2         3         6         5
         4         5         8         7
         5         6         9         8
part
         2
bottom right
coordinates
         9
         3
         4
         5
         8
```

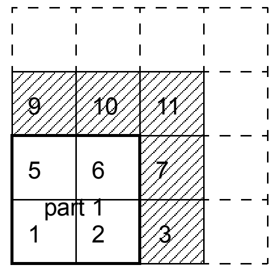
```
         9
        10
        13
        14
        15
 2.00000e+00
 3.00000e+00
 4.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 1.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
quad4
         4
         3
         4
         7
         8
         1         2         5         4
         2         3         6         5
         4         5         8         7
         5         6         9         8
part
         3
top left
coordinates
         9
        11
        12
        13
        16
        17
        18
        21
        22
        23
 0.00000e+00
 1.00000e+00
 2.00000e+00
 0.00000e+00
 1.00000e+00
 2.00000e+00
 0.00000e+00
 1.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 4.00000e+00
 4.00000e+00
 4.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
quad4
         4
         9
        10
        13
        14
         1         2         5         4
         2         3         6         5
         4         5         8         7
         5         6         9         8
part
         4
top right
coordinates
         9
        13
        14
        15
        18
        19
        20
        23
        24
        25
 2.00000e+00
 3.00000e+00
 4.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 4.00000e+00
 4.00000e+00
 4.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
quad4
         4
        11
        12
        15
        16
         1         2         5         4
         2         3         6         5
         4         5         8         7
         5         6         9         8
```

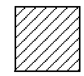
Portion with part 1 containing ghost cells (other parts are empty)

Nodes



Elements




Ghost Cell

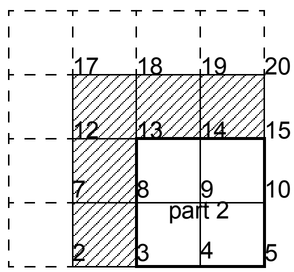
Note, the images are labeled with node ids - but the element connectivities are, and must be, based on local node indices.

```
EnSight Model Geometry File
part 1 portion
node id given
element id given
extents
 0.00000e+00 4.00000e+00     0.00000e+00
4.00000e+00
 0.00000e+00 0.00000e+00
part
         1
bottom left
coordinates
        16
         1
         2
         3
         4
         6
         7
         8
         9
        11
        12
        13
        14
        16
        17
        18
        19
 0.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 0.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 0.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 0.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 1.00000e+00
 1.00000e+00
 2.00000e+00
```

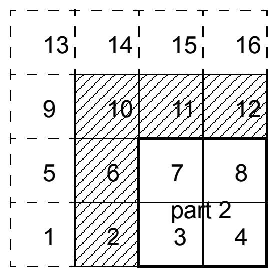
```
 2.00000e+00
 2.00000e+00
 2.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
quad4
         4
         1
         2
         5
         6
         1         2         6         5
         2         3         7         6
         5         6        10         9
         6         7        11        10
g_quad4
         5
         1
         1
         1
         1
         1
         3         4         8         7
         7         8        12        11
         9        10        14        13
        10        11        15        14
        11        12        16        15
part                                /* Empty part */
         2
bottom right
part                                /* Empty part */
         3
top left
part                                /* Empty part */
         4
top right
```

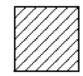
Portion with part 2 containing ghost cells (other parts are empty)

Nodes



Elements



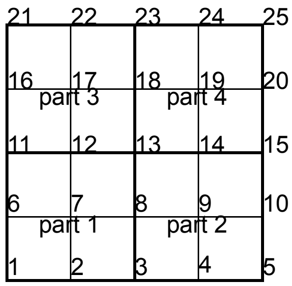


Note, the images are labeled with node ids - but the element connectivities are, and must be, based on local node indices.

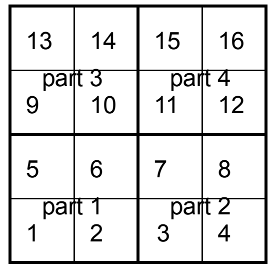
```
EnSight Model Geometry File
part 2 portion
node id given
element id given
extents
 0.00000e+00 4.00000e+00
 0.00000e+00 4.00000e+00
 0.00000e+00 0.00000e+00
part                          /* Empty part */
         1
bottom left
part
         2
bottom right
coordinates
        16
         2
         3
         4
         5
         7
         8
         9
        10
        12
        13
        14
        15
        17
        18
        19
        20
 1.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 1.00000e+00
 1.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 2.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
quad4
         4
         3
         4
         7
         8
         2         3         7         6
         3         4         8         7
         6         7        11        10
         7         8        12        11
g_quad4
         5
         1
         1
         1
         1
         1
         1         2         6         5
         5         6        10         9
         9        10        14        13
        10        11        15        14
        11        12        16        15
part                          /* Empty part */
         3
top left
part                          /* Empty part */
         4
top right
```

Structured model
(using essentially the same model, but in structured format):

Nodes



Elements

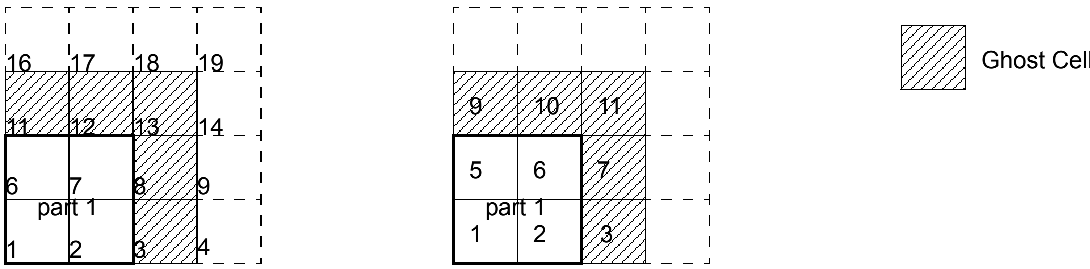


5 x 5 x 1 Total Structured Model

```
EnSight Model Geometry File
Total Structured Model
node id assign
element id assign
extents
 0.00000e+00 4.00000e+00
 0.00000e+00 4.00000e+00
 0.00000e+00 0.00000e+00
part
         1
left bottom
block uniform range
         5         5         1
         1         3         1         3         1         1
 0.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 0.00000e+00
part
         2
bottom right
block uniform range
         5         5         1
         3         5         1         3         1         1
 2.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 0.00000e+00
part
         3
top left
block uniform range
         5         5         1
         1         3         3         5         1         1
 0.00000e+00
 2.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 0.00000e+00
part
         4
top right
block uniform range
         5         5         1
         3         5         3         5         1         1
 2.00000e+00
 2.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 0.00000e+00
```

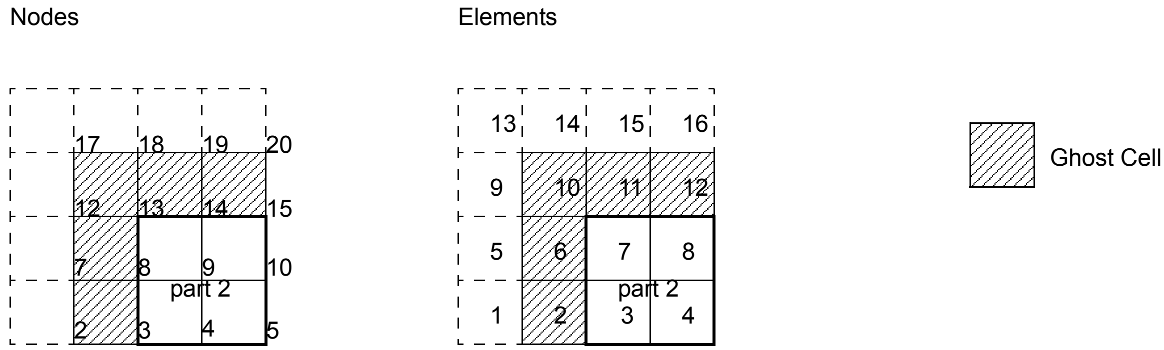
Portion with part 1 containing ghost cells (other parts are empty)

Nodes   Elements




```
EnSight Model Geometry File
part 1 portion only
node id assign
element id assign
extents
 0.00000e+00 4.00000e+00
 0.00000e+00 4.00000e+00
 0.00000e+00 0.00000e+00
part
         1
left bottom
block uniform range with_ghost
         4         4         1
         1         4         1         4         1         1
 0.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 0.00000e+00
ghost_flags
         0
         0
         1
         0
         0
         1
         1
         1
         1
Part                    /* Empty Part */
         2
right bottom
block
         0         0         0
part                    /* Empty Part */
         3
left top
block
         0         0         0
part                    /* Empty Part */
         4
right top
block
         0         0         0
```

Portion with part 2 containing ghost cells (other parts are empty)




```
EnSight Model Geometry File
part 2 portion only
node id assign
element id assign
extents
 0.00000e+00 4.00000e+00
 0.00000e+00 4.00000e+00
 0.00000e+00 0.00000e+00
part
         1
left bottom
block
         0         0         0
part
         2
right bottom
block uniform range with_ghost
         4         4         1
         2         5         1         4         1         1
 1.00000e+00
 0.00000e+00
 0.00000e+00
 1.00000e+00
 1.00000e+00
 0.00000e+00
ghost_flags
         1
         0
         0
         1
         0
         0
         1
         1
         1
part                 /* Empty Part */
         3
left top
block
         0         0        0
part                 /* Empty Part */
         4
right top
block
         0         0         0
```

Note: For both the unstructured and the structured model above, only the first two files (parts 1 and 2) are given. The portion files for parts 3 and 4 are not given, but would be similar to those for parts 1 and 2.

EnSight Gold Variable File Format

EnSight Gold variable files can either be per_node or per_element. They cannot be both. However, an EnSight model can have some variables which are per_node and others which are per_element.

EnSight Gold Per_Node Variable File Format

EnSight Gold variable files for per_node variables contain values for each unstructured node and for each structured node. First comes a single description line. Second comes a part line. Third comes a line containing the part number. Fourth comes a ‘coordinates’ line or a ‘block’ line. If a ‘coordinates’ line, the value for each unstructured node of the part follows. If it is a scalar file, there is one value per node, while for vector files there are three values per node (output in the same component order as the coordinates, namely, all x components, then all y components, then all z components). If it is a ‘block’ line, the value(s) for each structured node follows. The values for each node of the structured block are output in the same IJK order as the coordinates. (The number of nodes in the part are obtained from the corresponding EnSight Gold geometry file.)

*Note: If the geometry of given part is **empty**, nothing for that part needs to be in the variable file.*

C Binary form:

SCALAR FILE:

```
description line 1                                        80 chars
part                                                      80 chars
#                                                          1 int
coordinates                                               80 chars
s_n1 s_n2 ... s_nn                                        nn floats
part                                                      80 chars
 .
 .
part                                                      80 chars
#                                                          1 int
block                    # nn = i*j*k                     80 chars
s_n1 s_n2 ... s_nn                                        nn floats
```

VECTOR FILE:

```
description line 1                                        80 chars
part                                                      80 chars
#                                                          1 int
coordinates                                               80 chars
vx_n1 vx_n2 ... vx_nn                                     nn floats
vy_n1 vy_n2 ... vy_nn                                     nn floats
vz_n1 vz_n2 ... vz_nn                                     nn floats
part                                                      80 chars
 .
 .
part                                                      80 chars
#                                                          1 int
block                    # nn = i*j*k                     80 chars
vx_n1 vx_n2 ... vx_nn                                     nn floats
```

```
vy_n1 vy_n2 ... vy_nn                                        nn floats
vz_n1 vz_n2 ... vz_nn                                        nn floats
```

TENSOR FILE:

```
description line 1                                           80 chars
part                                                         80 chars
#                                                             1 int
coordinates                                                  80 chars
v11_n1 v11_n2 ... v11_nn                                     nn floats
v22_n1 v22_n2 ... v22_nn                                     nn floats
v33_n1 v33_n2 ... v33_nn                                     nn floats
v12_n1 v12_n2 ... v12_nn                                     nn floats
v13_n1 v13_n2 ... v13_nn                                     nn floats
v23_n1 v23_n2 ... v23_nn                                     nn floats
part                                                         80 chars
 .
 .
part                                                         80 chars
#                                                             1 int
block                    # nn = i*j*k                        80 chars
v11_n1 v11_n2 ... v11_nn                                     nn floats
v22_n1 v22_n2 ... v22_nn                                     nn floats
v33_n1 v33_n2 ... v33_nn                                     nn floats
v12_n1 v12_n2 ... v12_nn                                     nn floats
v13_n1 v13_n2 ... v13_nn                                     nn floats
v23_n1 v23_n2 ... v23_nn                                     nn floats
```

TENSOR9 FILE:

```
description line 1                                           80 chars
part                                                         80 chars
#                                                             1 int
coordinates                                                  80 chars
v11_n1 v11_n2 ... v11_nn                                     nn floats
v12_n1 v12_n2 ... v12_nn                                     nn floats
v13_n1 v13_n2 ... v13_nn                                     nn floats
v21_n1 v21_n2 ... v21_nn                                     nn floats
v22_n1 v22_n2 ... v22_nn                                     nn floats
v23_n1 v23_n2 ... v23_nn                                     nn floats
v31_n1 v31_n2 ... v31_nn                                     nn floats
v32_n1 v32_n2 ... v32_nn                                     nn floats
v33_n1 v33_n2 ... v33_nn                                     nn floats
part                                                         80 chars
 .
 .
part                                                         80 chars
#                                                             1 int
block                    # nn = i*j*k                        80 chars
v11_n1 v11_n2 ... v11_nn                                     nn floats
v12_n1 v12_n2 ... v12_nn                                     nn floats
v13_n1 v13_n2 ... v13_nn                                     nn floats
v21_n1 v21_n2 ... v21_nn                                     nn floats
v22_n1 v22_n2 ... v22_nn                                     nn floats
v23_n1 v23_n2 ... v23_nn                                     nn floats
v21_n1 v21_n2 ... v21_nn                                     nn floats
v22_n1 v22_n2 ... v22_nn                                     nn floats
v23_n1 v23_n2 ... v23_nn                                     nn floats
```

COMPLEX SCALAR FILES (Real and/or Imaginary):

```
description line 1                                    80 chars
part                                                  80 chars
#                                                      1 int
coordinates                                           80 chars
s_n1 s_n2 ... s_nn                                    nn floats
part                                                  80 chars
 .
 .
part                                                  80 chars
#                                                      1 int
block                   # nn = i*j*k                  80 chars
s_n1 s_n2 ... s_nn                                    nn floats
```

COMPLEX VECTOR FILES (Real and/or Imaginary):

```
description line 1                                    80 chars
part                                                  80 chars
#                                                      1 int
coordinates                                           80 chars
vx_n1 vx_n2 ... vx_nn                                 nn floats
vy_n1 vy_n2 ... vy_nn                                 nn floats
vz_n1 vz_n2 ... vz_nn                                 nn floats
part                                                  80 chars
 .
 .
part                                                  80 chars
#                                                      1 int
block                   # nn = i*j*k                  80 chars
vx_n1 vx_n2 ... vx_nn                                 nn floats
vy_n1 vy_n2 ... vy_nn                                 nn floats
vz_n1 vz_n2 ... vz_nn                                 nn floats
```

Fortran Binary form:

SCALAR FILE:

```
'description line 1'                                  80 chars
'part'                                                80 chars
'#'                                                    1 int
'coordinates'                                         80 chars
's_n1 s_n2 ... s_nn'                                  nn floats
'part'                                                80 chars
 .
 .
'part'                                                80 chars
'#'                                                    1 int
'block'                   # nn = i*j*k                80 chars
's_n1 s_n2 ... s_nn'                                  nn floats
```

VECTOR FILE:

```
'description line 1'                                  80 chars
'part'                                                80 chars
'#'                                                    1 int
'coordinates'                                         80 chars
'vx_n1 vx_n2 ... vx_nn'                               nn floats
```

```
'vy_n1 vy_n2 ... vy_nn'                                    nn floats
'vz_n1 vz_n2 ... vz_nn'                                    nn floats
'part'                                                     80 chars
 .
 .
'part'                                                     80 chars
'#'                                                         1 int
'block'                     # nn = i*j*k                   80 chars
'vx_n1 vx_n2 ... vx_nn'                                    nn floats
'vy_n1 vy_n2 ... vy_nn'                                    nn floats
'vz_n1 vz_n2 ... vz_nn'                                    nn floats
```

TENSOR FILE:

```
'description line 1'                                       80 chars
'part'                                                     80 chars
'#'                                                         1 int
'coordinates'                                              80 chars
'v11_n1 v11_n2 ... v11_nn'                                 nn floats
'v22_n1 v22_n2 ... v22_nn'                                 nn floats
'v33_n1 v33_n2 ... v33_nn'                                 nn floats
'v12_n1 v12_n2 ... v12_nn'                                 nn floats
'v13_n1 v13_n2 ... v13_nn'                                 nn floats
'v23_n1 v23_n2 ... v23_nn'                                 nn floats
'part'                                                     80 chars
 .
 .
'part'                                                     80 chars
'#'                                                         1 int
'block'                     # nn = i*j*k                   80 chars
'v11_n1 v11_n2 ... v11_nn'                                 nn floats
'v22_n1 v22_n2 ... v22_nn'                                 nn floats
'v33_n1 v33_n2 ... v33_nn'                                 nn floats
'v12_n1 v12_n2 ... v12_nn'                                 nn floats
'v13_n1 v13_n2 ... v13_nn'                                 nn floats
'v23_n1 v23_n2 ... v23_nn'                                 nn floats
```

TENSOR9 FILE:

```
'description line 1'                                       80 chars
'part'                                                     80 chars
'#'                                                         1 int
'coordinates'                                              80 chars
'v11_n1 v11_n2 ... v11_nn'                                 nn floats
'v12_n1 v12_n2 ... v12_nn'                                 nn floats
'v13_n1 v13_n2 ... v13_nn'                                 nn floats
'v21_n1 v21_n2 ... v21_nn'                                 nn floats
'v22_n1 v22_n2 ... v22_nn'                                 nn floats
'v23_n1 v23_n2 ... v23_nn'                                 nn floats
'v31_n1 v31_n2 ... v31_nn'                                 nn floats
'v32_n1 v32_n2 ... v32_nn'                                 nn floats
'v33_n1 v33_n2 ... v33_nn'                                 nn floats
'part'                                                     80 chars
 .
 .
'part'                                                     80 chars
'#'                                                         1 int
'block'                     # nn = i*j*k                   80 chars
'v11_n1 v11_n2 ... v11_nn'                                 nn floats
'v12_n1 v12_n2 ... v12_nn'                                 nn floats
```

```
'v13_n1 v13_n2 ... v13_nn'                        nn floats
'v21_n1 v21_n2 ... v21_nn'                        nn floats
'v22_n1 v22_n2 ... v22_nn'                        nn floats
'v23_n1 v23_n2 ... v23_nn'                        nn floats
'v31_n1 v31_n2 ... v31_nn'                        nn floats
'v32_n1 v32_n2 ... v32_nn'                        nn floats
'v33_n1 v33_n2 ... v33_nn'                        nn floats
```

COMPLEX SCALAR FILES (Real and/or Imaginary):

```
'description line 1'                              80 chars
'part'                                            80 chars
'#'                                                1 int
'coordinates'                                     80 chars
's_n1 s_n2 ... s_nn'                              nn floats
'part'                                            80 chars
 .
 .
'part'                                            80 chars
'#'                                                1 int
'block'                  # nn = i*j*k             80 chars
's_n1 s_n2 ... s_nn'                              nn floats
```

COMPLEX VECTOR FILES (Real and/or Imaginary):

```
'description line 1'                              80 chars
'part'                                            80 chars
'#'                                                1 int
'coordinates'                                     80 chars
'vx_n1 vx_n2 ... vx_nn'                           nn floats
'vy_n1 vy_n2 ... vy_nn'                           nn floats
'vz_n1 vz_n2 ... vz_nn'                           nn floats
'part'                                            80 chars
 .
 .
'part'                                            80 chars
'#'                                                1 int
'block'                  # nn = i*j*k             80 chars
'vx_n1 vx_n2 ... vx_nn'                           nn floats
'vy_n1 vy_n2 ... vy_nn'                           nn floats
'vz_n1 vz_n2 ... vz_nn'                           nn floats
```

ASCII form:

SCALAR FILE:

```
description line 1                                A (max of 79 typ)
part                                              A
#                                                 I10
coordinates                                       A
s_n1                                              E12.5  1/line (nn)
s_n2
 .
 .
s_nn
part                                              A
 .
```

```
 .
part                                                  A
#                                                     I10
block                    # nn = i*j*k                 A
s_n1                                                  E12.5  1/line (nn)
s_n2
 .
 .
s_nn
```

VECTOR FILE:

```
description line 1                                    A (max of 79 typ)
part                                                  A
#                                                     I10
coordinates                                           A
vx_n1                                                 E12.5  1/line (nn)
vx_n2
 .
 .
vx_nn
vy_n1                                                 E12.5  1/line (nn)
vy_n2
 .
 .
vy_nn
vz_n1                                                 E12.5  1/line (nn)
vz_n2
 .
 .
vz_nn
part                                                  A
 .
 .
part                                                  A
#                                                     I10
block                    # nn = i*j*k                 A
vx_n1                                                 E12.5  1/line (nn)
vx_n2
 .
 .
vx_nn
vy_n1                                                 E12.5  1/line (nn)
vy_n2
 .
 .
vy_nn
vz_n1                                                 E12.5  1/line (nn)
vz_n2
 .
 .
vz_nn
```

TENSOR FILE:

```
description line 1                                    A (max of 79 typ)
part                                                  A
#                                                     I10
coordinates                                           A
```

```
v11_n1                                        E12.5  1/line (nn)
v11_n2
 .
 .
v11_nn
v22_n1                                        E12.5  1/line (nn)
v22_n2
 .
 .
v22_nn
v33_n1                                        E12.5  1/line (nn)
v33_n2
 .
 .
v33_nn
v12_n1                                        E12.5  1/line (nn)
v12_n2
 .
 .
v12_nn
v13_n1                                        E12.5  1/line (nn)
v13_n2
 .
 .
v13_nn
v23_n1                                        E12.5  1/line (nn)
v23_n2
 .
 .
v23_nn
part                                          A
 .
 .
part                                          A
#                                             I10
block                 # nn = i*j*k            A
v11_n1                                        E12.5  1/line (nn)
v11_n2
 .
 .
v11_nn
v22_n1                                        E12.5  1/line (nn)
v22_n2
 .
 .
v22_nn
v33_n1                                        E12.5  1/line (nn)
v33_n2
 .
 .
v33_nn
v12_n1                                        E12.5  1/line (nn)
v12_n2
 .
 .
v12_nn
v13_n1                                        E12.5  1/line (nn)
v13_n2
 .
 .
v13_nn
```

```
v23_n1                                       E12.5  1/line (nn)
v23_n2
 .
 .
v23_nn
```

TENSOR9 FILE:

```
description line 1                           A (max of 79 typ)
part                                         A
#                                            I10
coordinates                                  A
v11_n1                                       E12.5  1/line (nn)
v11_n2
 .
 .
v11_nn
v12_n1                                       E12.5  1/line (nn)
v12_n2
 .
 .
v12_nn
v13_n1                                       E12.5  1/line (nn)
v13_n2
 .
 .
v13_nn
v21_n1                                       E12.5  1/line (nn)
v21_n2
 .
 .
v21_nn
v22_n1                                       E12.5  1/line (nn)
v22_n2
 .
 .
v22_nn
v23_n1                                       E12.5  1/line (nn)
v23_n2
 .
 .
v23_nn
v31_n1                                       E12.5  1/line (nn)
v31_n2
 .
 .
v31_nn
v32_n1                                       E12.5  1/line (nn)
v32_n2
 .
 .
v32_nn
v33_n1                                       E12.5  1/line (nn)
v33_n2
 .
 .
v33_nn
part                                         A
 .
 .
part                                         A
```

```
#                                                   I10
block                   # nn = i*j*k                A
v11_n1                                              E12.5  1/line (nn)
v11_n2
 .
 .
v11_nn
v12_n1                                              E12.5  1/line (nn)
v12_n2
 .
 .
v12_nn
v13_n1                                              E12.5  1/line (nn)
v13_n2
 .
 .
v13_nn
v21_n1                                              E12.5  1/line (nn)
v21_n2
 .
 .
v21_nn
v22_n1                                              E12.5  1/line (nn)
v22_n2
 .
 .
v22_nn
v23_n1                                              E12.5  1/line (nn)
v23_n2
 .
 .
v23_nn
v31_n1                                              E12.5  1/line (nn)
v31_n2
 .
 .
v31_nn
v32_n1                                              E12.5  1/line (nn)
v32_n2
 .
 .
v32_nn
v33_n1                                              E12.5  1/line (nn)
v33_n2
 .
 .
v33_nn
```

COMPLEX SCALAR FILES (Real and/or Imaginary):

```
description line 1                                  A (max of 79 typ)
part                                                A
#                                                   I10
coordinates                                         A
s_n1                                                E12.5  1/line (nn)
s_n2
 .
 .
s_nn
part                                                A
```

```
 .
 .
part                                                      A
#                                                         I10
block                       # nn = i*j*k                  A
s_n1                                                      E12.5  1/line (nn)
s_n2
 .
 .
s_nn
```

COMPLEX VECTOR FILES (Real and/or Imaginary):

```
description line 1                                        A (max of 79 typ)
part                                                      A
#                                                         I10
coordinates                                               A
vx_n1                                                     E12.5  1/line (nn)
vx_n2
 .
 .
vx_nn
vy_n1                                                     E12.5  1/line (nn)
vy_n2
 .
 .
vy_nn
vz_n1                                                     E12.5  1/line (nn)
vz_n2
 .
 .
vz_nn
part                                                      A
 .
 .
part                                                      A
#                                                         I10
block                       # nn = i*j*k                  A
vx_n1                                                     E12.5  1/line (nn)
vx_n2
 .
 .
vx_nn
vy_n1                                                     E12.5  1/line (nn)
vy_n2
 .
 .
vy_nn
vz_n1                                                     E12.5  1/line (nn)
vz_n2
 .
 .
vz_nn
```

The following variable file examples reflect scalar, vector, tensor, and complex variable values *per node* for the previously defined EnSight6 Gold Geometry File Example with 11 defined unstructured nodes and a 2x3x2 structured Part (Part number 3). The values are summarized in the following table.

Note: *These are the same values as listed in the EnSight6 per_node variable file section. Subsequently, the following example files contain the same data as the example files given in the EnSight6 section - only they are listed in gold format. (No asymmetric tensor example data given)*

	Node Index	Node Id	Scalar Value	Vector Values	Tensor (2nd order symm.) Values	ComplexScalar Real Value	ComplexScalar Imaginary Value
Unstructured							
	1	15	(1.)	(1.1, 1.2, 1.3)	(1.1, 1.2, 1.3, 1.4, 1.5, 1.6)	(1.1)	(1.2)
	2	31	(2.)	(2.1, 2.2, 2.3)	(2.1, 2.2, 2.3, 2.4, 2.5, 2.6)	(2.1)	(2.2)
	3	20	(3.)	(3.1, 3.2, 3.3)	(3.1, 3.2, 3.3, 3.4, 3.5, 3.6)	(3.1)	(3.2)
	4	40	(4.)	(4.1, 4.2, 4.3)	(4.1, 4.2, 4.3, 4.4, 4.5, 4.6)	(4.1)	(4.2)
	5	22	(5.)	(5.1, 5.2, 5.3)	(5.1, 5.2, 5.3, 5.4, 5.5, 5.6)	(5.1)	(5.2)
	6	44	(6.)	(6.1, 6.2, 6.3)	(6.1, 6.2, 6.3, 6.4, 6.5, 6.6)	(6.1)	(6.2)
	7	55	(7.)	(7.1, 7.2, 7.3)	(7.1, 7.2, 7.3, 7.4, 7.5, 7.6)	(7.1)	(7.2)
	8	60	(8.)	(8.1, 8.2, 8.3)	(8.1, 8.2, 8.3, 8.4, 8.5, 8.60	(8.1)	(8.2)
	9	61	(9.)	(9.1, 9.2, 9.3)	(9.1, 9.2, 9.3, 9.4, 9.5, 9.6)	(9.1)	(9.2)
	10	62	(10.)	(10.1,10.2,10.3)	(10.1,10.2,10.3,10.4,10.5,10.6)	(10.1)	(10.2)
	11	63	(11.)	(11.1,11.2,11.3)	(11.1,11.2,11.3,11.4,11.5,11.6)	(11.1)	(11.2)
Structured							
	1	1	(1.)	(1.1, 1.2, 1.3)	(1.1, 1.2, 1.3, 1.4, 1.5, 1.6)	(1.1)	(1.2)
	2	2	(2.)	(2.1, 2.2, 2.3)	(2.1, 2.2, 2.3, 2.4, 2.5, 2.6)	(2.1)	(2.2)
	3	3	(3.)	(3.1, 3.2, 3.3)	(3.1, 3.2, 3.3, 3.4, 3.5, 3.6)	(3.1)	(3.2)
	4	4	(4.)	(4.1, 4.2, 4.3)	(4.1, 4.2, 4.3, 4.4, 4.5, 4.6)	(4.1)	(4.2)
	5	5	(5.)	(5.1, 5.2, 5.3)	(5.1, 5.2, 5.3, 5.4, 5.5, 5.6)	(5.1)	(5.2)
	6	6	(6.)	(6.1, 6.2, 6.3)	(6.1, 6.2, 6.3, 6.4, 6.5, 6.6)	(6.1)	(6.2)
	7	7	(7.)	(7.1, 7.2, 7.3)	(7.1, 7.2, 7.3, 7.4, 7.5, 7.6)	(7.1)	(7.2)
	8	8	(8.)	(8.1, 8.2, 8.3)	(8.1, 8.2, 8.3, 8.4, 8.5, 8.6)	(8.1)	(8.2)
	9	9	(9.)	(9.1, 9.2, 9.3)	(9.1, 9.2, 9.3, 9.4, 9.5, 9.6)	(9.1)	(9.2)
	10	10	(10.)	(10.1,10.2,10.3)	(10.1,10.2,10.3,10.4,10.5,10.6)	(10.1)	(10.2)
	11	11	(11.)	(11.1,11.2,11.3)	(11.1,11.2,11.3,11.4,11.5,11.6)	(11.1)	(11.2)
	12	12	(12.)	(12.1,12.2,12.3)	(12.1,12.2,12.3,12.4,12.5,12.6)	(12.1)	(12.2)

Per_node (Scalar) Variable Example 1: This shows an ASCII scalar file (`engold.Nsca`) for the gold geometry example.

```
Per_node scalar values for the EnSight Gold geometry example
part
         1
coordinates
 1.00000E+00
 3.00000E+00
 4.00000E+00
 5.00000E+00
 6.00000E+00
 7.00000E+00
 8.00000E+00
 9.00000E+00
 1.00000E+01
 1.10000E+01
part
         2
coordinates
 1.00000E+00
 2.00000E+00
```

```
part
         3
block
 1.00000E+00
 2.00000E+00
 3.00000E+00
 4.00000E+00
 5.00000E+00
 6.00000E+00
 7.00000E+00
 8.00000E+00
 9.00000E+00
 1.00000E+01
 1.10000E+01
 1.20000E+01
```

Per_node (Vector) Variable Example 2: This example shows an ASCII vector file (`engold.Nvec`) for the gold geometry example.

```
Per_node vector values for the EnSight Gold geometry example
part
         1
coordinates
 1.10000E+00
 3.10000E+00
 4.10000E+00
 5.10000E+00
 6.10000E+00
 7.10000E+00
 8.10000E+00
 9.10000E+00
 1.01000E+01
 1.11000E+01
 1.20000E+00
 3.20000E+00
 4.20000E+00
 5.20000E+00
 6.20000E+00
 7.20000E+00
 8.20000E+00
 9.20000E+00
 1.02000E+01
 1.12000E+01
 1.30000E+00
 3.30000E+00
 4.30000E+00
 5.30000E+00
 6.30000E+00
 7.30000E+00
 8.30000E+00
 9.30000E+00
 1.03000E+01
 1.13000E+01
part
         2
coordinates
 1.10000E+00
 2.10000E+00
 1.20000E+00
 2.20000E+00
 1.30000E+00
 2.30000E+00
part
         3
block
 1.10000E+00
```

```
 2.10000E+00
 3.10000E+00
 4.10000E+00
 5.10000E+00
 6.10000E+00
 7.10000E+00
 8.10000E+00
 9.10000E+00
 1.01000E+01
 1.11000E+01
 1.21000E+01
 1.20000E+00
 2.20000E+00
 3.20000E+00
 4.20000E+00
 5.20000E+00
 6.20000E+00
 7.20000E+00
 8.20000E+00
 9.20000E+00
 1.02000E+01
 1.12000E+01
 1.22000E+01
 1.30000E+00
 2.30000E+00
 3.30000E+00
 4.30000E+00
 5.30000E+00
 6.30000E+00
 7.30000E+00
 8.30000E+00
 9.30000E+00
 1.03000E+01
 1.13000E+01
 1.23000E+01
```

Per_node (Tensor) Variable Example 3: This example shows an ASCII 2nd order symmetric tensor file (`engold.Nten`) for the gold geometry example.

```
Per_node symmetric tensor values for the EnSight Gold geometry example
part
         1
coordinates
 1.10000E+00
 3.10000E+00
 4.10000E+00
 5.10000E+00
 6.10000E+00
 7.10000E+00
 8.10000E+00
 9.10000E+00
 1.01000E+01
 1.11000E+01
 1.20000E+00
 3.20000E+00
 4.20000E+00
 5.20000E+00
 6.20000E+00
 7.20000E+00
 8.20000E+00
 9.20000E+00
 1.02000E+01
 1.12000E+01
 1.30000E+00
 3.30000E+00
 4.30000E+00
```

```
 5.30000E+00
 6.30000E+00
 7.30000E+00
 8.30000E+00
 9.30000E+00
 1.03000E+01
 1.13000E+01
 1.40000E+00
 3.40000E+00
 4.40000E+00
 5.40000E+00
 6.40000E+00
 7.40000E+00
 8.40000E+00
 9.40000E+00
 1.04000E+01
 1.14000E+01
 1.50000E+00
 3.50000E+00
 4.50000E+00
 5.50000E+00
 6.50000E+00
 7.50000E+00
 8.50000E+00
 9.50000E+00
 1.05000E+01
 1.15000E+01
 1.60000E+00
 3.60000E+00
 4.60000E+00
 5.60000E+00
 6.60000E+00
 7.60000E+00
 8.60000E+00
 9.60000E+00
 1.06000E+01
 1.16000E+01
part
         2
coordinates
 1.10000E+00
 2.10000E+00
 1.20000E+00
 2.20000E+00
 1.30000E+00
 2.30000E+00
 1.40000E+00
 2.40000E+00
 1.50000E+00
 2.50000E+00
 1.60000E+00
 2.60000E+00
part
         3
block
 1.10000E+00
 2.10000E+00
 3.10000E+00
 4.10000E+00
 5.10000E+00
 6.10000E+00
 7.10000E+00
 8.10000E+00
 9.10000E+00
 1.01000E+01
 1.11000E+01
 1.21000E+01
```

```
1.20000E+00
2.20000E+00
3.20000E+00
4.20000E+00
5.20000E+00
6.20000E+00
7.20000E+00
8.20000E+00
9.20000E+00
1.02000E+01
1.12000E+01
1.22000E+01
1.30000E+00
2.30000E+00
3.30000E+00
4.30000E+00
5.30000E+00
6.30000E+00
7.30000E+00
8.30000E+00
9.30000E+00
1.03000E+01
1.13000E+01
1.23000E+01
1.40000E+00
2.40000E+00
3.40000E+00
4.40000E+00
5.40000E+00
6.40000E+00
7.40000E+00
8.40000E+00
9.40000E+00
1.04000E+01
1.14000E+01
1.24000E+01
1.50000E+00
2.50000E+00
3.50000E+00
4.50000E+00
5.50000E+00
6.50000E+00
7.50000E+00
8.50000E+00
9.50000E+00
1.05000E+01
1.15000E+01
1.25000E+01
1.60000E+00
2.60000E+00
3.60000E+00
4.60000E+00
5.60000E+00
6.60000E+00
7.60000E+00
8.60000E+00
9.60000E+00
1.06000E+01
1.16000E+01
1.26000E+01
```

Per_node (Complex) Variable Example 4: This example shows ASCII complex real (`engold.Ncmp_r`) and imaginary (`engold.Ncmp_i`) *scalar* files for the gold geometry example. (The same methodology would apply for complex real and imaginary *vector* files.)

Real scalar File:

```
Per_node complex real scalar values for the EnSight Gold geometry example
part
         1
coordinates
 1.10000E+00
 3.10000E+00
 4.10000E+00
 5.10000E+00
 6.10000E+00
 7.10000E+00
 8.10000E+00
 9.10000E+00
 1.01000E+01
 1.11000E+01
part
         2
coordinates
 1.10000E+00
 2.10000E+00
part
         3
block
 1.10000E+00
 2.10000E+00
 3.10000E+00
 4.10000E+00
 5.10000E+00
 6.10000E+00
 7.10000E+00
 8.10000E+00
 9.10000E+00
 1.01000E+01
 1.11000E+01
 1.21000E+01
```

Imaginary scalar File:

```
Per_node complex imaginary scalar values for the EnSight Gold geometry example
part
         1
coordinates
 1.20000E+00
 3.20000E+00
 4.20000E+00
 5.20000E+00
 6.20000E+00
 7.20000E+00
 8.20000E+00
 9.20000E+00
 1.02000E+01
 1.12000E+01
part
         2
coordinates
 1.20000E+00
 2.20000E+00
part
         3
block
```

```
 1.20000E+00
 2.20000E+00
 3.20000E+00
 4.20000E+00
 5.20000E+00
 6.20000E+00
 7.20000E+00
 8.20000E+00
 9.20000E+00
 1.02000E+01
 1.12000E+01
 1.22000E+01
```

EnSight Gold Per_Element Variable File Format

EnSight Gold variable files for per_element variables contain values for each element of designated types of designated Parts. First comes a single description line. Second comes a Part line. Third comes a line containing the part number. Fourth comes an element type line and then comes the value for each element of that type and part. If it is a scalar variable, there is one value per element, while for vector variables there are three values per element. (The number of elements of the given type are obtained from the corresponding EnSight Gold geometry file.)

*Note: If the geometry of given part is **empty**, nothing for that part needs to be in the variable file.*

C Binary form:

SCALAR FILE:

```
description line 1                                              80 chars
part                                                            80 chars
#                                                                1 int
element type                                                    80 chars
s_e1 s_e2 ... s_ne                                              ne floats
element type                                                    80 chars
 .
 .
part                                                            80 chars
 .
 .
part                                                            80 chars
#                                                                1 int
block                    # nn = (i-1)*(j-1)*(k-1)               80 chars
s_n1 s_n2 ... s_nn                                              nn floats
```

VECTOR FILE:

```
description line 1                                              80 chars
part                                                            80 chars
#                                                                1 int
element type                                                    80 chars
vx_e1 vx_e2 ... vx_ne                                           ne floats
vy_e1 vy_e2 ... vy_ne                                           ne floats
vz_e1 vz_e2 ... vz_ne                                           ne floats
```

```
element type                                                80 chars
 .
 .
part                                                        80 chars
 .
 .
part                                                        80 chars
#                                                            1 int
block                    # nn = (i-1)*(j-1)*(k-1)           80 chars
vx_n1 vx_n2 ... vx_nn                                       nn floats
vy_n1 vy_n2 ... vy_nn                                       nn floats
vz_n1 vz_n2 ... vz_nn                                       nn floats
```

TENSOR FILE:

```
description line 1                                          80 chars
part                                                        80 chars
#                                                            1 int
element type                                                80 chars
v11_e1 v11_e2 ... v11_ne                                    ne floats
v22_e1 v22_e2 ... v22_ne                                    ne floats
v33_e1 v33_e2 ... v33_ne                                    ne floats
v12_e1 v12_e2 ... v12_ne                                    ne floats
v13_e1 v13_e2 ... v13_ne                                    ne floats
v23_e1 v23_e2 ... v23_ne                                    ne floats
element type                                                80 chars
 .
 .
part                                                        80 chars
 .
 .
part                                                        80 chars
#                                                            1 int
block                    # nn = (i-1)*(j-1)*(k-1)           80 chars
v11_n1 v11_n2 ... v11_nn                                    nn floats
v22_n1 v22_n2 ... v22_nn                                    nn floats
v33_n1 v33_n2 ... v33_nn                                    nn floats
v12_n1 v12_n2 ... v12_nn                                    nn floats
v13_n1 v13_n2 ... v13_nn                                    nn floats
v23_n1 v23_n2 ... v23_nn                                    nn floats
```

TENSOR9 FILE:

```
description line 1                                          80 chars
part                                                        80 chars
#                                                            1 int
element type                                                80 chars
v11_e1 v11_e2 ... v11_ne                                    ne floats
v12_e1 v12_e2 ... v12_ne                                    ne floats
v13_e1 v13_e2 ... v13_ne                                    ne floats
v21_e1 v21_e2 ... v21_ne                                    ne floats
v22_e1 v22_e2 ... v22_ne                                    ne floats
v23_e1 v23_e2 ... v23_ne                                    ne floats
v31_e1 v31_e2 ... v31_ne                                    ne floats
v32_e1 v32_e2 ... v32_ne                                    ne floats
v33_e1 v33_e2 ... v33_ne                                    ne floats
element type                                                80 chars
 .
 .
```

```
part                                                   80 chars
 .
 .
part                                                   80 chars
#                                                       1 int
block                    # nn = (i-1)*(j-1)*(k-1)      80 chars
v11_n1 v11_n2 ... v11_nn                               nn floats
v12_n1 v12_n2 ... v12_nn                               nn floats
v13_n1 v13_n2 ... v13_nn                               nn floats
v21_n1 v21_n2 ... v21_nn                               nn floats
v22_n1 v22_n2 ... v22_nn                               nn floats
v23_n1 v23_n2 ... v23_nn                               nn floats
v31_n1 v31_n2 ... v31_nn                               nn floats
v32_n1 v32_n2 ... v32_nn                               nn floats
v33_n1 v33_n2 ... v33_nn                               nn floats
```

COMPLEX SCALAR FILES (Real and/or Imaginary):

```
description line 1                                     80 chars
part                                                   80 chars
#                                                       1 int
element type                                           80 chars
s_e1 s_e2 ... s_ne                                     ne floats
element type                                           80 chars
 .
 .
part                                                   80 chars
 .
 .
part                                                   80 chars
#                                                       1 int
block                    # nn = (i-1)*(j-1)*(k-1)      80 chars
s_n1 s_n2 ... s_nn                                     nn floats
```

COMPLEX VECTOR FILES (Real and/or Imaginary):

```
description line 1                                     80 chars
part                                                   80 chars
#                                                       1 int
element type                                           80 chars
vx_e1 vx_e2 ... vx_ne                                  ne floats
vy_e1 vy_e2 ... vy_ne                                  ne floats
vz_e1 vz_e2 ... vz_ne                                  ne floats
element type                                           80 chars
 .
 .
part                                                   80 chars
 .
 .
part                                                   80 chars
#                                                       1 int
block                    # nn = (i-1)*(j-1)*(k-1)      80 chars
vx_n1 vx_n2 ... vx_nn                                  nn floats
vy_n1 vy_n2 ... vy_nn                                  nn floats
vz_n1 vz_n2 ... vz_nn                                  nn floats
```

Fortran Binary form:

SCALAR FILE:

```
'description line 1'                                  80 chars
'part'                                                80 chars
'#'                                                    1 int
'element type'                                        80 chars
's_e1 s_e2 ... s_ne'                                  ne floats
'element type'                                        80 chars
 .
 .
'part'                                                80 chars
 .
 .
'part'                                                80 chars
'#'                                                    1 int
'block'               # nn = (i-1)*(j-1)*(k-1)        80 chars
's_n1 s_n2 ... s_nn'                                  nn floats
```

VECTOR FILE:

```
'description line 1'                                  80 chars
'part'                                                80 chars
'#'                                                    1 int
'element type'                                        80 chars
'vx_e1 vx_e2 ... vx_ne'                               ne floats
'vy_e1 vy_e2 ... vy_ne'                               ne floats
'vz_e1 vz_e2 ... vz_ne'                               ne floats
'element type'                                        80 chars
 .
 .
'part'                                                80 chars
 .
 .
'part'                                                80 chars
'#'                                                    1 int
'block'               # nn = (i-1)*(j-1)*(k-1)        80 chars
'vx_n1 vx_n2 ... vx_nn'                               nn floats
'vy_n1 vy_n2 ... vy_nn'                               nn floats
'vz_n1 vz_n2 ... vz_nn'                               nn floats
```

TENSOR FILE:

```
'description line 1'                                  80 chars
'part'                                                80 chars
'#'                                                    1 int
'element type'                                        80 chars
'v11_e1 v11_e2 ... v11_ne'                            ne floats
'v22_e1 v22_e2 ... v22_ne'                            ne floats
'v33_e1 v33_e2 ... v33_ne'                            ne floats
'v12_e1 v12_e2 ... v12_ne'                            ne floats
'v13_e1 v13_e2 ... v13_ne'                            ne floats
'v23_e1 v23_e2 ... v23_ne'                            ne floats
'element type'                                        80 chars
 .
 .
'part'                                                80 chars
 .
 .
'part'                                                80 chars
'#'                                                    1 int
```

```
'block'                  # nn = (i-1)*(j-1)*(k-1)           80 chars
'v11_n1 v11_n2 ... v11_nn'                                  nn floats
'v22_n1 v22_n2 ... v22_nn'                                  nn floats
'v33_n1 v33_n2 ... v33_nn'                                  nn floats
'v12_n1 v12_n2 ... v12_nn'                                  nn floats
'v13_n1 v13_n2 ... v13_nn'                                  nn floats
'v23_n1 v23_n2 ... v23_nn'                                  nn floats
```

TENSOR9 FILE:

```
'description line 1'                                        80 chars
'part'                                                      80 chars
'#'                                                          1 int
'element type'                                              80 chars
'v11_e1 v11_e2 ... v11_ne'                                  ne floats
'v12_e1 v12_e2 ... v12_ne'                                  ne floats
'v13_e1 v13_e2 ... v13_ne'                                  ne floats
'v21_e1 v21_e2 ... v21_ne'                                  ne floats
'v22_e1 v22_e2 ... v22_ne'                                  ne floats
'v23_e1 v23_e2 ... v23_ne'                                  ne floats
'v31_e1 v31_e2 ... v31_ne'                                  ne floats
'v32_e1 v32_e2 ... v32_ne'                                  ne floats
'v33_e1 v33_e2 ... v33_ne'                                  ne floats
'element type'                                              80 chars
 .
 .
'part'                                                      80 chars
 .
 .
'part'                                                      80 chars
'#'                                                          1 int
'block'                  # nn = (i-1)*(j-1)*(k-1)           80 chars
'v11_n1 v11_n2 ... v11_nn'                                  nn floats
'v12_n1 v12_n2 ... v12_nn'                                  nn floats
'v13_n1 v13_n2 ... v13_nn'                                  nn floats
'v21_n1 v21_n2 ... v21_nn'                                  nn floats
'v22_n1 v22_n2 ... v22_nn'                                  nn floats
'v23_n1 v23_n2 ... v23_nn'                                  nn floats
'v31_n1 v31_n2 ... v31_nn'                                  nn floats
'v32_n1 v32_n2 ... v32_nn'                                  nn floats
'v33_n1 v33_n2 ... v33_nn'                                  nn floats
```

COMPLEX SCALAR FILES (Real and/or Imaginary):

```
'description line 1'                                        80 chars
'part'                                                      80 chars
'#'                                                          1 int
'element type'                                              80 chars
's_e1 s_e2 ... s_ne'                                        ne floats
'element type'                                              80 chars
 .
 .
'part'                                                      80 chars
 .
 .
'part'                                                      80 chars
'#'                                                          1 int
'block'                  # nn = (i-1)*(j-1)*(k-1)           80 chars
```

```
's_n1 s_n2 ... s_nn'                                   nn floats
```

COMPLEX VECTOR FILES (Real and/or Imaginary):

```
'description line 1'                                   80 chars
'part'                                                 80 chars
'#'                                                     1 int
'element type'                                         80 chars
'vx_e1 vx_e2 ... vx_ne'                                ne floats
'vy_e1 vy_e2 ... vy_ne'                                ne floats
'vz_e1 vz_e2 ... vz_ne'                                ne floats
'element type'                                         80 chars
 .
 .
'part'                                                 80 chars
 .
 .
'part'                                                 80 chars
'#'                                                     1 int
'block'              # nn = (i-1)*(j-1)*(k-1)          80 chars
'vx_n1 vx_n2 ... vx_nn'                                nn floats
'vy_n1 vy_n2 ... vy_nn'                                nn floats
'vz_n1 vz_n2 ... vz_nn'                                nn floats
```

ASCII form:

SCALAR FILE:

```
description line 1                                     A (max of 80 typ)
part                                                   A
#                                                      I10
element type                                           A
s_e1                                                   12.5  1/line (ne)
s_e2
 .
 .
s_ne
element type                                           A
 .
 .
part                                                   A
 .
 .
part                                                   A
#                                                      I10
block                # nn = (i-1)*(j-1)*(k-1)          A
s_n1                                                   E12.5  1/line (nn)
s_n2
 .
 .
s_nn
```

VECTOR FILE:

```
description line 1                                     A (max of 80 typ)
part                                                   A
#                                                      I10
element type                                           A
```

```
vx_e1                                                E12.5  1/line (ne)
vx_e2
 .
 .
vx_ne
vy_e1                                                E12.5  1/line (ne)
vy_e2
 .
 .
vy_ne
vz_e1                                                E12.5  1/line (ne)
vz_e2
 .
 .
vz_ne
element type                                         A
 .
 .
part                                                 A
 .
 .
part                                                 A
#                                                    I10
block                    # nn = (i-1)*(j-1)*(k-1)    A
vx_n1                                                E12.5  1/line (nn)
vx_n2
 .
 .
vx_nn
vy_n1                                                E12.5  1/line (nn)
vy_n2
 .
 .
vy_nn
vz_n1                                                E12.5  1/line (nn)
vz_n2
 .
 .
vz_nn
```

TENSOR FILE:

```
description line 1                                   A (max of 80 typ)
part                                                 A
#                                                    I10
element type                                         A
v11_e1                                               E12.5  1/line (ne)
v11_e2
 .
 .
v11_ne
v22_e1                                               E12.5  1/line (ne)
v22_e2
 .
 .
v22_ne
v33_e1                                               E12.5  1/line (ne)
v33_e2
 .
 .
v33_ne
```

```
v12_e1                                        E12.5  1/line (ne)
v12_e2
 .
 .
v12_ne
v13_e1                                        E12.5  1/line (ne)
v13_e2
 .
 .
v13_ne
v23_e1                                        E12.5  1/line (ne)
v23_e2
 .
 .
v23_ne
element type                                  A
 .
 .
part                                          A
 .
 .
part                                          A
#                                             I10
block                 # nn = (i-1)*(j-1)*(k-1)  A
v11_n1                                        E12.5  1/line (nn)
v11_n2
 .
 .
v11_nn
v22_n1                                        E12.5  1/line (nn)
v22_n2
 .
 .
v22_nn
v33_n1                                        E12.5  1/line (nn)
v33_n2
 .
 .
v33_nn
v12_n1                                        E12.5  1/line (nn)
v12_n2
 .
 .
v12_nn
v13_n1                                        E12.5  1/line (nn)
v13_n2
 .
 .
v13_nn
v23_n1                                        E12.5  1/line (nn)
v23_n2
 .
 .
v23_nn
```

TENSOR9 FILE:

```
description line 1                            A (max of 80 typ)
part                                          A
#                                             I10
element type                                  A
```

```
v11_e1                                      E12.5  1/line (ne)
v11_e2
 .
 .
v11_ne
v12_e1                                      E12.5  1/line (ne)
v12_e2
 .
 .
v12_ne
v13_e1                                      E12.5  1/line (ne)
v13_e2
 .
 .
v13_ne
v21_e1                                      E12.5  1/line (ne)
v21_e2
 .
 .
v21_ne
v22_e1                                      E12.5  1/line (ne)
v22_e2
 .
 .
v22_ne
v23_e1                                      E12.5  1/line (ne)
v23_e2
 .
 .
v23_ne
v31_e1                                      E12.5  1/line (ne)
v31_e2
 .
 .
v31_ne
v32_e1                                      E12.5  1/line (ne)
v32_e2
 .
 .
v32_ne
v33_e1                                      E12.5  1/line (ne)
v33_e2
 .
 .
v33_ne
element type                                A
 .
 .
part                                        A
 .
 .
part                                        A
#                                           I10
block                # nn = (i-1)*(j-1)*(k-1)    A
v11_n1                                      E12.5  1/line (nn)
v11_n2
 .
 .
v11_nn
v12_n1                                      E12.5  1/line (nn)
v12_n2
```

```
 .
 .
v12_nn
v13_n1                                                        E12.5  1/line (nn)
v13_n2
 .
 .
v13_nn
v21_n1                                                        E12.5  1/line (nn)
v21_n2
 .
 .
v21_nn
v22_n1                                                        E12.5  1/line (nn)
v22_n2
 .
 .
v22_nn
v23_n1                                                        E12.5  1/line (nn)
v23_n2
 .
 .
v23_nn
v31_n1                                                        E12.5  1/line (nn)
v31_n2
 .
 .
v31_nn
v32_n1                                                        E12.5  1/line (nn)
v32_n2
 .
 .
v32_nn
v33_n1                                                        E12.5  1/line (nn)
v33_n2
 .
 .
v33_nn
```

COMPLEX SCALAR FILES (Real and/or Imaginary):

```
description line 1                                            A (max of 80 typ)
part                                                          A
#                                                             I10
element type                                                  A
s_e1                                                          12.5  1/line (ne)
s_e2
 .
 .
s_ne
element type                                                  A
 .
 .
part                                                          A
 .
 .
part                                                          A
#                                                             I10
block                  # nn = (i-1)*(j-1)*(k-1)               A
```

```
s_n1                                              E12.5  1/line (nn)
s_n2
 .
 .
s_nn
```

COMPLEX VECTOR FILES (Real and/or Imaginary):

```
description line 1                                A (max of 80 typ)
part                                              A
#                                                 I10
element type                                      A
vx_e1                                             E12.5  1/line (ne)
vx_e2
 .
 .
vx_ne
vy_e1                                             E12.5  1/line (ne)
vy_e2
 .
 .
vy_ne
vz_e1                                             E12.5  1/line (ne)
vz_e2
 .
 .
vz_ne
element type                                      A
 .
 .
part                                              A
 .
 .
part                                              A
#                                                 I10
block                    # nn = (i-1)*(j-1)*(k-1) A
vx_n1                                             E12.5  1/line (nn)
vx_n2
 .
 .
vx_nn
vy_n1                                             E12.5  1/line (nn)
vy_n2
 .
 .
vy_nn
vz_n1                                             E12.5  1/line (nn)
vz_n2
 .
 .
vz_nn
```

The following variable file examples reflect scalar, vector, tensor, and complex variable values *per element* for the previously defined EnSight Gold Geometry File Example with 11 defined unstructured nodes and a 2x3x2 structured Part (Part number 3). The values are summarized in the following table

Note: These are the same values as listed in the EnSight6 per_element variable file section. Subsequently, the following example files contain the same data as the example files in the EnSight6 section - only they are listed in gold format. (No asymmetric tensor example data given)

		Element Index	Element Id	Scalar Value	Vector Values	Tensor (2nd order symm.) Values	Complex Scalar Real Value	Complex Scalar Imaginary Value
Unstructured								
	bar2							
		1	101	(1.)	(1.1, 1.2, 1.3)	(1.1, 1.2, 1.3, 1.4, 1.5, 1.6)	(1.1)	(1.2)
	tria3							
		1	102	(2.)	(2.1, 2.2, 2.3)	(2.1, 2.2, 2.3, 2.4, 2.5, 2.6)	(2.1)	(2.2)
		2	103	(3.)	(3.1, 3.2, 3.3)	(3.1, 3.2, 3.3, 3.4, 3.5, 3.6)	(3.1)	(3.2)
	hexa8							
		1	104	(4.)	(4.1, 4.2, 4.3)	(4.1, 4.2, 4.3, 4.4, 4.5, 4.6)	(4.1)	(4.2)
Structured								
	block	1	1	(5.)	(5.1, 5.2, 5.3)	(5.1, 5.2, 5.3, 5.4, 5.5, 5.6)	(5.1)	(5.2)

Per_element (Scalar) Variable Example 1: This example shows an ASCII scalar file (`engold.Esca`) for the gold geometry example.

```
Per_elem scalar values for the EnSight Gold geometry example
part
         1
tria3
 2.00000E+00
 3.00000E+00
hexa8
 4.00000E+00
part
         2
bar2
 1.00000E+00
part
         3
block
 5.00000E+00
 6.00000E+00
```

Per_element (Vector) Variable Example 2: This example shows an ASCII vector file (`engold.Evec`) for the gold geometry example.

```
Per_elem vector values for the EnSight Gold geometry example
part
         1
tria3
 2.10000E+00
 3.10000E+00
 2.20000E+00
 3.20000E+00
```

```
 2.30000E+00
 3.30000E+00
hexa8
 4.10000E+00
 4.20000E+00
 4.30000E+00
part
         2
bar2
 1.10000E+00
 1.20000E+00
 1.30000E+00
part
         3
block
 5.10000E+00
 6.10000E+00
 5.20000E+00
 6.20000E+00
 5.30000E+00
 6.30000E+00
```

Per_element (Tensor) Variable Example3: This example shows an ASCII 2nd order symmetric tensor file (`engold.Eten`) for the gold geometry example.

```
Per_elem symmetric tensor values for the EnSight Gold geometry example
part
         1
tria3
 2.10000E+00
 3.10000E+00
 2.20000E+00
 3.20000E+00
 2.30000E+00
 3.30000E+00
 2.40000E+00
 3.40000E+00
 2.50000E+00
 3.50000E+00
 2.60000E+00
 3.60000E+00
hexa8
 4.10000E+00
 4.20000E+00
 4.30000E+00
 4.40000E+00
 4.50000E+00
 4.60000E+00
part
         2
bar2
 1.10000E+00
 1.20000E+00
 1.30000E+00
 1.40000E+00
 1.50000E+00
 1.60000E+00
part
         3
block
```

```
 5.10000E+00
 6.10000E+00
 5.20000E+00
 6.20000E+00
 5.30000E+00
 6.30000E+00
 5.40000E+00
 6.40000E+00
 5.50000E+00
 6.50000E+00
 5.60000E+00
 6.60000E+00
```

Per_element (Complex) Variable Example 4: This example shows ASCII complex real (`engold.Ecmp_r`) and imaginary (`engold.Ecmp_i`) *scalar* files for the gold geometry example. (The same methodology would apply for complex real and imaginary *vector* files.)

Real scalar File:

```
Per_elem complex real scalar values for the EnSight Gold geometry example
part
         1
tria3
 2.10000E+00
 3.10000E+00
hexa8
 4.10000E+00
part
         2
bar2
 1.10000E+00
part
         3
block
 5.10000E+00
 6.10000E+00
```

Imaginary scalar File:

```
Per_elem complex imaginary scalar values for the EnSight Gold geometry example
part
         1
tria3
 2.20000E+00
 3.20000E+00
hexa8
 4.20000E+00
part
         2
bar2
 1.20000E+00
part
         3
block
 5.20000E+00
 6.20000E+00
```

EnSight Gold Undefined Variable Values Format

Undefined variable values are allowed in EnSight Gold scalar, vector, tensor and complex variable file formats. Undefined values are specified on a "per section" basis (i.e. `coordinates, element_type`, or `block`) in each EnSight Gold variable file. EnSight first parses any undefined keyword "undef" that may follow the sectional keyword (i.e. `coordinates undef`, `element_type undef`, or `block undef`) on its line. This indicates that the next floating point value is the undefined value used in that section. EnSight reads this undefined value, reads all subsequent variable values for that section; and then converts any undefined (file section) values to an internal undefined value (currently -1.2345e-10) recognized computationally by EnSight (Note: the internal, or computational, undefined value can be changed by the user via the "`test: change_undef_value`" command **before any data is read**.)

Note: EnSight's undefined capability is for variables only - not for geometry! Also, in determining internally whether a vector or tensor variable is undefined at a node or element, the first component is all that is examined. You cannot have some components defined and others undefined.

The following per_node and per_element ASCII scalar files contain examples of undefined values. For your comparison, these two files are the files `engold.Nsca` and `engold.Esca` written with some undefined values specified. Note that the undefined values per section need not be the same value; rather, it may be any value - usually outside the interval range of the variable. *The same methodology applies to vector, tensor, and complex files*.

C Binary form: (Per_node)

SCALAR FILE:

```
description line 1                                    80 chars
part                                                  80 chars
#                                                      1 int
coordinates undef                                     80 chars
undef_value                                            1  float
s_n1 s_n2 ... s_nn                                    nn floats
part                                                  80 chars
 .
 .
part                                                  80 chars
#                                                      1 int
block undef                # nn = i*j*k               80 chars
undef_value                                            1 float
s_n1 s_n2 ... s_nn                                    nn floats
```

Fortran Binary form: (Per_node)

SCALAR FILE:

```
'description line 1'                                  80 chars
'part'                                                80 chars
'#'                                                    1 int
'coordinates undef'                                   80 chars
'undef_value'                                          1 float
's_n1 s_n2 ... s_nn'                                  nn floats
'part'                                                80 chars
```

```
 .
 .
'part'                                                     80 chars
'#'                                                         1 int
'block undef'                  # nn = i*j*k                80 chars
'undef_value'                                               1 float
's_n1 s_n2 ... s_nn'                                       nn floats
```

ASCII form: (Per_node)

SCALAR FILE:

```
description line 1                                         A (max of 79 typ)
part                                                       A
#                                                          I10
coordinates undef                                          A
undef_value                                                E12.5
s_n1                                                       E12.5  1/line (nn)
s_n2
 .
 .
s_nn
part                                                       A
 .
 .
part                                                       A
#                                                          I10
block undef                     # nn = i*j*k               A
undef_value                                                E12.5
s_n1                                                       E12.5  1/line (nn)
s_n2
 .
 .
s_nn
```

Undefined per_node (Scalar) Variable Example: This example shows undefined data in an ASCII scalar file (`engold.Nsca_u`) for the gold geometry example.

```
Per_node undefined scalar values for the EnSight Gold geometry example
part
         1
coordinates undef
-1.00000E+04
-1.00000E+04
 3.00000E+00
 4.00000E+00
 5.00000E+00
 6.00000E+00
 7.00000E+00
 8.00000E+00
 9.00000E+00
 1.00000E+01
 1.10000E+01
part
         2
coordinates
 1.00000E+00
 2.00000E+00
part
         3
```

```
block undef
-1.23450E-10
 1.00000E+00
 2.00000E+00
 3.00000E+00
 4.00000E+00
 5.00000E+00
-1.23450E-10
 7.00000E+00
 8.00000E+00
 9.00000E+00
 1.00000E+01
 1.10000E+01
 1.20000E+01
```

C Binary form: (Per_element)

SCALAR FILE:

```
description line 1                                          80 chars
part                                                        80 chars
#                                                            1 int
element type undef                                          80 chars
undef_value                                                  1 float
s_e1 s_e2 ... s_ne                                          ne floats
element type undef                                          80 chars
undef_value                                                  1 float
.
 .
part                                                        80 chars
 .
 .
part                                                        80 chars
#                                                            1 int
block undef                  # nn = (i-1)*(j-1)*(k-1)       80 chars
undef_value                                                  1 float
s_n1 s_n2 ... s_nn                                          nn floats
```

Fortran Binary form: (Per_element)

SCALAR FILE:

```
'description line 1'                                        80 chars
'part'                                                      80 chars
'#'                                                          1 int
'element type undef'                                        80 chars
'undef_value'                                                1 float
's_e1 s_e2 ... s_ne'                                        ne floats
'element type undef'                                        80 chars
'undef_value'                                                1 float
.
 .
'part'                                                      80 chars
 .
 .
```

```
'part'                                                   80 chars
'#'                                                       1 int
'block undef'              # nn = (i-1)*(j-1)*(k-1)      80 chars
'undef_value'                                             1 float
's_n1 s_n2 ... s_nn'                                     nn floats
```

ASCII form: (Per_element)

SCALAR FILE:

```
description line 1                                       A (max of 80 typ)
part                                                     A
#                                                        I10
element type undef                                       A
undef_value                                              E12.5
s_e1                                                     E12.5  1/line (ne)
s_e2
 .
 .
s_ne
element type undef                                       A
undef_value                                              E12.5
 .
 .
part                                                     A
 .
 .
part                                                     A
#                                                        I10
block undef                # nn = (i-1)*(j-1)*(k-1)      A
undef_value                                              E12.5
s_n1                                                     E12.5  1/line (nn)
s_n2
 .
 .
s_nn
```

Undefined per_element (Scalar) Variable Example: This example shows undefined data in an ASCII scalar file (`engold.Esca_u`) for the gold geometry example.

```
Per_elem undefined scalar values for the EnSight Gold geometry example
part
         1
tria3 undef
-1.00000E+02
 2.00000E+00
-1.00000E+02
hexa8
 4.00000E+00
part
         2
bar2
 1.00000E+00
part
         3
block undef
-1.23450E-10
-1.23450E-10
6.00000E+00
```

EnSight Gold Partial Variable Values Format

Partial variable values are allowed in EnSight Gold scalar, vector, tensor and complex variable file formats. Partial values are specified on a “per section” basis (i.e. `coordinates, element_type`, or `block`) in each EnSight Gold variable file. EnSight first parses any partial keyword “`partial`” that may follow the sectional keyword (i.e. `coordinates partial`, `element_type partial`, or `block partial`) on its line. This indicates that the next integer value is the number of partial values defined in that section. EnSight reads the number of defined partial values, next reads this number of integer partial indices, and finally reads all corresponding partial variable values for that section. Afterwords, any variable value not specified in the list of partial indices is assigned the internal “undefined” (see previous section) value. Values interpolated between time steps must be defined for both time steps; otherwise, they are undefined.

The following per_node and per_element ASCII scalar files contain examples of partial values. For your comparison, these two files are the files `engold.Nsca` and `engold.Esca` written with some partial values specified. The same methodology applies to vector, tensor, and complex files.

C Binary form: (Per_node)

SCALAR FILE:

```
description line 1                                            80 chars
part                                                          80 chars
#                                                              1 int
coordinates partial                                           80 chars
nn                                                             1 int
i_n1 i_n2 ... i_nn                                            nn  ints
s_n1 s_n2 ... s_nn                                            nn floats
part                                                          80 chars
 .
 .
part                                                          80 chars
#                                                              1 int
block partial                    # nn = i*j*k                 80 chars
nn                                                             1 int
i_n1 i_n2 ... i_nn                                            nn  ints
s_n1 s_n2 ... s_nn                                            nn floats
```

Fortran Binary form: (Per_node)

SCALAR FILE:

```
'description line 1'                                          80 chars
'part'                                                        80 chars
'#'                                                            1 int
'coordinates partial'                                         80 chars
'nn'                                                           1 int
'i_n1 i_n2 ... i_nn'                                          nn ints
's_n1 s_n2 ... s_nn'                                          nn floats
'part'                                                        80 chars
 .
```

```
 .
'part'                                                80 chars
'#'                                                    1 int
'block partial'                 # nn = i*j*k          80 chars
'nn'                                                   1 int
'i_n1 i_n2  ...  i_nn'                                nn ints
's_n1 s_n2 ... s_nn'                                  nn floats
```

ASCII form: (Per_node)

SCALAR FILE:

```
description line 1                                    A (max of 79 typ)
part                                                  A
#                                                     I10
coordinates partial                                   A
nn                                                    I10
i_n1                                                  I10  1/line (nn)
i_n2
 .
 .
i_nn
s_n1                                                  E12.5  1/line (nn)
s_n2
 .
 .
s_nn
part                                                  A
 .
 .
part                                                  A
#                                                     I10
block partial                   # nn = i*j*k          A
nn                                                    I10
i_n1                                                  I10  1/line (nn)
i_n2
 .
 .
i_nn
s_n1                                                  E12.5  1/line (nn)
s_n2
 .
 .
s_nn
```

Partial per_node (Scalar) Variable Example: This example shows partial data in an ASCII scalar file (`engold.Nsca_p`) for the gold geometry example.

```
Per_node partial scalar values for the EnSight Gold geometry example
part
         1
coordinates partial
         9
         2
         3
         4
         5
         6
         7
         8
```

```
         9
        10
 3.00000E+00
 4.00000E+00
 5.00000E+00
 6.00000E+00
 7.00000E+00
 8.00000E+00
 9.00000E+00
 1.00000E+01
 1.10000E+01
part
         2
coordinates
 1.00000E+00
 2.00000E+00
part
         3
block
 1.00000E+00
 2.00000E+00
 3.00000E+00
 4.00000E+00
 5.00000E+00
 6.00000E+00
 7.00000E+00
 8.00000E+00
 9.00000E+00
 1.00000E+01
 1.10000E+01
 1.20000E+01
```

C Binary form: (Per_element)

SCALAR FILE:

```
description line 1                                            80 chars
part                                                          80 chars
#                                                              1 int
element type partial                                          80 chars
ne                                                             1 int
i_n1 i_n2 ... i_ne                                            ne ints
s_e1 s_e2 ... s_ne                                            ne floats
element type partial                                          80 chars
ne                                                             1 int
i_n1 i_n2 ... i_ne                                            ne ints
 .
 .
part                                                          80 chars
 .
 .
part                                                          80 chars
#                                                              1 int
block partial                 # me= (i-1)*(j-1)*(k-1)         80 chars
me                                                             1 int
i_n1 i_n2 ... i_me                                            me ints
s_n1 s_n2 ... s_me                                            me floats
```

Fortran Binary form: (Per_element)

SCALAR FILE:

```
'description line 1'                                80 chars
'part'                                              80 chars
'#'                                                  1 int
'element type partial'                              80 chars
'ne'                                                 1 int
'i_n1 i_n2 ... i_ne'                                ne ints
's_e1 s_e2 ... s_ne'                                ne floats
'element type partial'                              80 chars
'ne'                                                 1 int
'i_n1 i_n2 ... i_ne'                                ne ints
  .
  .
'part'                                              80 chars
  .
 .
'part'                                              80 chars
'#'                                                  1 int
'block partial'            # me = (i-1)*(j-1)*(k-1) 80 chars
'me'                                                 1 int
'i_n1 i_n2 ... i_me'                                me ints
's_n1 s_n2 ... s_me'                                me floats
```

ASCII form: (Per_element)

SCALAR FILE:

```
description line 1                                  A (max of 80 typ)
part                                                A
#                                                   I10
element type partial                                A
ne                                                  I10
i_n1                                                I10  1/line  (ne)
i_n2
 .
 .
i_ne
s_e1                                                E12.5  1/line (ne)
s_e2
 .
 .
s_ne
element type partial                                A
ne                                                  I10
i_n1                                                I10  1/line  (ne)
i_n2
 .
 .
i_ne
 .
 .
part                                                A
 .
 .
part                                                A
```

```
#                                                       I10
block partial                # me = (i-1)*(j-1)*(k-1)   A
me                                                      I10
i_n1                                                    I10  1/line  (me)
i_n2
 .
 .
i_me
s_n1                                                    E12.5  1/line (me)
s_n2
 .
 .
s_me
```

Partial per_element (Scalar) Variable Example: This example shows partial data in an ASCII scalar file (`engold.Esca_p`) for the gold geometry example.

```
Per_elem partial scalar values for the EnSight Gold geometry example
part
         1
tria3 partial
         1
         1
 2.00000E+00
hexa8
 4.00000E+00
part
         2
bar2
 1.00000E+00
part
         3
block partial
         1
         2
 6.00000E+00
```

EnSight Gold Measured/Particle File Format

Changes

Measured/Particle file formats for both geometry and variables have remained unchanged since EnSight 5. The only change is the contents of EnSight 5 results file (.mea suffix) containing geometry and variable filenames and time values are now entered directly into the EnSight Gold Case file.

While the format of a Measured/Particle geometry file is exactly the same as the EnSight 5& 6 geometry file, it is repeated below for convenience:

- Line 1

  This line is a description line.

- Line 2

  Indicates that this file contains particle coordinates. The words "`particle coordinates`" should be entered on this line without the quotes.

- Line 3

  Specifies the number of Particles.

- Line 4 through the end of the file.

  Each line contains the ID and the X, Y, and Z coordinates of each Particle. The format of this line is "integer real real real" written out in the following format:

  From C: `%8d%12.5e%12.5e%12.5e` format

  From FORTRAN: `i8, 3e12.5` format

A generic measured/Particle geometry file is as follows:

```
A description line
particle coordinates
#_of_Particles
id xcoord ycoord zcoord
id xcoord ycoord zcoord
id xcoord ycoord zcoord
 .
 .
 .
```

Measured Geometry Example

The following illustrates a measured/Particle file with seven points:

```
This is a simple measured geometry file
particle coordinates
       7
     101 0.00000E+00 0.00000E+00 0.00000E+00
     102 1.00000E+00 0.00000E+00 0.00000E+00
     103 1.00000E+00 1.00000E+00 0.00000E+00
     104 0.00000E+00 1.00000E+00 0.00000E+00
     205 5.00000E-01 0.00000E+00 2.00000E+00
     206 5.00000E-01 1.00000E+00 2.00000E+00
     307 0.00000E+00 0.00000E+00-1.50000E+00
```

Measured Variable Files

Measured variable files have remained unchanged since EnSight 5.
The particle variable file is also the same as **EnSight6 case** per_node variable files.
Please note that they are NOT the same as the EnSight gold per_node variable files. *(see EnSight6 Per_Node Variable File Format, in Section 11.2)*

EnSight Gold Material Files Format

This section contains descriptions of the three EnSight Gold material files; i.e. material id file, mixed-material id file, and mixed-material values file. A simple example dataset is also appended for quick reference.

All three EnSight Gold material files correlate to and follow the same syntax of the other EnSight Gold file formats.

Material Id File

The material id file follows the same syntax as the per_element variable files, except that its values are integers for each element of designated types of designated parts. First comes a single description line. Second comes a Part line. Third comes a line containing the part number. Fourth comes an element type line ***(Note, this is the only material file that has an element type line)***. And then comes the corresponding integer value for each element of that type and part (and so on for each part).

The integer value is either positive or negative. A positive integer is the material number/id for the entire element. A negative integer indicates that this element is composed of multiple, or mixed, materials. The absolute value of this negative number is a relative (1-bias) index into the mixed ids file that points to the mixed material data for each element under its part (see example below).

Mixed (Material) Ids File

The mixed-material id file also contains integer values, and follows EnSight Gold syntax with exceptions as noted below. First comes a single description line. Second comes a Part line. Third comes a line containing the part number. Fourth comes a “mixed ids” keyword line. Fifth comes the size of the total mixed id array for all the mixed elements of this part. Next comes the mixed id element data for each of the elements with mixed materials for this part (and so on for each part).

The mixed id data for each of the “mixed elements” has the following order of syntax. First comes the number of mixed materials. Second comes a list of material ids that comprise that element. Next comes a negative number whose absolute value is a relative (1-bias) index into the mixed values file that points to the group of mixed-material fraction values that correspond to each listed material of that element under its part (see example below).

Mixed (Material) Values File

The mixed-material values file contains float values, and also follows EnSight Gold syntax with exceptions as noted below. First comes a single description line. Second comes a Part line. Third comes a line containing the part number. Fourth comes a “mixed values” keyword line. Fifth comes the size of the total mixed values array for all the mixed elements of this part. Next comes the mixed material fraction values whose order corresponds to the order of the material ids listed for that element in the mixed ids file.

Species (Element) Values File

This is the same as the Mixed (Material) Values file except replace “mixed values” with “species “values” and “mixval” with “spval.”

Materials via Per Element Scalar Variables

Materials defined by per element scalar variables are specified in the case file under the MATERIAL section. The per element scalar files are the same format as defined under the variable files subsection for scalar variable per element.

C Binary form:

MATERIAL ID FILE:

```
description line 1                                            80 chars
part                                                          80 chars
#                                                             1 int
element type                                                  80 chars
matid_e1 matid_e2 ... matid_ne                                ne ints
element type                                                  80 chars
.
.
part                                                          80 chars
.
.
part                                                          80 chars
#                                                             1 int
block      # nbe = (i-1)*(j-1)*(k-1)                          80 chars
matid_e1 matid_e2 ... matid_nbe                               nbe ints
```

MIXED IDS FILE:

```
description line 1                                            80 chars
part                                                          80 chars
#                                                             1 int
mixed ids                                                     80 chars
ni                                                            1 int
mixid_1 mixid_2 ... mixid_ni                                  ni ints
.
.
part                                                          80 chars
.
.
part                                                          80 chars
#                                                             1 int
mixed ids                                                     80 chars
ni                                                            1 int
mixid_1 mixid_2 ... mixid_ni                                  ni ints
```

MIXED VALUES FILE:

```
description line 1                                            80 chars
part                                                          80 chars
#                                                             1 int
mixed values                                                  80 chars
nf                                                            1 int
mixval_1 mixval_2 ... mixval_nf                               nf floats
.
.
part                                                          80 chars
.
.
part                                                          80 chars
#                                                             1 int
mixed values                                                  80 chars
nf                                                            1 int
mixval_1 mixval_2 ... mixval_nf                               nf floats
```

SPECIES VALUES FILE:

```
same as Mixed Vaues File above except...
replace “mixed values” with “species values”
and “mixval” with “spval”
```

MATERIALS VIA PER ELEMENT SCALAR VARIABLES:

```
same as under EnSight Gold Per-Element Variable File Format
```

Fortran Binary form:

MATERIAL ID FILE:

```
'description line 1'                              80 chars
'part'                                            80 chars
'#'                                               1 int
'element type'                                    80 chars
'matid_e1 matid_e2 ... matid_ne'                  ne ints
'element type'                                    80 chars
.
.
'part'                                            80 chars
.
.
'part'                                            80 chars
'#'                                               1 int
'block'    # nbe = (i-1)*(j-1)*(k-1)              80 chars
'matid_e1 matid_e2 ... matid_nbe'                 nbe ints
```

MIXED IDS FILE:

```
'description line 1'                              80 chars
'part'                                            80 chars
'#'                                               1 int
'mixed ids'                                       80 chars
'ni'                                              1 int
'mixid_1 mixid_2 ... mixid_ni'                    ni ints
.
.
'part'                                            80 chars
.
.
'part'                                            80 chars
'#'                                               1 int
'mixed ids'                                       80 chars
'ni'                                              1 int
'mixid_1 mixid_2 ... mixid_ni'                    ni ints
```

MIXED VALUES FILE:

```
'description line 1'                              80 chars
'part'                                            80 chars
'#'                                               1 int
'mixed values'                                    80 chars
'nf'                                              1 int
'mixval_1 mixval_2 ... mixval_nf'                 nf floats
.
.
'part'                                            80 chars
.
.
'part'                                            80 chars
 '#'                                              1 int
'mixed values'                                    80 chars
'nf'                                              1 int
'mixval_1 mixval_2 ... mixval_nf'                 nf floats
```

SPECIES VALUES FILE:

```
same as Mixed Vaues File above except...
replace "mixed values" with "species values"
and "mixval" with "spval"
```

MATERIALS VIA PER ELEMENT SCALAR VARIABLES:

```
same as under EnSight Gold Per-Element Variable File Format
```

ASCII form:

MATERIAL ID FILE:

```
description line 1                                          A (max of 79 typ)
part                                                        A
#                                                           I10
element type                                                A
matid_e1                                                    I10 1/line (ne)
matid_e2
...
matid_ne
element type                                                A
.
.
part                                                        A
.
.
part                                                        A
#                                                           I10
block      # nbe = (i-1)*(j-1)*(k-1)                        A
matid_e1                                                    I10 1/line (nbe)
matid_e2
...
matid_nbe
```

MIXED IDS FILE:

```
description line 1                                          A (max of 79 typ)
part                                                        A
#                                                           I10
mixed ids                                                   A
ni                                                          I10
mixid_1                                                     I10 1/line (ni)
mixid_2
...
mixid_ni . .
part                                                        A
.
.
part                                                        A
#                                                           I10
mixed ids                                                   A
ni                                                          I10
mixid_1                                                     I10 1/line (ni)
mixid_2
...
mixid_ni
```

MIXED VALUES FILE:

```
description line 1                                          A (max of 80 typ)
part                                                        A
#                                                           I10
mixed values                                                A
nf                                                          I10
mixval_1                                                    E12.5 1/line (nf)
mixval_2
...
mixval_nf
.
.
part                                                        A
.
.
part                                                        A
#                                                           I10
mixed values                                                A
nf                                                          I10
mixval_1                                                    E12.5 1/line (nf)
mixval_2
...
mixval_nf
```

SPECIES VALUES FILE:

```
same as Mixed Vaues File above except...
replace “mixed values” with “species values”
and “mixval” with “spval”
```

MATERIALS VIA PER ELEMENT SCALAR VARIABLES:

```
same as under EnSight Gold Per-Element Variable File Format
```

Example Material Dataset (without species)

The following example dataset of ASCII EnSight Gold geometry and material files show the definition of material fractions for an unstructured model.

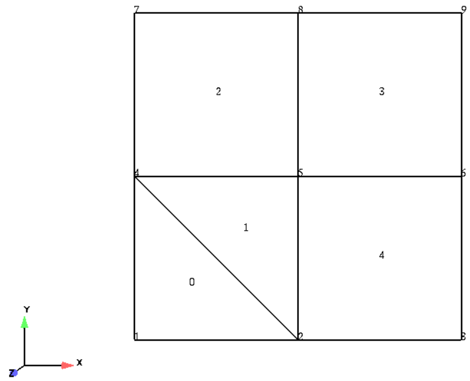


Figure 11-2
Geometry for Example Material Dataset

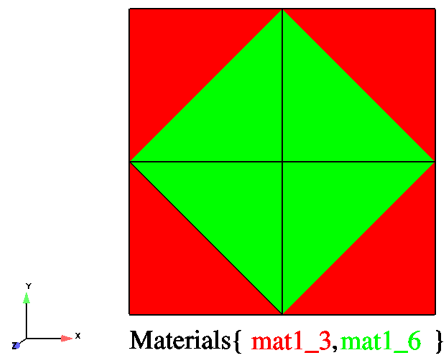


Figure 11-2
Materials for Example Material Dataset

Case file

```
# Sample Case File for 2D Material Dataset
# Created: 03Apr03:mel
#
FORMAT
type:  ensight gold

GEOMETRY
model:                    zmat2d.geo

VARIABLE
scalar per node:          scalar zmat2d.sca

MATERIAL
material set number:      1 Mat1
material id count:        2
material id numbers:      3 6
material id names:        mat1_3 mat1_6  #Air H2O
material id per element:  zmat2d.mati
material mixed ids:       zmat2d.mixi
material mixed values:    zmat2d.mixv

#       y
#
#       ^
#       |  Case Material ids = {3,6}
#
#   6.  7----------8----------9
#       |         /|\         |
#       |        / | \        |
#       |   e2  /  |  \ e3    |
#       |      /   |   \      |
#       |     /    |    \     |
#       |   q0     |     q1   |
#       |   /      |      \   |
#       | {.5,.5}  | {.5,.5}  |
#       |  /       |        \ |
#       | /        |         \|
#       |/         |          \
#   3.  4----------5----------6
#       |\         |         /|
#       | \ {0.,1.}|        / |
#       |  \       |  e4   /  |
#       |   \  t1  |      /   |
#       |    \     |     /    |
#       | e0  \ e1 |   q2     |
#       |      \   |   /      |
#       | t0    \  | {.5,.5}  |
#       |        \ | /        |
#       | {1.,0.} \|/         |
#       |          |          |
#   0.  1----------2----------3 -> x
#
#       0.         3.         6.
#
```

Geometry File (zmat2d.geo)

```
Geometry file
Example 2D Material Dataset
node id given
element id given
part
         1
2d-mesh
coordinates
         9
         1
         2
         3
         4
         5
         6
         7
         8
         9
 0.00000e+00
 3.00000e+00
 6.00000e+00
 0.00000e+00
 3.00000e+00
 6.00000e+00
 0.00000e+00
 3.00000e+00
 6.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 3.00000e+00
 3.00000e+00
 3.00000e+00
 6.00000e+00
 6.00000e+00
 6.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
 0.00000e+00
tria3
         2
         0
         1
         1         2         4
         2         5         4
quad4
         3
         2
         3
         4
         4         5         8         7
         8         5         6         9
         2         3         6         5
```

Material Number/ID File (zmat2d.mati)

```
Material Number/Id File
part
         1
tria3
         3
         6
quad4
        -1
        -5
        -9
```

Mixed Material Ids File (zmat2d.mixi)

```
Mixed Ids File
part
         1
mixed ids
        12
         2
         3
         6
        -1
         2
         3
         6
        -3
         2
         3
         6
        -5
```

Mixed Material Values File (zmat2d.mixv)

```
Mixed Values File
part
         1
mixed values
         6
 0.50000e+00
 0.50000e+00
 0.50000e+00
 0.50000e+00
 0.50000e+00
 0.50000e+00
```

Scalar File (zmat2d.sca)

```
Scalar File
part
        1
coordinates
 0.00000e+00
 1.00000e+00
 2.00000e+00
 1.00000e+00
 2.00000e+00
 3.00000e+00
 2.00000e+00
 3.00000e+00
 4.00000e+00
```

Example Material Dataset (with Species)

Same as previous Material Dataset example except:

-append species information to MATERIAL section in case file, and
-add species element values file
(see example below).

Case file

```
# Sample Case File for 2D Material Dataset - with Speicies
# Created: 03Apr05:mel
#
FORMAT
type:                        ensight gold

GEOMETRY
Model:                       zmat2d.geo

VARIABLE
scalar per node:             Nscalarzmat2d.sca
scalar per element:          Escalarzmat2d.sca

MATERIAL
material set number:         1 Mat1
material id count:           2
material id numbers:         3 6
material id names:           Air H2O
material id per element:     zmat2d.mati
material mixed ids:          zmat2d.mixi
material mixed values:       zmat2d.mixv
                             # Optional Species data
species id count:            4
species id numbers:          11 12 13 14
species id names:            Hydrogen Nitrogen Oxygen Argon
species per material counts: 3 2
species per material lists:  12 13 14 11 13
species element values:      zmat2d.spv
```

Note: The following relationships are significant in the above case file.

```
material id numbers:              3      6
material id names:               Air    H20
species per material counts:      3      2
                              ----^--- --^--
species per material lists:   12 13 14 11 13
                               N  O Ar  H  O
```

where:

Case material ids (w/species) = {3,6} = {Air, $H_2$0} = {Air{ N, O, Ar), $H_2$0(H, O}}

Note: typical values " {Air(.78,.21.01), $H_2$0(.33,.67)}

```
 M
 a
 t   S
 e   p   Species Element Values File (zmat2d.spv)
 r   e   Species Element Values File
 i   c   part
 a   i            1                   0-bias  Element   Element by type in
 l   e   species values                Index   Label   the connectivity list
=== ===          20                    =====  =======  =====================
Air  N    0.78000e+00   <----------------  0  <---- e0 = t0 = 1st Triangle
 "   O    0.21000e+00                      1
 "   Ar   0.01000e+00                      2
H2O  H    0.33000e+00   <----------------  3  <---- e1 = t1 = 2nd Triangle
 "   O    0.67000e+00                      4
Air  N    0.78100e+00   <----------------  5  <---- e2 = q0 = 1st Quad
 "   O    0.20900e+00                      6
 "   Ar   0.01000e+00                      6
H2O  H    0.33300e+00                      8
 "   O    0.66700e+00                      9
Air  N    0.78010e+00   <---------------- 10  <---- e3 = q1 = 2nd Quad
 "   O    0.20990e+00                     11
 "   Ar   0.01000e+00                     12
H2O  H    0.33330e+00                     13
 "   O    0.66670e+00                     14
Air  N    0.78001e+00   <---------------- 15  <---- e4 = q2 = 3rd Quad
 "   O    0.20999e+00                     16
 "   Ar   0.01000e+00                     17
H2O  H    0.33333e+00                     18
 "   O    0.66667e+00                     19
```

Note: The above species element values are accessed via part-element connectivity (`zmat2d.geo`) and material element data (`zmat2d.mati` and `zmat2d.mixi`) as illustrated above.

For more information on species see Species, located in Material Parts Create/Update, MATERIAL Section under EnSight Gold Casefile Format, and MatSpecies function under 4.3 Variable Creation.

Example Material Dataset With Materials Defined By Per Element Scalar Variables

Note: Species are not supported under this format.

The following example dataset of ASCII EnSight Gold geometry and material files show the definition of material fractions for an unstructured model.

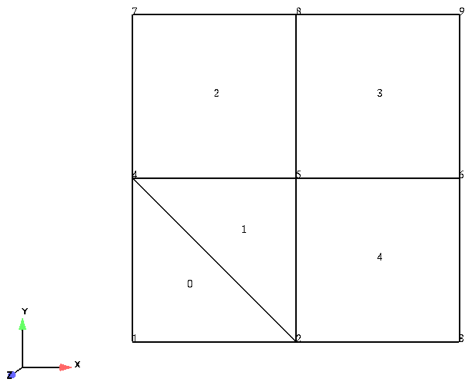


Figure 11-3
Geometry for Example Material Dataset

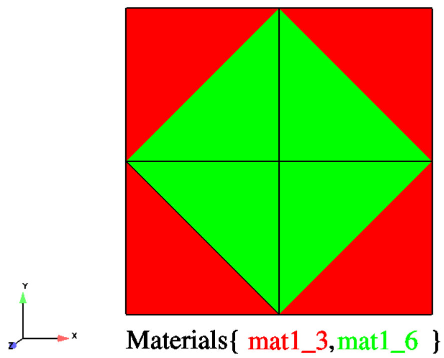


Figure 11-3
Materials for Example Material Dataset

Case file

```
# Sample Case File for 2D Material Dataset w/materials via per element scalars
# Created: 07Feb20:mel
#
FORMAT
type:                     ensight gold

GEOMETRY
model:                    zmat2d.geo

VARIABLE
scalar per node:          scalarN zmat2d.sca
scalar per element:       scal_E3 zmat2d.se3
scalar per element:       scal_E6 zmat2d.se6

MATERIAL
material set number:      1 Mat1
material id count:        2
material id numbers:      3 6
material id names:        matl_3 mat1_6  #Air H2O
material scalars per element:  scal_E3 scal_E6

#       y
#
#       ^
#       |  Case Material ids = {3,6}
#
#   6.  7----------8----------9
#       |         /|\         |
#       |        / | \        |
#       |   e2  /  |  \  e3   |
#       |      /   |   \      |
#       |     /    |    \     |
#       |   q0     |     q1   |
#       |   /      |      \   |
#       | {.5,.5}  |  {.5,.5} |
#       |  /       |       \  |
#       | /        |        \ |
#       |/         |         \|
#   3.  4----------5----------6
#       |\         |         /|
#       | \ {0.,1.}|        / |
#       |  \       |  e4   /  |
#       |   \  t1  |      /   |
#       |    \     |     /    |
#       | e0  \ e1 |   q2     |
#       |      \   |   /      |
#       | t0    \  | {.5,.5}  |
#       |        \ |  /       |
#       |{1.,0.}  \| /        |
#       |         \|/         |
#   0.  1----------2----------3 -> x
#
#       0.         3.         6.
```

zmat2d.geo and zmat2d.sca are the same as in the Material Dataset (without species) example above.

zmat2d.se3 file contents:

```
Scalar3 File
part
         1
tria3
 1.00000e+00
 0.00000e+00
quad4
 5.00000e-01
 5.00000e-01
 5.00000e-01
```

zmat2d.se6 file contents:

```
Scalar6 File
part
         1
tria3
 0.00000e+00
 1.00000e+00
quad4
 5.00000e-01
 5.00000e-01
 5.00000e-01
```

11.2 EnSight6 Casefile Format

Included in this section:

EnSight6 General Description

EnSight6 data consists of the following files:

- Case (required) (points to all other needed files including model geometry, variables, and possibly measured geometry and variables)

EnSight6 supports constant result values as well as scalar, vector, 2nd order symmetric tensor, and complex variable fields.

EnSight makes no assumptions regarding the physical significance of the variable values in the files. These files can be from any discipline. For example, the scalar file can include such things as pressure, temperature, and stress. The vector file can be velocity, displacement, or any other vector data. And so on.

All variable results for EnSight6 are contained in disk files—one variable per file. Additionally, if there are multiple time steps, there must either be a set of disk files for each time step (transient multiple-file format), or all time steps of a particular variable or geometry in one disk file (transient single-file format). Thus, all EnSight6 transient geometry and variable files can be expressed in either multiple file format or single file format.

Sources of EnSight6 data include the following:

- Data that can be translated to conform to the EnSight6 data format
- Data that originates from one of the translators supplied with the EnSight application

The EnSight6 format supports an unstructured defined element set as shown in the figure on the following page. Unstructured data must be defined in this element set. Elements that do not conform to this set must either be subdivided or discarded. The EnSight6 format also supports a structured block data format which is very similar to the PLOT3D format. *For the structured format, the standard order of nodes is such that I's advance quickest, followed by J's, and then K's.* A given EnSight6 model may have either unstructured data, structured data, or a mixture of both.

ens_checker

A program is supplied with EnSight which attempts to verify the integrity of the *format* of EnSight 6 and EnSight Gold files. If you are producing EnSight formatted data, this program can be very helpful, especially in your development stage, in making sure that you are adhering to the published format. It makes no attempt to verify the validity of floating point values, such as coordinates, variable values, etc. This program takes a casefile as input. Thus, it will check the format of the casefile, and all associated geometry and variable files referenced in the casefile. See How To Use ens_checker.

Supported EnSight Elements

The elements that are supported by the EnSight6 format are:

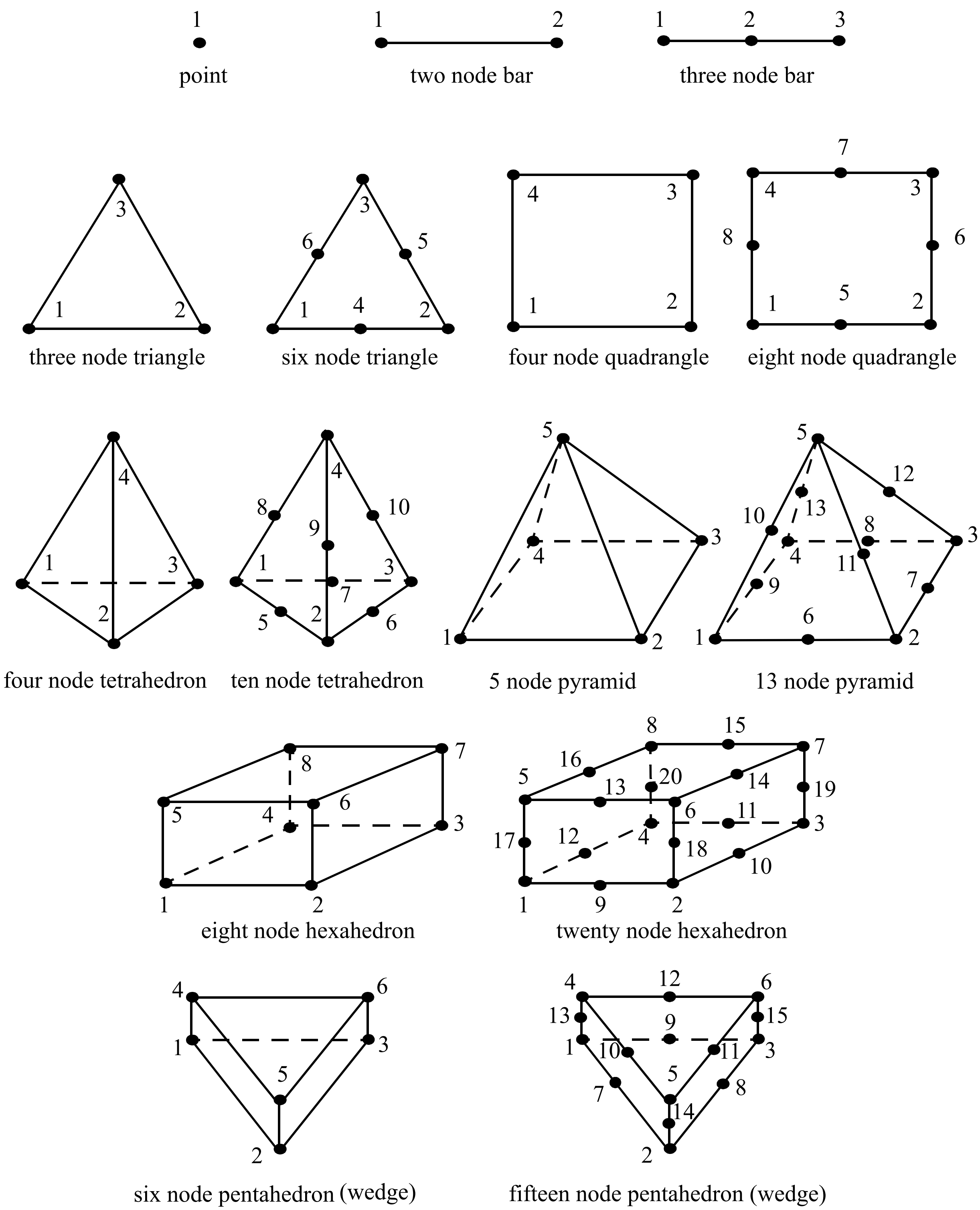


Figure 11-4
Supported EnSight6 Elements

EnSight6 Case File Format

The Case file is an ASCII free format file that contains all the file and name information for accessing model (and measured) geometry, variable, and time information. It is comprised of five sections (`FORMAT, GEOMETRY, VARIABLE, TIME, FILE`) as described below:

Notes: *All lines in the Case file are limited to 79 characters.*

The titles of each section must be in all capital letters.

Anything preceded by a "#" denotes a comment and is ignored. Comments may append information lines or be placed on their own lines.

Information following ":" may be separated by white spaces or tabs.

Specifications encased in "[]" are optional, as indicated.

Format Section

This is a required section which specifies the type of data to be read.

Usage:

```
FORMAT
type:    ensight
```

Geometry Section

This is a required section which specifies the geometry information for the model (as well as measured geometry if present, and periodic match file (see Section 11.9, Periodic Matchfile Format) if present).

Usage:

```
GEOMETRY
model:      [ts]  [fs]        filename    [change_coords_only]
measured:   [ts]  [fs]        filename    [change_coords_only]
match:                        filename
boundary:                     filename
```

where: `ts` = time set number as specified in TIME section. This is optional.
`fs` = corresponding file set number as specified in FILE section below.
`filename` = The filename of the appropriate file.
-> Model or measured filenames for a static geometry case, as well as match and boundary filenames will not contain "*" wildcards.
-> Model or measured filenames for a changing geometry case will contain "*" wildcards.
`change_coords_only` = The option to indicate that the changing geometry (as indicated by wildcards in the filename) is coords only. Otherwise, changing geometry connectivity will be assumed.

Variable Section

This is an optional section which specifies the files and names of the variables. Constant variable values can also be set in this section.

Usage:

```
VARIABLE
constant per case:            [ts]        description  const_value(s)
scalar per node:              [ts]  [fs]  description  filename
vector per node:              [ts]  [fs]  description  filename
tensor symm per node:         [ts]  [fs]  description  filename
scalar per element:           [ts]  [fs]  description  filename
vector per element:           [ts]  [fs]  description  filename
tensor symm per element:      [ts]  [fs]  description  filename
scalar per measured node:     [ts]  [fs]  description  filename
vector per measured node:     [ts]  [fs]  description  filename
complex scalar per node:      [ts]  [fs]  description  Re_fn    Im_fn    freq
complex vector per node:      [ts]  [fs]  description  Re_fn    Im_fn    freq
```

```
complex scalar per element:     [ts] [fs] description Re_fn    Im_fn    freq
complex vector per element:     [ts] [fs] description Re_fn    Im_fn    freq
```

where:

`ts` = The corresponding time set number (or index) as specified in TIME section below. This is only required for transient constants and variables.

`fs` = The corresponding file set number (or index) as specified in FILE section below.

`description` = The variable (GUI) name (ex. Pressure, Velocity, etc.)

`const_value(s)` = The constant value. If constants change over time, then `ns` (see TIME section below) constant values of `ts`.

`filename` = The filename of the variable file. Note: only transient filenames contain "*" wildcards.

`Re_fn` = The filename for the file containing the real values of the complex variable.

`Im_fn` = The filename for the file containing the imaginary values of the complex variable.

`freq` = The corresponding harmonic frequency of the complex variable. For complex variables where harmonic frequency is undefined, simply use the text string: UNDEFINED.

Note: *As many variable description lines as needed may be used.*

Note: *Variable descriptions have the following restrictions:*

The maximum variable name length is documented at the beginning of this chapter.

Duplicate variable descriptions are not allowed.

Leading and trailing white space will be eliminated.

Variable descriptions must not start with a numeric digit.

Variable descriptions must not contain any of the following reserved characters:

(	[	+	@	!	*	$
)	]	-	space	#	^	/

Time Section

This is an optional section for steady state cases, but is required for transient cases. It contains time set information. Shown below is information for one time set. Multiple time sets (up to 16) may be specified for measured data as shown in Case File Example 3 below.

Usage:

```
TIME
time set:              ts  [description]
number of steps:       ns
filename start number: fs
filename increment:    fi
time values:           time_1  time_2  .... time_ns
```

or

```
TIME
time set:              ts  [description]
number of steps:       ns
filename numbers:      fn
time values:           time_1  time_2  .... time_ns
```

where: `ts` = timeset number. This is the number referenced in the GEOMETRY

and VARIABLE sections.
`description` = optional timeset description which will be shown in user interface.
`ns` = number of transient steps
`fs` = the number to replace the "*" wildcards in the filenames, for the first step
`fi` = the increment to fs for subsequent steps
`time` = the actual time values for each step, each of which must be separated by a white space and which may continue on the next line if needed
`fn` = a list of numbers or indices, to replace the "*" wildcards in the filenames.

File Section

This section is optional for expressing a transient case with single-file formats. This section contains single-file set information. This information specifies the number of time steps in each file of each *data entity*, i.e. each geometry and each variable (model and/or measured). Each data entity's corresponding file set might have multiple *continuation* files due to system file size limit, i.e. ~2 GB for 32-bit and ~4 TB for 64-bit architectures. Each file set corresponds to one and only one time set, but a time set may be referenced by many file sets. The following information may be specified in each file set. For file sets where all of the time set data exceeds the maximum file size limit of the system, both `filename index` and `number of steps` are repeated within the file set definition for each continuation file required. Otherwise `filename index` may be omitted if there is only one file. File set information is shown in Case File Example 4 below.

Usage:

```
FILE
file set:            fs
filename index:      fi  # Note: only used when data continues in other files
number of steps:     ns
```

where: `fs` = file set number. This is the number referenced in the GEOMETRY and VARIABLE sections above.
`ns` = number of transient steps
`fi` = file index number in the file name (replaces "*" in the filenames)

Case File Example 1

The following is a minimal EnSight6 case file for a steady state model with some results.

Note: this (`en6.case`*) file, as well as all of its referenced geometry and variable files (along with a couple of command files) can be found under your installation directory (path:* `$CEI_HOME/ensight92/data/user_manual`*). The EnSight6 Geometry File Example and the Variable File Examples are the contents of these files.*

```
FORMAT
type:  ensight

GEOMETRY
model:  en6.geo

VARIABLE
constant per case:            Cden   .8

scalar per element:           Esca   en6.Esca
scalar per node:              Nsca   en6.Nsca

vector per element:           Evec   en6.Evec
vector per node:              Nvec   en6.Nvec

tensor symm per element:      Eten   en6.Eten
tensor symm per node:         Nten   en6.Nten

complex scalar per element:   Ecmp   en6.Ecmp_r  en6.Ecmp_i   2.
complex scalar per node:      Ncmp   en6.Ncmp_r  en6.Ncmp_i   4.
```

Case File Example 2 The following is a Case file for a transient model. The connectivity of the geometry is also changing.

```
FORMAT
type:  ensight

GEOMETRY
model:                  1                       example2.geo**

VARIABLE
scalar per node:        1     Stress            example2.scl**
vector per node:        1     Displacement      example2.dis**

TIME
time set:                     1
number of steps:              3
filename start number:        0
filename increment:           1
time values:       1.0   2.0   3.0
```

The following files would be needed for Example 2:

```
example2.geo00        example2.scl00        example2.dis00
example2.geo01        example2.scl01        example2.dis01
example2.geo02        example2.scl02        example2.dis02
```

Case File Example 3 The following is a Case file for a transient model with measured data.

This example has pressure given per element.

```
FORMAT
type:  ensight


GEOMETRY
model:                  1                 example3.geo*
measured:               2                 example3.mgeo**


VARIABLE
constant per case:                        Gamma       1.4
constant per case:                  1     Density     .9 .9 .7 .6 .6
scalar per element                  1     Pressure          example3.pre*
vector per node:                    1     Velocity          example3.vel*
scalar per measured node:           2     Temperature       example3.mtem**
vector per measured node:           2     Velocity          example3.mvel**

TIME
time set:                     1
number of steps:              5
filename start number:        1
filename increment:           2
time values:                  .1 .2 .3          # This example shows that time
                              .4 .5             # values can be on multiple lines
time set:                     2
number of steps:              6
filename start number:        0
filename increment:           2
time values:
.05 .15 .25 .34 .45 .55
```

The following files would be needed for Example 3:

```
example3.geo1        example3.pre1        example3.vel1
example3.geo3        example3.pre3        example3.vel3
example3.geo5        example3.pre5        example3.vel5
example3.geo7        example3.pre7        example3.vel7
example3.geo9        example3.pre9        example3.vel9

example3.mgeo00      example3.mtem00      example3.mvel00
example3.mgeo02      example3.mtem02      example3.mvel02
example3.mgeo04      example3.mtem04      example3.mvel04
example3.mgeo06      example3.mtem06      example3.mvel06
example3.mgeo08      example3.mtem08      example3.mvel08
example3.mgeo10      example3.mtem10      example3.mvel10
```

Case File Example 4

The following is Case File Example 3 expressed in transient single-file formats.

In this example, the transient data for the measured velocity data entity happens to be greater than the maximum file size limit. Therefore, the first four time steps fit and are contained in the first file, and the last two time steps are 'continued' in a second file.

```
FORMAT
type: ensight

GEOMETRY
model:                  1     example4.geo
measured:               2     example4.mgeo

VARIABLE
constant per case:                        Density          .5
scalar per element:           1     1     Pressure         example4.pre
vector per node:              1     1     Velocity         example4.vel
scalar per measured node:     2     2     Temperature      example4.mtem
vector per measured node:     2     3     Velocity         example4.mvel*

TIME
time set:               1     Model
number of steps:        5
time values:            .1 .2 .3 .4 .5

time set:               2     Measured
number of steps:        6
time values:            .05 .15 .25 .34 .45 .55

FILE
file set:               1
number of steps:        5

file set:               2
number of steps:        6

file set:               3
filename index:         1
number of steps:        4
filename index:         2
number of steps:        2
```

The following files would be needed for Example 4:

```
example4.geo     example4.pre     example4.vel
example4.mgeo    example4.mtem    example4.mvel1
                                  example4.mvel2
```

Contents of Transient Single Files

Each file contains transient data that corresponds to the specified number of time steps. The data for each time step sequentially corresponds to the simulation time values `(time values)` found listed in the `TIME` section. In transient single-file format, the data for each time step essentially corresponds to a standard EnSight6 geometry or variable file (model or measured) as expressed in multiple file format. The data for each time step is enclosed between two *wrapper* records, i.e. preceded by a `BEGIN TIME STEP` record and followed by an `END TIME STEP` record. Time step data is not split between files. If there is not enough room to append the data from a time step to the file without exceeding the maximum file limit of a particular system, then a continuation file must be created for the time step data and any subsequent time step. Any type of user comments may be included before and/or after each transient step wrapper.

Note 1: If transient single file format is used, EnSight expects all files of a dataset to be specified in transient single file format. Thus, even static files must be enclosed between a `BEGIN TIME STEP` *and an* `END TIME STEP` *wrapper.*

Note 2: For binary geometry files, the first `BEGIN TIME STEP` *wrapper must follow the <C Binary/Fortran Binary> line. Both* `BEGIN TIME STEP` *and* `END TIME STEP` *wrappers are written according to type (1) in binary.* *(see Writing EnSight6 Binary Files, in Section 11.2)*

*Note 3: Efficient reading of each file (especially binary) is facilitated by appending each file with a **file index**. A file index contains appropriate information to access the file byte positions of each time step in the file. (EnSight automatically appends a file index to each file when exporting in transient single file format.) If used, the file index must follow the last* `END TIME STEP` *wrapper in each file.*

File Index Usage:

ASCII	Binary	Item	Description
“%20d\n”	sizeof(int)	n	Total number of data time steps in the file.
“%20d\n”	sizeof(long)	fb_1	File byte loc for contents of 1st time step*
“%20d\n”	sizeof(long)	fb_2	File byte loc for contents of 2nd time step*
. . .	. . .	. . .	. . .
“%20d\n”	sizeof(long)	fb_n	File byte loc for contents of nth time step*
“%20d\n”	sizeof(int)	flag	Miscellaneous flag (= 0 for now)
“%20d\n”	sizeof(long)	fb of item n	File byte loc for Item n above
“%s\n”	sizeof(char)*80	“FILE_INDEX”	File index keyword

* *Each file byte location is the first byte that follows the* `BEGIN TIME STEP` *record.*

Shown below are the contents of each of the above files, using the data files from Case File Example 3 for reference (without FILE_INDEX for simplicity).

Contents of file `example4.geo_1`:

BEGIN TIME STEP
Contents of file `example3.geo1`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.geo3`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.geo5`

END TIME STEP
BEGIN TIME STEP
Contents of file `example3.geo7`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.geo9`
END TIME STEP

Contents of file `example4.pre_1`:

BEGIN TIME STEP
Contents of file `example3.pre1`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.pre3`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.pre5`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.pre7`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.pre9`
END TIME STEP

Contents of file `example4.vel_1`:

BEGIN TIME STEP
Contents of file `example3.vel1`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.vel3`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.vel5`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.vel7`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.vel9`
END TIME STEP

Contents of file `example4.mgeo_1`:

BEGIN TIME STEP
Contents of file `example3.mgeo00`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mgeo02`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mgeo04`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mgeo06`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mgeo08`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mgeo10`
END TIME STEP

Contents of file `example4.mtem_1:`

BEGIN TIME STEP
Contents of file `example3.mtem00`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mtem02`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mtem04`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mtem06`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mtem08`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mtem10`
END TIME STEP

Contents of file `example4.mvel1_1:`

BEGIN TIME STEP
Contents of file `example3.mvel00`
END TIME STEP

BEGIN TIME STEP
Contents of file `example3.mvel02`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mvel04`
END TIME STEP
BEGIN TIME STEP
Contents of file `example3.mvel06`
END TIME STEP

Contents of file `example4.mvel2_1:`

Comments can precede the beginning wrapper here.

BEGIN TIME STEP
Contents of file `example3.mvel08`
END TIME STEP

Comments can go between time step wrappers here.

BEGIN TIME STEP
Contents of file `example3.mvel10`
END TIME STEP

Comments can follow the ending time step wrapper.

EnSight6 Wild Card Name Specification

For transient data, if multiple time files are involved, the file names must conform to the EnSight wild-card specification. This specification is as follows:

- File names must include numbers that are in ascending order from beginning to end.
- Numbers in the files names must be zero filled if there is more than one significant digit.
- Numbers can be anywhere in the file name.
- When the file name is specified in the EnSight result file, you must replace the numbers in the file with an asterisk(*). The number of asterisks specified is the number of significant digits. The asterisk must occupy the same place as the numbers in the file names.

EnSight6 Geometry File Format

The EnSight6 format consists of keywords followed by information. The following seven items are important when working with EnSight6 geometry files:

1. You do not have to assign node IDs. If you do, the element connectivities are based on the node numbers. If you let EnSight assign the node IDs, the nodes are considered to be sequential starting at node 1, and element connectivity is done accordingly. If node IDs are set to off, they are numbered internally; however, you will not be able to display or query on them. If you have node IDs in your data, you can have EnSight ignore them by specifying “node id ignore.” Using this option may reduce some of the memory taken up by the Client and Server, but display and query on the nodes will not be available.
2. You do not need to specify element IDs. If you specify element IDs, or you let EnSight assign them, you can show them on the screen. If they are set to off, you will not be able to show or query on them. If you have element IDs in your data you can have EnSight ignore them by specifying “element id ignore.” Using this option will reduce some of the memory taken up by the Client and Server. This may or may not be a significant amount, and remember that display and query on the elements will not be available.
3. The format of integers and real numbers **must be followed** (See the Geometry Example below).
4. Integers are written out using the following integer format:

From C:	`8d` format
From FORTRAN:	`i8` format

   Real numbers are written out using the following floating-point format:

From C:	`12.5e` format
From FORTRAN:	`e12.5` format

   The number of integers or reals per line must also be followed!
5. By default, a Part is processed to show the outside boundaries. This representation is loaded to the Client host system when the geometry file is read (unless other attributes have been set on the workstation, such as feature angle).
6. Coordinates for unstructured data must be defined before any Parts can be defined. The different elements can be defined in any order (that is, you can define a hexa8 before a bar2).
7. A Part containing structured data cannot contain any unstructured element types or more than one block. **Each structured Part is limited to a single block.** A structured block is indicated by following the Part description line with either the “block” line or the “block iblanked” line. An “iblanked” block must contain an additional integer array of values at each node, traditionally called the iblank array. Valid iblank values for the EnSight format are:

0	for nodes which are exterior to the model, sometimes called blanked-out nodes
1	for nodes which are interior to the model, thus in the free stream and to be used
<0 or >1	for any kind of boundary nodes

In EnSight’s structured Part building dialog, the iblank option selected will control which portion of the structured block is “created”. Thus, from the same structured block, the interior flow field part as well as a symmetry boundary part could be “created”.

Note: By default EnSight does not do any “partial” cell iblank processing. Namely, only complete cells containing no “exterior” nodes are created. It is possible to obtain partial cell processing by issuing the “test:partial_cells_on” command in the Command Dialog before reading the file.

Note also that for the structured format, the standard order of nodes is such that I’s advance quickest, followed by J’s, and then K’s.

8. Maximum number of parts is 65000
   Maximum number of variables is 10000
   Maximum file name length is 1024
   Maximum part name length is 79, but the GUI will only display 49
   Maximum variable name length is 79, but the GUI will only display 49

Generic Format

Not all of the lines included in the following generic example file are necessary:

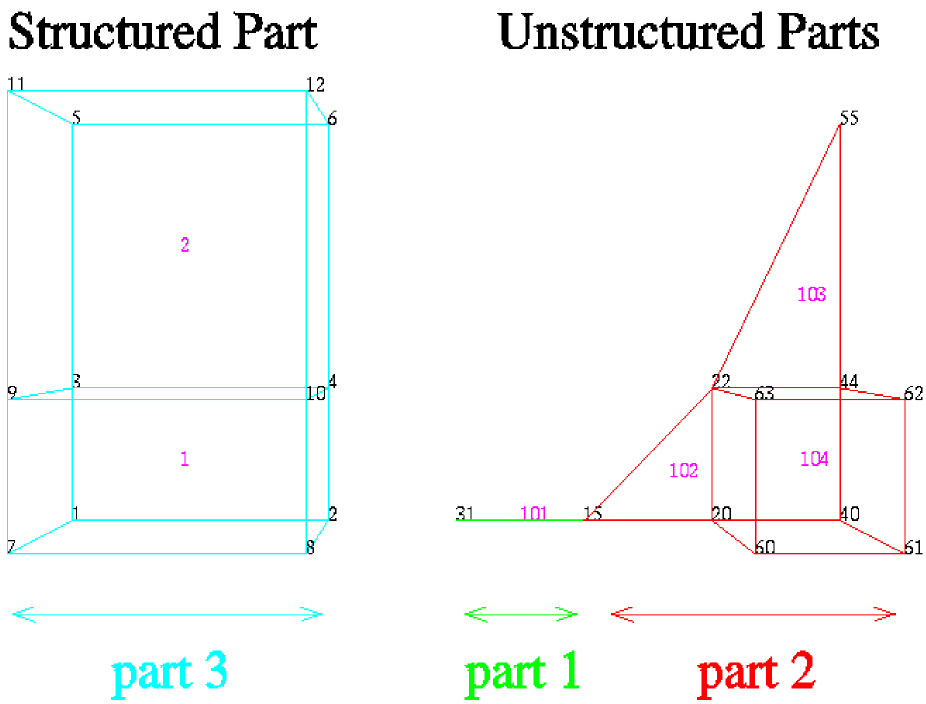
```
description line 1                           |
description line 2                           |
node id <off/given/assign/ignore>            |  All geometry files must
element id <off/given/assign/ignore>         |  contain these first six lines
coordinates                                  |
# of unstructured nodes                      |
id x y z
id x y z
id x y z
.
.
.
part #
description line
point
number of points
id nd
id nd
id nd
.
.
.
bar2
number of bar2's
id nd nd
id nd nd
id nd nd
.
.
.
bar3
number of bar3's
id nd nd nd
id nd nd nd
id nd nd nd
.
.
.
tria3
number of three node triangles
```

```
id nd nd nd
id nd nd nd
id nd nd nd
.
.
.
tria6
number of six node triangles
id nd nd nd nd nd nd
.
.
.
quad4
number of quad 4’s
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
.
.
.
quad8
number of quad 8’s
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
.
.
.
tetra4
number of 4 node tetrahedrons
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
.
.
.
tetra10
number of 10 node tetrahedrons
id nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd
.
.
.
pyramid5
number of 5 node pyramids
id nd nd nd nd nd
id nd nd nd nd nd
id nd nd nd nd nd
id nd nd nd nd nd
.
.
.
pyramid13
number of 13 node pyramids
id nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd
```

```
id nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd
.
.
.
hexa8
number of 8 node hexahedrons
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
.
.
.
hexa20
number of 20 node hexahedrons
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
.
.
.
penta6
number of 6 node pentahedrons
id nd nd nd nd nd nd
id nd nd nd nd nd nd
id nd nd nd nd nd nd
id nd nd nd nd nd nd
id nd nd nd nd nd nd
.
.
.
penta15
number of 15 node pentahedrons
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
.
.
.

part #
description line
block                               #nn=i*j*k
       i       j       k
x_n1 x_n2 x_n3 ..... x_nn           (6/line)
y_n1 y_n2 y_n3 ..... y_nn            "
z_n1 z_n2 z_n3 ..... z_nn            "

part #
description line
block     iblanked                  #nn=i*j*k
       i       j       k
x_n1 x_n2 x_n3 ..... x_nn           (6/line)
y_n1 y_n2 y_n3 ..... y_nn            "
z_n1 z_n2 z_n3 ..... z_nn            "
ib_n1 ib_n2 ib_n3 .... ib_nn        (10/line)
```



EnSight6 Geometry File Example

The following is an example of an ASCII EnSight6 geometry file with 11 defined unstructured nodes from which 2 unstructured parts are defined, and a 2x3x2 structured part as depicted in the above diagram. (See Case File Example 1 for reference to this file.)

```
This is the 1st description line of the EnSight6 geometry example
This is the 2nd description line of the EnSight6 geometry example
node id given
element id given
coordinates
      11
      15 4.00000e+00 0.00000e+00 0.00000e+00
      31 3.00000e+00 0.00000e+00 0.00000e+00
      20 5.00000e+00 0.00000e+00 0.00000e+00
      40 6.00000e+00 0.00000e+00 0.00000e+00
      22 5.00000e+00 1.00000e+00 0.00000e+00
      44 6.00000e+00 1.00000e+00 0.00000e+00
      55 6.00000e+00 3.00000e+00 0.00000e+00
      60 5.00000e+00 0.00000e+00 2.00000e+00
      61 6.00000e+00 0.00000e+00 2.00000e+00
      62 6.00000e+00 1.00000e+00 2.00000e+00
      63 5.00000e+00 1.00000e+00 2.00000e+00
part 1
2D uns-elements  (description line for part 1)
tria3
       2
     102      15      20      22
     103      22      44      55
hexa8
       1
     104      20      40      44      22      60      61      62      63
part 2
1D uns-elements  (description line for part 2)
bar2
       1
     101     31      15
part 3
```

```
3D struct-part  (description line for part 3)
block iblanked
       2       3       2
 0.00000e+00 2.00000e+00 0.00000e+00 2.00000e+00 0.00000e+00 2.00000e+00
 0.00000e+00 2.00000e+00 0.00000e+00 2.00000e+00 0.00000e+00 2.00000e+00
 0.00000e+00 0.00000e+00 1.00000e+00 1.00000e+00 3.00000e+00 3.00000e+00
 0.00000e+00 0.00000e+00 1.00000e+00 1.00000e+00 3.00000e+00 3.00000e+00
 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
 2.00000e+00 2.00000e+00 2.00000e+00 2.00000e+00 2.00000e+00 2.00000e+00
       1       1       1      1       1       1       1       1       1       1
       1       1
```

EnSight6 Variable File Format

EnSight6 variable files can either be per_node or per_element. They cannot be both. However, an EnSight model can have some variables which are per_node and other variables which are per_element.

EnSight6 Per_Node Variable File Format

EnSight6 variable files for per_node variables contain any values for each unstructured node followed by any values for each structured node.

First comes a single description line.

Second comes any unstructured node value. The number of values per node depends on the type of field. An unstructured scalar field has one, a vector field has three (order: x,y,z), a 2nd order symmetric tensor field has 6 (order: 11, 22, 33, 12, 13, 23), and a 2nd order asymmetric tensor field has 9 values per node (order: 11, 12, 13, 21, 22, 23, 31, 32, 33). An unstructured complex variable in EnSight6 consists of two scalar or vector fields (one real and one imaginary), with scalar and vector values written to their separate files respectively.

Third comes any structured data information, starting with a part # line, followed by a line containing the “block”, and then lines containing the values for each structured node which are output in the same IJK component order as the coordinates. Briefly, a structured scalar is the same as an unstructured scalar, one value per node. A structured vector is written one value per node per component, thus three sequential scalar field blocks. Likewise for a structured 2nd order symmetric tensor, written as six sequential scalar field blocks, and a 2nd order tensor, written as nine sequential scalar field blocks. The same methodology applies for a complex variable only with the real and imaginary fields written to separate structured scalar or vector files.

The values **must be written** in the following floating point format (**6 per line** as shown in the examples below):

From C: `12.5e` format

From FORTRAN: `e12.5` format

The format of a per_node variable file is as follows:

- Line 1

  This line is a description line.

- Line 2 through the end of the file contains the values at each node in the model.

A generic example for *per_node* variables:

```
One description line for the entire file
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+**
part #
block
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+**
```

Note that there is a format difference between the unstructured and structured (block) portions of the vector and tensor data. For example a multiple component unstructured vector appears as x y z triplets, while the structured counterpart lists all x then all y and finally all z.

The following variable file examples reflect scalar, vector, tensor, and complex variable values *per node* for the previously defined EnSight6 Geometry File Example with 11 defined unstructured nodes and a 2x3x2 structured Part (Part number 3). The values are summarized in the following table.

						ComplexScalar	
	Node Index	Node Id	Scalar Value	Vector Values	Tensor (2nd order symm.) Values	Real Value	Imaginary Value
Unstructured							
	1	15	(1.)	(1.1, 1.2, 1.3)	(1.1, 1.2, 1.3, 1.4, 1.5, 1.6)	(1.1)	(1.2)
	2	31	(2.)	(2.1, 2.2, 2.3)	(2.1, 2.2, 2.3, 2.4, 2.5, 2.6)	(2.1)	(2.2)
	3	20	(3.)	(3.1, 3.2, 3.3)	(3.1, 3.2, 3.3, 3.4, 3.5, 3.6)	(3.1)	(3.2)
	4	40	(4.)	(4.1, 4.2, 4.3)	(4.1, 4.2, 4.3, 4.4, 4.5, 4.6)	(4.1)	(4.2)
	5	22	(5.)	(5.1, 5.2, 5.3)	(5.1, 5.2, 5.3, 5.4, 5.5, 5.6)	(5.1)	(5.2)
	6	44	(6.)	(6.1, 6.2, 6.3)	(6.1, 6.2, 6.3, 6.4, 6.5, 6.6)	(6.1)	(6.2)
	7	55	(7.)	(7.1, 7.2, 7.3)	(7.1, 7.2, 7.3, 7.4, 7.5, 7.6)	(7.1)	(7.2)
	8	60	(8.)	(8.1, 8.2, 8.3)	(8.1, 8.2, 8.3, 8.4, 8.5, 8.60	(8.1)	(8.2)
	9	61	(9.)	(9.1, 9.2, 9.3)	(9.1, 9.2, 9.3, 9.4, 9.5, 9.6)	(9.1)	(9.2)
	10	62	(10.)	(10.1,10.2,10.3)	(10.1,10.2,10.3,10.4,10.5,10.6)	(10.1)	(10.2)
	11	63	(11.)	(11.1,11.2,11.3)	(11.1,11.2,11.3,11.4,11.5,11.6)	(11.1)	(11.2)
Structured							
	1	1	(1.)	(1.1, 1.2, 1.3)	(1.1, 1.2, 1.3, 1.4, 1.5, 1.6)	(1.1)	(1.2)
	2	2	(2.)	(2.1, 2.2, 2.3)	(2.1, 2.2, 2.3, 2.4, 2.5, 2.6)	(2.1)	(2.2)
	3	3	(3.)	(3.1, 3.2, 3.3)	(3.1, 3.2, 3.3, 3.4, 3.5, 3.6)	(3.1)	(3.2)
	4	4	(4.)	(4.1, 4.2, 4.3)	(4.1, 4.2, 4.3, 4.4, 4.5, 4.6)	(4.1)	(4.2)
	5	5	(5.)	(5.1, 5.2, 5.3)	(5.1, 5.2, 5.3, 5.4, 5.5, 5.6)	(5.1)	(5.2)
	6	6	(6.)	(6.1, 6.2, 6.3)	(6.1, 6.2, 6.3, 6.4, 6.5, 6.6)	(6.1)	(6.2)
	7	7	(7.)	(7.1, 7.2, 7.3)	(7.1, 7.2, 7.3, 7.4, 7.5, 7.6)	(7.1)	(7.2)
	8	8	(8.)	(8.1, 8.2, 8.3)	(8.1, 8.2, 8.3, 8.4, 8.5, 8.6)	(8.1)	(8.2)
	9	9	(9.)	(9.1, 9.2, 9.3)	(9.1, 9.2, 9.3, 9.4, 9.5, 9.6)	(9.1)	(9.2)
	10	10	(10.)	(10.1,10.2,10.3)	(10.1,10.2,10.3,10.4,10.5,10.6)	(10.1)	(10.2)
	11	11	(11.)	(11.1,11.2,11.3)	(11.1,11.2,11.3,11.4,11.5,11.6)	(11.1)	(11.2)

Per_node (Scalar) Variable Example 1 This example shows ASCII scalar file (`en6.Nsca`) for the geometry example.

```
Per_node scalar values for the EnSight6 geometry example
 1.00000E+00 2.00000E+00 3.00000E+00 4.00000E+00 5.00000E+00 6.00000E+00
 7.00000E+00 8.00000E+00 9.00000E+00 1.00000E+01 1.10000E+01
part 3
block
 1.00000E+00 2.00000E+00 3.00000E+00 4.00000E+00 5.00000E+00 6.00000E+00
 7.00000E+00 8.00000E+00 9.00000E+00 1.00000E+01 1.10000E+01
```

Per_node (Vector) Variable Example 2 This example shows ASCII vector file (`en6.Nvec`) for the geometry example.

```
Per_node vector values for the EnSight6 geometry example
 1.10000E+00 1.20000E+00 1.30000E+00 2.10000E+00 2.20000E+00 2.30000E+00
 3.10000E+00 3.20000E+00 3.30000E+00 4.10000E+00 4.20000E+00 4.30000E+00
 5.10000E+00 5.20000E+00 5.30000E+00 6.10000E+00 6.20000E+00 6.30000E+00
 7.10000E+00 7.20000E+00 7.30000E+00 8.10000E+00 8.20000E+00 8.30000E+00
 9.10000E+00 9.20000E+00 9.30000E+00 1.01000E+01 1.02000E+01 1.03000E+01
 1.11000E+01 1.12000E+01 1.13000E+01
part 3
block
 1.10000E+00 2.10000E+00 3.10000E+00 4.10000E+00 5.10000E+00 6.10000E+00
 7.10000E+00 8.10000E+00 9.10000E+00 1.01000E_01 1.11000E+01
 1.20000E+00 2.20000E+00 3.20000E+00 4.20000E+00 5.20000E+00 6.20000E+00
 7.20000E+00 8.20000E+00 9.20000E+00 1.02000E+01 1.12000E+01
 1.30000E+00 2.30000E+00 3.30000E+00 4.30000E+00 5.30000E+00 6.30000E+00
 7.30000E+00 8.30000E+00 9.30000E+00 1.03000E+01 1.13000E+01
```

Per_node (Tensor) Variable Example 3 This example shows an ASCII 2nd order symmetric tensor file (`en6.Nten`) for the geometry example.

```
Per_node symmetric tensor values for the EnSight6 geometry example
 1.10000E+00 1.20000E+00 1.30000E+00 1.40000E+00 1.50000E+00 1.60000E+00
 2.10000E+00 2.20000E+00 2.30000E+00 2.40000E+00 2.50000E+00 2.60000E+00
 3.10000E+00 3.20000E+00 3.30000E+00 3.40000E+00 3.50000E+00 3.60000E+00
 4.10000E+00 4.20000E+00 4.30000E+00 4.40000E+00 4.50000E+00 4.60000E+00
 5.10000E+00 5.20000E+00 5.30000E+00 5.40000E+00 5.50000E+00 5.60000E+00
 6.10000E+00 6.20000E+00 6.30000E+00 6.40000E+00 6.50000E+00 6.60000E+00
 7.10000E+00 7.20000E+00 7.30000E+00 7.40000E+00 7.50000E+00 7.60000E+00
 8.10000E+00 8.20000E+00 8.30000E+00 8.40000E+00 8.50000E+00 8.60000E+00
 9.10000E+00 9.20000E+00 9.30000E+00 9.40000E+00 9.50000E+00 9.60000E+00
 1.01000E+01 1.02000E+01 1.03000E+01 1.04000E+01 1.05000E+01 1.06000E+01
 1.11000E+01 1.12000E+01 1.13000E+01 1.14000E+01 1.15000E+01 1.16000E+01
part 3
block
 1.10000E+00 2.10000E+00 3.10000E+00 4.10000E+00 5.10000E+00 6.10000E+00
 7.10000E+00 8.10000E+00 9.10000E+00 1.01000E+01 1.11000E+01
 1.20000E+00 2.20000E+00 3.20000E+00 4.20000E+00 5.20000E+00 6.20000E+00
 7.20000E+00 8.20000E+00 9.20000E+00 1.02000E+01 1.12000E+01
 1.30000E+00 2.30000E+00 3.30000E+00 4.30000E+00 5.30000E+00 6.30000E+00
 7.30000E+00 8.30000E+00 9.30000E+00 1.03000E+01 1.13000E+01
 1.40000E+00 2.40000E+00 3.40000E+00 4.40000E+00 5.40000E+00 6.40000E+00
 7.40000E+00 8.40000E+00 9.40000E+00 1.04000E+01 1.14000E+01
 1.50000E+00 2.50000E+00 3.50000E+00 4.50000E+00 5.50000E+00 6.50000E+00
 7.50000E+00 8.50000E+00 9.50000E+00 1.05000E+01 1.15000E+01
 1.60000E+00 2.60000E+00 3.60000E+00 4.60000E+00 5.60000E+00 6.60000E+00
 7.60000E+00 8.60000E+00 9.60000E+00 1.06000E+01 1.16000E+01
```

Per_node (Complex) Variable Example 4 This example shows the ASCII complex real (`en6.Ncmp_r`) and imaginary (`en6.Ncmp_i`) *scalar* files for the geometry example. (The same methodology would apply for complex real and imaginary *vector* files.)

Real scalar File:

```
Per_node complex real scalar values for the EnSight6 geometry example
 1.10000E+00 2.10000E+00 3.10000E+00 4.10000E+00 5.10000E+00 6.10000E+00
 7.10000E+00 8.10000E+00 9.10000E+00 1.01000E+01 1.11000E+01
 part 3
 block
 1.10000E+00 2.10000E+00 3.10000E+00 4.10000E+00 5.10000E+00 6.10000E+00
 7.10000E+00 8.10000E+00 9.10000E+00 1.01000E+01 1.11000E+01 1.21000E+00
```

Imaginary scalar File:

```
Per_node complex imaginary scalar values for the EnSight6 geometry example
 1.20000E+00 2.20000E+00 3.20000E+00 4.20000E+00 5.20000E+00 6.20000E+00
 7.20000E+00 8.20000E+00 9.20000E+00 1.02000E+01 1.12000E+01
 part 3
 block
 1.20000E+00 2.20000E+00 3.20000E+00 4.20000E+00 5.20000E+00 6.20000E+00
 7.20000E+00 8.20000E+00 9.20000E+00 1.02000E+01 1.12000E+01 1.22000E+00
```

EnSight6 Per_Element Variable File Format

EnSight variable files for per_element variables contain values for each element of designated types of designated Parts. First comes a single description line. Second comes a Part line. Third comes an element type line and fourth comes the value for each element of that type and part. If it is a scalar variable, there is one value per element, while for vector variables there are three values per element. (The number of elements of the given type are obtained from the corresponding EnSight6 geometry file.)

The values must be written in the following floating point format (6 per line as shown in the examples below):

From C: `12.5e` format

From FORTRAN: `e12.5` format

The format of a per_element variable file is as follows:

- Line 1 This line is a description line.
- Line 2 Part line, with part number corresponding to the geometry file.
- Line 3 Element type line (example: tria3, hexa8, ...)
- Line 4 Repeats until next element type line, part line, or end of file is reached. Lists values for each element of this part and type.

A generic example for *per_element* variables:

```
One description line for the entire file
part  #
element type
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+**
part  #
block
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+**
```

Note that there is a format difference between the unstructured and structured (block) portions of the vector and tensor data. For example a multiple component unstructured vector appears as x y z triplets, while the structured counterpart lists all x then all y and finally all z.

The following variable file examples reflect scalar, vector, tensor, and complex variable values *per element* for the previously defined EnSight6 Geometry File Example with 11 defined unstructured nodes and a 2x3x2 structured Part (Part number 3). The values are summarized in the following table.

	Element Index	Element Id	Scalar Value	Vector Values	Tensor (2nd order symm.) Values	Complex Scalar Real Value	Complex Scalar Imaginary Value
Unstructured							
bar2							
	1	101	(1.)	(1.1, 1.2, 1.3)	(1.1, 1.2, 1.3, 1.4, 1.5, 1.6)	(1.1)	(1.2)
tria3							
	1	102	(2.)	(2.1, 2.2, 2.3)	(2.1, 2.2, 2.3, 2.4, 2.5, 2.6)	(2.1)	(2.2)
	2	103	(3.)	(3.1, 3.2, 3.3)	(3.1, 3.2, 3.3, 3.4, 3.5, 3.6)	(3.1)	(3.2)
hexa8							
	1	104	(4.)	(4.1, 4.2, 4.3)	(4.1, 4.2, 4.3, 4.4, 4.5, 4.6)	(4.1)	(4.2)
Structured							
block	1	1	(5.)	(5.1, 5.2, 5.3)	(5.1, 5.2, 5.3, 5.4, 5.5, 5.6)	(5.1)	(5.2)
	2	2	(6.)	(6.1, 6.2, 6.3)	(6.1, 6.2, 6.3, 6.4, 6.5, 6.6)	(6.1)	(6.2)
	3	3	(7.)	(7.1, 7.2, 7.3)	(7.1, 7.2, 7.3, 7.4, 7.5, 7.6)	(7.1)	(7.2)
	4	4	(8.)	(8.1, 8.2, 8.3)	(8.1, 8.2, 8.3, 8.4, 8.5, 8.6)	(8.1)	(8.2)
	5	5	(9.)	(9.1, 9.2, 9.3)	(9.1, 9.2, 9.3, 9.4, 9.5, 9.6)	(9.1)	(9.2)
	6	6	(10.)	(10.1, 10.2, 10.3)	(10.1, 10.2, 10.3, 10.4, 10.5, 10.6)	(10.1)	(10.2)
	7	7	(11.)	(11.1, 11.2, 11.3)	(11.1, 11.2, 11.3, 11.4, 11.5, 11.6)	(11.1)	(11.2)
	8	8	(12.)	(12.1, 12.2, 12.3)	(12.1, 12.2, 12.3, 12.4, 12.5, 12.6)	(12.1)	(12.2)

Per_element (Scalar) Variable Example 1 This example shows an ASCII scalar file (`en6.Esca`) for the geometry example.

```
Per_elem scalar values for the EnSight6 geometry example
part 1
tria3
 2.00000E+00 3.00000E+00
hexa8
 4.00000E+00
part 2
bar2
 1.00000E+00
part 3
block
 5.00000E+00 6.00000E+00 7.00000E+00 8.00000E+00 9.00000E+00 1.00000E+01
 1.10000E+01 1.20000E+01
```

Per_element (Vector) Variable Example 2 This example shows an ASCII vector file (`en6.Evec`) for the geometry example.

```
Per_elem vector values for the EnSight6 geometry example
part 1
tria3
 2.10000E+00 2.20000E+00 2.30000E+00 3.10000E+00 3.20000E+00 3.30000E+00
hexa8
 4.10000E+00 4.20000E+00 4.30000E+00
```

```
part 2
bar2
 1.10000E+00 1.20000E+00 1.30000E+00
part 3
block
 5.10000E+00 6.10000E+00 7.10000E+00 8.10000E+00 9.10000E+00 1.01000E+01
 1.11000E+01 1.21000E+01
 5.20000E+00 6.20000E+00 7.20000E+00 8.20000E+00 9.20000E+00 1.02000E+01
 1.12000E+01 1.22000E+01
 5.30000E+00 6.30000E+00 7.30000E+00 8.30000E+00 9.30000E+00 1.03000E+01
 1.13000E+01 1.23000E+01
```

Per_element (Tensor) Variable Example 3 This example shows the ASCII 2nd order symmetric tensor file (en6.Eten) for the geometry example.

```
Per_elem symmetric tensor values for the EnSight6 geometry example
part 1
tria3
 2.10000E+00 2.20000E+00 2.30000E+00 2.40000E+00 2.50000E+00 2.60000E+00
 3.10000E+00 3.20000E+00 3.30000E+00 3.40000E+00 3.50000E+00 3.60000E+00
hexa8
 4.10000E+00 4.20000E+00 4.30000E+00 4.40000E+00 4.50000E+00 4.60000E+00
part 2
bar2
 1.10000E+00 1.20000E+00 1.30000E+00 1.40000E+00 1.50000E+00 1.60000E+00
part 3
block
 5.10000E+00 6.10000E+00 7.10000E+00 8.10000E+00 9.10000E+00 1.01000E+01
 1.11000E+01 1.21000E+01
 5.20000E+00 6.20000E+00 7.20000E+00 8.20000E+00 9.20000E+00 1.02000E+01
 1.12000E+01 1.22000E+01
 5.30000E+00 6.30000E+00 7.30000E+00 8.30000E+00 9.30000E+00 1.03000E+01
 1.13000E+01 1.23000E+01
 5.40000E+00 6.40000E+00 7.40000E+00 8.40000E+00 9.40000E+00 1.04000E+01
 1.14000E+01 1.24000E+01
 5.50000E+00 6.50000E+00 7.50000E+00 8.50000E+00 9.50000E+00 1.05000E+01
 1.15000E+01 1.25000E+01
 5.60000E+00 6.60000E+00 7.60000E+00 8.60000E+00 9.60000E+00 1.06000E+01
 1.16000E+01 1.26000E+01
```

Per_element (Complex) Variable Example 4 This example shows the ASCII complex real (`en6.Ecmp_r`) and imaginary (`en6.Ecmp_i`) *scalar* files for the geometry example. (The same methodology would apply for complex real and imaginary *vector* files).

Real scalar File:

```
Per_elem complex real scalar values for the EnSight6 geometry example
part 1
tria3
 2.10000E+00 3.10000E+00
hexa8
 4.10000E+00
part 2
bar2
 1.10000E+00
part 3
block
 5.10000E+00 6.10000E+00 7.10000E+00 8.10000E+00 9.10000E+00 1.01000E+01
 1.11000E+01 1.21000E+01
```

Imaginary scalar File:

```
Per_elem complex imaginary scalar values for the EnSight6 geometry example
part 1
tria3
 2.20000E+00 3.20000E+00
hexa8
 4.20000E+00
part 2
```

```
bar2
 1.20000E+00
part 3
block
 5.20000E+00 6.20000E+00 7.20000E+00 8.20000E+00 9.20000E+00 1.02000E+01
 1.12000E+01 1.22000E+01
```

EnSight6 Measured/Particle File Format

The format of a Measured/Particle geometry file is exactly the same as an EnSight 5 measured/particle geometry file and is repeated below as follows:

- Line 1

  This line is a description line.

- Line 2

  Indicates that this file contains particle coordinates. The words "`particle coordinates`" should be entered on this line without the quotes.

- Line 3

  Specifies the number of Particles.

- Line 4 through the end of the file.

  Each line contains the ID and the X, Y, and Z coordinates of each Particle. The format of this line is "integer real real real" written out in the following format:

  From C: `%8d%12.5e%12.5e%12.5e` format

  From FORTRAN: `i8, 3e12.5` format

A generic measured/Particle geometry file is as follows:

```
A description line
particle coordinates
#_of_Particles
id xcoord ycoord zcoord
id xcoord ycoord zcoord
id xcoord ycoord zcoord
 .
 .
 .
```

Measured Geometry Example

The following illustrates a measured/Particle file with seven points:

```
This is a simple measured geometry file
particle coordinates
       7
     101 0.00000E+00 0.00000E+00 0.00000E+00
     102 1.00000E+00 0.00000E+00 0.00000E+00
     103 1.00000E+00 1.00000E+00 0.00000E+00
     104 0.00000E+00 1.00000E+00 0.00000E+00
     205 5.00000E-01 0.00000E+00 2.00000E+00
     206 5.00000E-01 1.00000E+00 2.00000E+00
     307 0.00000E+00 0.00000E+00-1.50000E+00
```

Measured Variable Files

Measured variable files use the same format as EnSight6 per_node variable files, which is exactly the same as the EnSight 5 measured/particle variable files.

Writing EnSight6 Binary Files

This section describes the EnSight6 binary files. This format is used to increase the speed of reading data into EnSight.

For binary files, there is a header that designates the type of binary file. This header is: “C Binary” or “Fortran Binary.” This must be the first thing in the geometry file only. The format for the file is then essentially the same format as the ASCII format, with the following exceptions:

The ASCII format puts the node and element ids on the same “line” as the corresponding coordinates. The BINARY format writes all node id’s then all coordinates.

The ASCII format puts all element id’s of a type within a Part on the same “line” as the corresponding connectivity. The BINARY format writes all the element ids for that type, then all the corresponding connectivities of the elements.

FORTRAN binary files should be created as sequential access unformatted files.

Float arrays (such as coordinates and variable values) must be single precision. Double precision is not supported.

In all the descriptions of binary files that follow, the number on the left end of the line corresponds to the type of write of that line, according to the following code:

1. This is a write of 80 characters to the file:

C example:
```
char buffer[80];
strcpy(buffer,"C Binary");
fwrite(buffer,sizeof(char),80,file_ptr);
```

FORTRAN:
```
character*80   buffer
buffer = "Fortran Binary"
write(10) buffer
```

2. This is a write of a single integer:

C example: `fwrite(&num_nodes,sizeof(int),1,file_ptr);`

FORTRAN: `write(10) num_nodes`

3. This is a write of an integer array:

C example: `fwrite(node_ids,sizeof(int),num_nodes,file_ptr);`

FORTRAN: `write(10) (node_ids(i),i=1,num_nodes)`

4. This is a write of a float array:

C example: `fwrite(coords,sizeof(float),3*num_nodes,file_ptr);`

FORTRAN: `write(10) ((coords(i,j),i=1,3),j=1,num_nodes)`

*Note: For EnSight6 format, when using **Fortran binary**, and writing arrays composed of components, such as coordinates, or vector values, it is **very important** that they all be written with a **single Fortran write statement**. If you instead were to write the coords in the statement above with a loop per component, such that the write statement is executed three times, like the following, **EnSight will not be able to read it!***

FORTRAN:
```
do 200 i=1,3
       write(10) (coords(i,j),j=1,num_nodes)
200 continue
```

EnSight6 Binary Geometry

An EnSight binary geometry file contains information in the following order:

(1) <C Binary/Fortran Binary>
(1) description line 1
(1) description line 2
(1) node id <given/off/assign/ignore>
(1) element id <given/off/assign/ignore>
(1) coordinates
(2) #_of_points
(3) [point_ids]
(4) coordinate_array *(For FORTRAN make sure only one write statement is used)*
(1) part #
(1) description line
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
:
(1) part #
(1) description line
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
(1) part #
(1) description line
(1) block [iblanked]
(3) i j k
(4) all i coords, all j coords, all k coords *(For FORTRAN make sure only one write statement is used)*
(3) [iblanking]
:

Per_node Binary Scalar

An EnSight6 binary scalar file contains information in the following order:

(1) description line
(4) scalar_array for unstructured nodes
(1) part #
(1) block
(4) scalar_array for part's structured nodes

Per_node Binary Vector

An EnSight6 binary vector file contains information in the following order:

(1) description line
(4) vector_array for unstructured nodes *(For FORTRAN make sure only one write statement is used)*

(1) part #
(1) block
(4) vector_array for part's structured nodes *(For FORTRAN make sure only one write statement is used)*

Per_node Binary Tensor An EnSight6 binary tensor file contains information in the following order:

(1) description line
(4) tensor_array for unstructured nodes *(For FORTRAN make sure only one write statement is used)*
(1) part #
(1) block
(4) tensor_array for part's structured nodes *(For FORTRAN make sure only one write statement is used)*

Per_node Binary Complex An EnSight6 binary complex real and imaginary *scalar* files contain information in the following order: (The same methodology applies for the complex real and imaginary vector files.)

Real scalar file:

(1) description line
(4) real scalar_array for unstructured nodes
(1) part #
(1) block
(4) real scalar_array for part's structured nodes

Imaginary scalar file:

(1) description line
(4) imaginary scalar_array for unstructured nodes
(1) part #
(1) block
(4) imaginary scalar_array for part's structured nodes

Per_element Binary Scalar An EnSight6 binary scalar file contains information in the following order:

(1) description line
(1) part #
(1) element type (tria3, quad4, ...)
(4) scalar_array for elements of part and type
(1) part #
(1) block
(4) scalar_array for structured elements of part

Per_element Binary Vector An EnSight6 binary vector file contains information in the following order:

(1) description line
(1) part #
(1) element type (tria3, quad4, ...)
(4) vector_array for elements of part and type *(For FORTRAN make sure only one write statement is used)*
(1) part #
(1) block
(4) vector_array for structured elements of part *(For FORTRAN make sure only one write statement is used)*

Per_element Binary Tensor An EnSight6 binary tensor file contains information in the following order:

(1) description line
(1) part #
(1) element type (tria3, quad4, ...)
(4) tensor_array for unstructured elements of part and type *(For FORTRAN make sure only one write statement is used)*
(1) part #
(1) block
(4) tensor_array for structured elements of part and type *(For FORTRAN make sure only one write statement is used)*

Per_element Binary Complex EnSight6 binary complex real and imaginary *scalar* files contain information in the following order: (The same methodology applies for the complex real and imaginary vector files.)

Real scalar file:

(1) description line
(1) part #
(1) element type (tria3, quad4, ...)
(4) real scalar_array for unstructured elements of part and type
(1) part #
(1) block
(4) real scalar_array for structured elements of part and type

Imaginary scalar file:

(1) description line
(1) part #
(1) element type (tria3, quad4, ...)
(4) imaginary scalar_array for unstructured elements of part and type
(1) part #
(1) block
(4) imaginary scalar_array for structured elements of part and type

Binary Measured Geometry An EnSight6 binary measured/particle geometry file contains information in the following order:

(1) <C Binary/Fortran Binary>
(1) description line 1
(1) particle coordinates
(2) #_of_points
(3) point_ids
(4) coordinate_array *(For FORTRAN make sure only one write statement is used)*

Binary Measured Variable Files EnSight6 binary measured/discrete particle scalar and vector files follow the same binary formats as EnSight6 model per-node scalar and vector files.

11.3 EnSight5 Format

Included in this section:

EnSight5 General Description

Note: The EnSight6 format replaces and includes all aspects of the older EnSight5 format. This description is included for completeness but use of the EnSight6 format with EnSight 6.x and later versions is encouraged!

EnSight5 data consists of the following files:

- Geometry (required)
- Results (optional) (points to other variable files and possibly to changing geometry files)
- Measured (optional) (points to measured geometry and variable files)

The results file contains information concerning scalar and vector variables. EnSight makes no assumptions regarding the physical significance of the scalar and vector variables. These files can be from any discipline. For example, the scalar file can include such things as pressure, temperature, and stress. The vector file can be velocity, displacement, or any other vector data.

All variable results for EnSight5 are contained in disk files—one variable per file. Additionally, if there are multiple time steps, there must be a set of disk files for each time step.

Sources of EnSight5 data include the following:

- Data that can be translated to conform to the EnSight5 data format
- Data that originates from one of the translators supplied with the EnSight application

The EnSight5 format supports a defined element set as shown below. The data must be defined in this element set. Elements that do not conform to this set must either be subdivided or discarded.

Supported EnSight5 Elements

The elements that are supported by the EnSight5 format are:

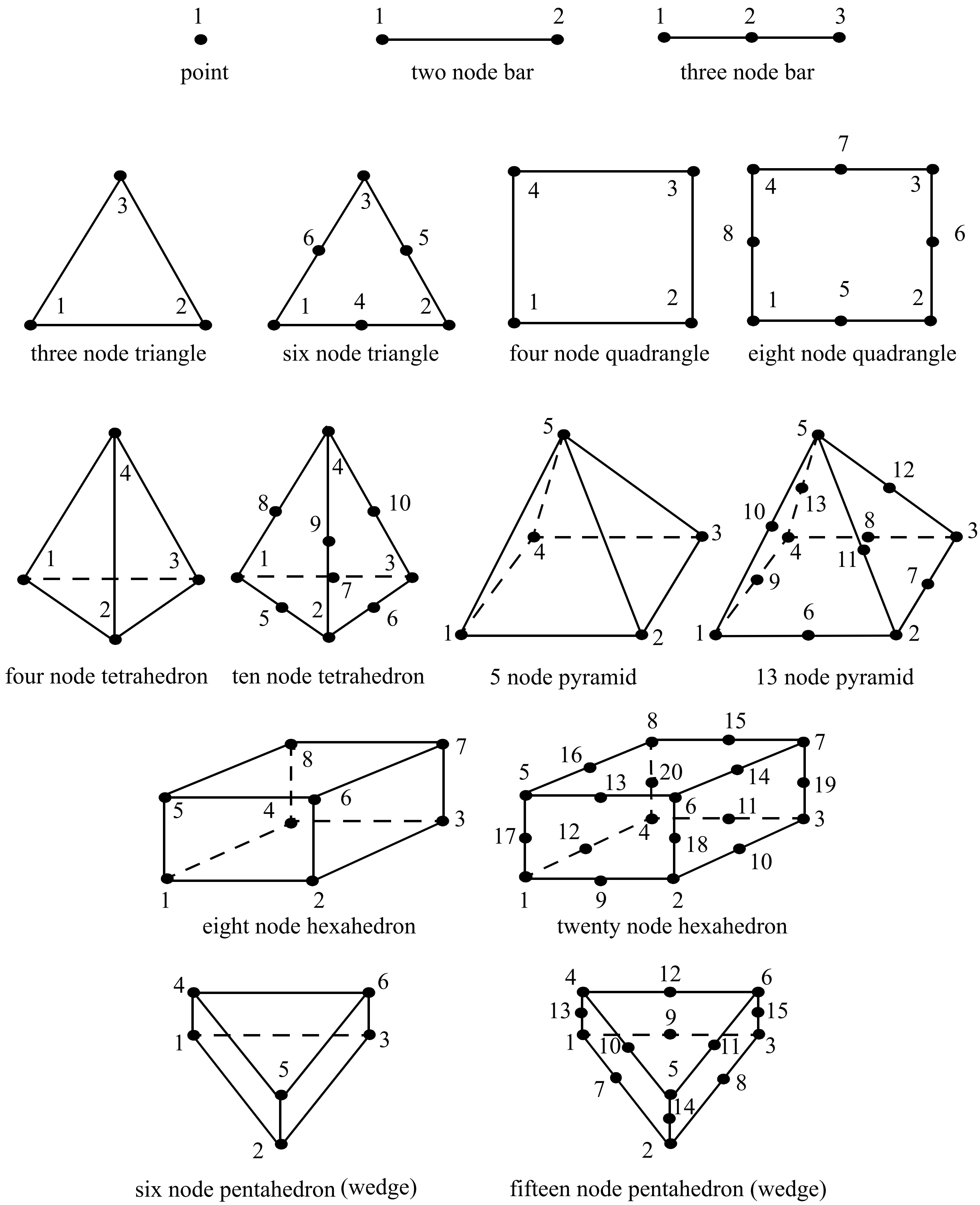


Figure 11-5
Supported EnSight5 Elements

EnSight5 Geometry File Format

The EnSight5 format consists of keywords followed by information. The following items are important to remember when working with EnSight5 geometry files:

1. You do not have to assign node IDs. If you do, the element connectivities are based on the node numbers. If you let EnSight assign the node IDs, the nodes are considered to be sequential starting at node 1, and element connectivity is done accordingly. If node IDs are set to off, they are numbered internally; however, you will not be able to display or query on them. If you have node IDs in your data, you can have EnSight ignore them by specifying “node id ignore.” Using this option may reduce some of the memory taken up by the Client and Server, but remember that display and query on the nodes will not be available.
2. You do not need to specify element IDs. If you specify element IDs, or you let EnSight assign them, you can show them on the screen. If they are set to off, you will not be able to show or query on them. If you have element IDs in your data you can have EnSight ignore them by specifying “element id ignore.” Using this option will reduce some of the memory taken up by the Client and Server. This may or may not be a significant amount, and remember that display and query on the elements will not be available.
3. The format of integers and real numbers **must be followed** (See the Geometry Example below).
4. Integers are written out using the following integer format:

From C:	`8d` format
From FORTRAN:	`i8` format

   Real numbers are written out using the following floating-point format:

From C:	`12.5e` format
From FORTRAN:	`e12.5` format

   The number of integers or reals per line must also be followed!
5. By default, a Part is processed to show the outside boundaries. This representation is loaded to the Client host system when the geometry file is read (unless other attributes have been set on the workstation, such as feature angle).
6. Coordinates must be defined before any Parts can be defined. The different elements can be defined in any order (that is, you can define a hexa8 before a bar2).

Generic Format

Not all of the lines included in the following generic example file are necessary:

```
description line 1
description line 2
node id <off/given/assign/ignore>
element id <off/given/assign/ignore>
coordinates
# of points
id x y z
id x y z
```

```
id x y z
.
.
.
part #
description line
point
number of points
id nd
id nd
id nd
.
.
.
bar2
number of bar2's
id nd nd
id nd nd
id nd nd
.
.
.
bar3
number of bar3's
id nd nd nd
id nd nd nd
id nd nd nd
.
.
.
tria3
number of three node triangles
id nd nd nd
id nd nd nd
id nd nd nd
.
.
.
tria6
number of six node triangles
id nd nd nd nd nd nd
.
.
.
quad4
number of quad 4's
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
.
.
.
quad8
number of quad 8's
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
.
.
```

```
.
tetra4
number of 4 node tetrahedrons
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
id nd nd nd nd
.
.
.
tetra10
number of 10 node tetrahedrons
id nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd
.
.
.
pyramid5
number of 5 node pyramids
id nd nd nd nd nd
id nd nd nd nd nd
id nd nd nd nd nd
id nd nd nd nd nd
.
.
.
pyramid13
number of 13 node pyramids
id nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd
.
.
.
hexa8
number of 8 node hexahedrons
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd
.
.
.
hexa20
number of 20 node hexahedrons
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
.
.
.
penta6
number of 6 node pentahedrons
id nd nd nd nd nd nd
id nd nd nd nd nd nd
id nd nd nd nd nd nd
```

```
id nd nd nd nd nd nd
id nd nd nd nd nd nd
.
.
.
penta15
number of 15 node pentahedrons
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
id nd nd nd nd nd nd nd nd nd nd nd nd nd nd nd
.
.
.
```

EnSight5 Geometry Example

The following is an example of an EnSight geometry file:

```
this is an example problem
this is the second description line
node id given
element id given
coordinates
      10
       5 1.00000e+00 0.00000e+00 0.00000e+00
     100 0.00000e+00 1.00000e+00 0.00000e+00
     200 0.00000e+00 0.00000e+00 1.00000e+00
      40 1.00000e+00 1.00000e+00 0.00000e+00
      22 1.00000e+00 0.00000e+00 1.00000e+00
    1000 2.00000e+00 0.00000e+00 0.00000e+00
      55 0.00000e+00 2.00000e+00 0.00000e+00
      44 0.00000e+00 0.00000e+00 2.00000e+00
     202 2.00000e+00 2.00000e+00 0.00000e+00
     101 2.00000e+00 0.00000e+00 2.00000e+00
part 1
This is Part 1, a pretty strange Part
tria3
       2
     101     100     200      40
     201     101       5    1000
tetra4
       1
     102     100     202     101    1000
part 2
This is Part 2, it’s pretty strange also
bar2
       1
     103     101    1000
```

EnSight5 Result File Format

The Result file is an ASCII free format file that contains variable and time step information that pertains to a Particular geometry file. The following information is included in this file:

- Number of scalar variables
- Number of vector variables
- Number of time steps
- Starting file number extension and skip-by value
- Flag that specifies whether there is changing geometry
- Names of the files that contain the values of scalar and vector variables
- The names of the geometry files that will be used for the changing geometry.

The format of the EnSight5 result file is as follows:

- Line 1

  Contains the number of scalar variables, the number of vector variables and a geometry-changing flag. (If the geometry-changing flag is 0, the geometry of the model does not change over time. If it is 1, then there is connectivity changing geometry. If it is 2, then there is coordinate only changing geometry.)

- Line 2

  Indicates the number of time steps that are available.

- Line 3

  Lists the time that is associated with each time step. There must be the same number of values as are indicated in Line 2. This “line” can actually span several lines in the file. You do not have to have one very long line.

- Line 4

  Specified only if more than one time step is indicated in Line 2. The two values on this line indicate the file extension value for the first time step and the offset between files. If the values on this line are 0 5, the first time step available has a subscript of 0, the second time step available has a subscript of 5, the third time step has a subscript of 10, and so on.

- Line 5

  Contains the names of the geometry files that will be used for changing geometry. This line exists only if the flag on Line 1 is set to 1 or 2. The geometry file name must follow the EnSight5 wild card specification.

- Line 6 through Line [5+N] where N is the number of scalar variables specified in Line 1.

  List **BOTH** the file names **AND** variable description that correspond to each scalar variable. There must be a file name for each scalar variable that is specified in Line 1.

If there is more than one time step, the file name must follow the EnSight5 wild card specification. See Note below.

- Lines that follow the scalar variable files.

  List the file names that correspond to each vector variable. There must be a file name for each vector variable that is specified in Line 1. If there is more than one time step, the file name must follow the EnSight5 wild card specification. See Note below.

Note: *Variable descriptions have the following restrictions:*

The maximum variable name length is documented at the beginning of this chapter.

Duplicate variable descriptions are not allowed.

Leading and trailing white space will be eliminated.

Variable descriptions must not start with a numeric digit.

Variable descriptions must not contain any of the following reserved characters:

(	[	+	@	!	*	$
)	]	-	space	#	^	/

The generic format of a result file is as follows:

```
#_of_scalars #_of_vectors geom_chang_flag
#_of_timesteps
time1 time2 time3 .....
start_file_# skip_by_value
geometry_file_name.geo**
scalar0_file_name**   description (19 characters max)
scalar1_file_name**   description
.
.
.
vector0_file_name**   description (19 characters max)
vector1_file_name**   description
.
```

EnSight5 Result File Example 1

The following example illustrates a result file specified for a non-changing geometry file with only one time step:

```
2 1 0
1
0.0
exone.scl0 pressure
exone.scl1 temperature
exone.dis0 velocity
```

EnSight5 Result File Example 2

This example illustrates a result file that specifies a connectivity changing geometry that has multiple time steps.

```
1 2 1
4
1.0 2.0 2.5 5.0
0 1
extwo.geom**
pres.scl** pressure
vel.dis** velocity
grad.dis** gradient
```

The following files would be needed for example 2:

```
extwo.geom00 pres.scl00 vel.dis00 grad.dis00
extwo.geom01 pres.scl01 vel.dis01 grad.dis01
extwo.geom02 pres.scl02 vel.dis02 grad.dis02
extwo.geom03 pres.scl03 vel.dis03 grad.dis03
```

EnSight5 Wild Card Name Specification

If multiple time steps are involved, the file names must conform to the EnSight5 wild-card specification. This specification is as follows:

- File names must include numbers that are in ascending order from beginning to end.
- Numbers in the files names must be zero filled if there is more than one significant digit.
- Numbers can be anywhere in the file name.
- When the file name is specified in the EnSight5 result file, you must replace the numbers in the file with an asterisk(*). The number of asterisks specified is the number of significant digits. The asterisk must occupy the same place as the numbers in the file names.

EnSight5 Variable File Format

Variables files have one description line followed by a value for each node. For a scalar file there is one value per node, while for vector files there are three values per node.

The values **must be written** in the following floating point format (**6 per line** as shown in the examples below):

From C: `12.5e` format

From FORTRAN: `e12.5` format

The format of a variables file is as follows:

- Line 1

  This line is a description line.

- Line 2 through the end of the file contains the values at each node in the model. A generic example:

```
A description line
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
 *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+** *.*****E+**
```

EnSight5 Variable File Example 1

This example shows a scalar file for a geometry with seven defined nodes.

```
These are the pressure values for a 7 node geometry
 1.00000E+00 2.00000E+00 3.00000E+00 4.00000E+00 5.00000E+00 6.00000E+00
 7.00000E+00
```

EnSight5 Variable File Example 2

This example shows the vector file for a geometry with seven defined nodes.

```
These are the velocity values for a 7 node geometry
```

```
1.00000E+00 1.00000E+00 1.00000E+00 2.00000E+00 2.00000E+00 2.00000E+00
3.00000E+00 3.00000E+00 3.00000E+00 4.00000E+00 4.00000E+00 4.00000E+00
5.00000E+00 5.00000E+00 5.00000E+00 6.00000E+00 6.00000E+00 6.00000E+00
7.00000E+00 7.00000E+00 7.00000E+00
```

EnSight5 Measured/Particle File Format

This file allows you to define Particle locations, sizes, etc. to display with the geometry. Typical uses are fuel droplets for combustion analysis or data derived from experiments on prototypes.

The measured/Particle files consist of the following:

- Measured/Particle geometry file (referenced by the measured results file)
- Measured/Particle results file (the filename, typically with a .mea suffix, which is put into the Data Reader's "(Set) Measured" field)
- Measured/Particle variables file (referenced by the measured results file) which is the same format as an EnSight 5 variable file.

The format of the EnSight5 Measured/Particle geometry file is described below.

Note that there is only one description line and there *must* be an ID for each measured point.

Note also that the number of Particles can be different in each of the geometry file (if you have transient data), however, the number of values in each of the corresponding variable files must coincide, and the IDs of the Particles must be consistent in order to track the Particles at intermediate times or locations.

The format of an EnSight5 Measured/Particle geometry file is as follows:

- Line 1

  This line is a description line.
- Line 2

  Indicates that this file contains Particle coordinates. The words "particle coordinates" should be entered on this line without the quotes.
- Line 3

  Specifies the number of Particles.
- Line 4 through the end of the file.

  Each line contains the ID and the X, Y, and Z coordinates of each Particle. The format of this line is "integer real real real" written out in the following format:

  From C: `%8d%12.5e%12.5e%12.5e` format

  From FORTRAN: `i8, 3e12.5` format

A generic measured/Particle geometry file is as follows:

```
A description line
particle coordinates
#_of_Particles
```

```
id xcoord ycoord zcoord
id xcoord ycoord zcoord
id xcoord ycoord zcoord
 .
 .
 .
```

EnSight5 Measured Geometry/Particle File Example

The following illustrates an EnSight5 Measured Geometry/Particle file with seven points:

```
This is a simple ensight5 measured geometry/particle file
particle coordinates
       7
     101 0.00000E+00 0.00000E+00 0.00000E+00
     102 1.00000E+00 0.00000E+00 0.00000E+00
     103 1.00000E+00 1.00000E+00 0.00000E+00
     104 0.00000E+00 1.00000E+00 0.00000E+00
     205 5.00000E-01 0.00000E+00 2.00000E+00
     206 5.00000E-01 1.00000E+00 2.00000E+00
     307 0.00000E+00 0.00000E+00-1.50000E+00
```

EnSight5 Measured/ Particle File Format

The format of the EnSight5 Measured/Particle results file (typically a .mea suffix) is as follows:

- Line 1

  Contains the number of scalar variables, the number of vector variables, and a measured geometry changing flag. If the measured geometry changing flag is 0, only one time step is indicated.

- Line 2

  Indicates the number of available time steps.

- Line 3

  Lists the time that is associated with each time step. The time step information does not have to coincide with the model time step information. This “line” can actually span several lines in the file. You do not have to have one very long line.

- Line 4

  Specified only if Line 2 specifies more than one time step. The line contains two values; the first value indicates the file extension value for the first time step, and the second value indicates the offset between files. If this line contains the values 0 and 5, the first time step has a subscript of 0, the second of 5, the third of 10, and so on.

- Line 5

  Contains the name of the measured geometry file. If there is more than one time step, the file name must follow the EnSight wild card specification.

- Line 6 through Line [5+N] where N is the number of scalar variables specified in Line 1.

List the file names that correspond to each scalar variable. There must be a file name for each scalar variable that is specified in Line 1. If there is more than one time step, the file name must follow the EnSight wild card specification.

- Lines that follow the scalar variable files.

  List the names of the files that correspond to each vector variable. There must be a file name for each vector variable that is specified in Line 1. If there is more than one time step, the file name must follow the EnSight wild card specification.

A generic EnSight5 Measured/Particle results file is as follows:

```
#_of_scalars #_of_vectors geom_chang_flag
#_of_timesteps
time1 time2 time3 .....
start_file_# skip_by_value
measured_geom_file_name**
scalar0_file_name**   description
scalar1_file_name**   description
.
.
.
vector0_file_name**   description
vector1_file_name**   description
.
.
.
```

Measured/Particle Results File Example 1

This example illustrates an EnSight5 Measured/Particle result file that specifies a non-changing geometry with only one time step:

```
2 1 0
1
0.0
exone.geom
exone.scl0 pressure
exone.scl1 temperature
exone.dis0 velocity
```

Measured/Particle Results File Example 2

This example illustrates an EnSight5 Measured/Particle result file that specifies a changing geometry with multiple time steps:

```
1 2 1
4
1.0 2.0 2.5 5.0
0 1
extwo.geom**
pres.scl** pressure
vel.dis** velocity
grad.dis** gradient
```

The following files are needed for Example 2:

```
extwo.geom00pres.scl00vel.dis00       grad.dis00
extwo.geom01pres.scl01vel.dis01       grad.dis01
extwo.geom02pres.scl02vel.dis02       grad.dis02
extwo.geom03pres.scl03vel.dis03       grad.dis03
```

Measured /Particle Results Variable files

The EnSight5 Measured/Particle variable files referred to in the measured Results file follow the same format as EnSight5 Variable files. The number of values in each of these variable files must correspond properly to the number of Particles in the corresponding measured geometry files.

Writing EnSight5 Binary Files

This section describes the EnSight5 binary files. This format is used to increase the speed of reading data into EnSight. A utility exists for converting EnSight5 ASCII files to EnSight5 binary files—it is called asciitobin5 and is found on the release tape under ensight/server/utilities/asciitobin5.

For binary files, there is a header that designates the type of binary file. This header is: "C Binary" or "Fortran Binary." This must be the first thing in the file. The format for the file is then essentially the same format as the ASCII format, with the following exceptions:

The ASCII format puts the node and element ids on the same "line" as the corresponding coordinates. The BINARY format writes all node id's then all coordinates.

The ASCII format puts all element id's of a type within a Part on the same "line" as the corresponding connectivity. The BINARY format writes all the element ids for that type, then all the corresponding connectivities of the elements.

In all the descriptions of binary files that follow, the number on the left end of the line corresponds to the type of write of that line, according to the following code:

1. This is a write of 80 characters to the file:

   C example:
   ```
   char buffer[80];
   strcpy(buffer,"C Binary");
   fwrite(buffer,sizeof(char),80,file_ptr);
   ```

   FORTRAN:
   ```
   character*80   buffer
   buffer = "Fortran Binary"
   write(10) buffer
   ```

2. This is a write of a single integer:

   C example: `fwrite(&num_nodes,sizeof(int),1,file_ptr);`

   FORTRAN: `write(10) num_nodes`

3. This is a write of an integer array:

   C example: `fwrite(node_ids,sizeof(int),num_nodes,file_ptr);`

   FORTRAN: `write(10) (node_ids(i),i=1,num_nodes)`

4. This is a write of a float array:

   C example: `fwrite(coords,sizeof(float),3*num_nodes,file_ptr);`

FORTRAN: `write(10) ((coords(i,j),i=1,3),j=1,num_nodes)`

(Note: Coords is a single precision array, double precision will not work!)

EnSight5 Binary Geometry File Format

An EnSight5 binary geometry file contains information in the following order:

(1) <C Binary/Fortran Binary>
(1) description line 1
(1) description line 2
(1) node id <given/off/assign/ignore>
(1) element id <given/off/assign/ignore>
(1) coordinates
(2) #_of_points
(3) [point_ids]
(4) coordinate_array
(1) part #
(1) description line
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
.
.
.
(1) part #
(1) description line
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
(1) element_type
(2) #_of_element_type
(3) [element_ids] for the element_type
(3) connectivities for the element_type
.
.
.

Binary Scalar

An EnSight5 binary scalar file contains information in the following order:

(1) description line
(4) scalar_array

Binary Vector

An EnSight5 binary vector file contains information in the following order:

(1) description line

(4) vector_array

Binary Measured

An EnSight5 binary measured/Particle geometry file contains information in the following order:

(1) <C Binary/Fortran Binary>
(1) description line 1
(1) particle coordinates
(2) #_of_points
(3) point_ids
(4) coordinate_array

11.4 FAST UNSTRUCTURED Results File Format

FAST UNSTRUCTURED input data consists of the following:

- Geometry file (required) (GRID file).
- Results file (optional).
- EnSight5 Measured/Particle Files (optional). The measured .res file references the measured geometry and variable files.

FAST UNSTRUCTURED data files can be read as:

Workstation: ASCII, C Binary, or FORTRAN binary

Cray: ASCII, C Binary, or COS-Blocked FORTRAN binary

Due to the different number of representations on a Cray Research vector system and workstations, binary files created on a Cray Research vector system can *not* be read on the workstation, and visa versa.

EnSight reads the geometry (grid files) directly. However, an EnSight-like results file is needed in order to read the results unless a "standard" Q-file is provided in its place. See FAST UNSTRUCTURED Result File below.

FAST UNSTRUCTURED Geometry file notes

Only the single zone format can be read into EnSight. Any tetrahedral elements will be placed into the first "domain" Part. Triangular elements are placed into Parts based on their "tag" value.

The FAST UNSTRUCTURED solution file or function file formats can be used for variable results. The I J K values need to be I=Number of points and J=K=1. This does require the use of a modified EnSight results file as explained below.

Node and element numbers are assigned sequentially allowing for queries to be made within EnSight. Tetrahedron elements will be assigned before triangular elements.

FAST UNSTRUCTURED Result file format

The FAST UNSTRUCTURED result file was defined by CEI and is very similar to the EnSight results file and contains information needed to relate variable names to variable files, step information, etc. There is a slight variation from the normal EnSight results file because of the differences between the solution (Q file) and function files. The difference lies on the lines which relate variable filenames to a description. These lines have the following format:

```
<filename> <type> <number(s)> <description>
```

See FAST UNSTRUCTURED Result File below for the definition of each.

The following information is included in a FAST UNSTRUCTURED result file:

- Number of scalar variables
- Number of vector variables
- Number of time steps

- Starting file number extension and skip-by value
- Flag that specifies whether there is changing geometry.
- Names of the files that contain the values of scalar and vector variables. An indication as to the type of the file being used for the variable, which variable in the file and the name given to that variable.
- The names of the geometry files that will be used for the changing geometry.

Generic FAST UNSTRUCTURED Result File Format

The format of the Result file is as follows:

- Line 1

  Contains the number of scalar variables, the number of vector variables and a geometry changing flag. If the geometry changing flag is 0, the geometry of the model does not change over time. If the flag is 1, the geometry can change connectivity. If the flag is 2, only coordinates can change.

- Line 2

  Indicates the number of time steps that are available. If this number is positive, then line 3 information must be present. If this number is negative, then Line 3 information must not be present and the times will be read from the solution file. Thus, one must have a solution file in one of the lines from Line 6 on.

- Line 3

  Lists the time that is associated with each time step. There must be the same number of values as are indicated in Line 2. This “line” can actually span several lines in the file. Specify only if Line 2 value is positive.

- Line 4

  Specified only if more than one time step is indicated in Line 2. The two values on this line indicate the file extension value for the first time step and the offset between files. If the values on this line are 0 5, the first time step available has a subscript of 0, the second time step available has a subscript of 5, the third time step has a subscript of 10, and so on.

- Line 5

  This line exists only if the changing geometry flag on Line 1 has been set to 1 or 2. Line contains name of the FAST UNSTRUCTURED grid file. The file name must follow the EnSight wild card specification.

- Line 6 through Line [5+N] where N is the number of scalar variables specified in Line 1.

  List the file names that correspond to each scalar variable. There must be a file name for each scalar variable that is specified in Line 1. If there is more than one time step, the file name must follow the EnSight wild card specification.

  These lines also contain the type of file being used, solution or function, and the location of the variable value in the file. The contents are:

```
<filename> <type> <number> <description>
```

where filename is the name of solution file or function file containing the variable; type is “S” for solution file, or “F” for function file; numberis which variable in the file to use (specify just one number); and `description` is the Description of the variable.

The solution file (“s”) is the traditional .q file in which normally the first variable is density, the second through fourth variables are the components of momentum, and the fifth variable is total energy.

- Lines that follow the scalar variable files.

  List the file names that correspond to each vector variable. There must be a file name for each vector variable that is specified in Line 0. If there is more than one time step, the file name must follow the EnSight wild card specification.

  These lines also contain the type of file being used, solution or function, and the location(s) of the variable values in the file. The contents are:

  ```
  <filename> <type> <numbers> <description>
  ```

  where filename is the name of solution file or function file containing the variable; type is “S” for solution file, or “F” for function file; numbersare which variables in the file to use (specify just three numbers); and description is the Description of the variable.

The generic format of the result file is as follows:

```
#_of_scalars #_of_vectors geom_chng_flag
#_of_timesteps
time1 time2 time3 .....
start_file_# skip_by_value
geometry_file_name.geo**
scalar0_file_name** type # description
scalar1_file_name** type # description
.
.
.
vector0_file_name** type # # # description
vector1_file_name** type # # # description
.
.
.
```

FAST UNSTRUCTURED Example

This example illustrates a result file that specifies a non-changing geometry with only one time step.

```
3 2 0
1
0.0
block.sol S 1 Density
block.sol S 5 Total_Energy
block.scl F 1 Temperature
block.var F 1 2 3 Displacement
block.sol S 2 3 4 Momentum
```

Thus, this model will get two scalars from the solution file (block.sol). The first is Density in the first location in the file and the next is Total energy in the fifth location in the solution file. It will also get a Temperature scalar from the first location in the function file (block.scl).

It will get a Displacement vector from the function file called block.var. The three components of this vector are in the 1st, 2nd, and 3rd locations in the file. Finally, a Momentum vector will be obtained from the 2nd, 3rd, and 4th locations of the solution file.

Example 2 is somewhat similar, except that it is transient, with coordinate changing geometry. Note also that the times will come from the solution file.

```
3 2 2
-10
0 1
block***.grid
block***.sol S 1 Density
block***.sol S 5 Total_Energy
block***.scl F 1 Temperature
block***.var F 1 2 3 Displacement
block***.sol S 2 3 4 Momentum
```

11.5 FLUENT UNIVERSAL Results File Format

This section describes the FLUENT results file format and provides an example of this file. For transient cases, you *must* supply this result file. For static models this file is not required. The FLUENT result file is a slightly modified EnSight5 results file and provides a way to describe multiple time-step FLUENT Universal files to EnSight.

When using multiple FLUENT files with this result file definition, you *must* make sure that the files contain the same defined variables. In other words, any variable that exists in one must exist in all.

The Result file is an ASCII free format file that contains time step and universal file information for each available time step. The following information is included in this file:

- Number of time steps
- Simulation Time Values
- Starting file number extension and skip-by value
- Name of the universal file with EnSight wild card specification.

The format of the Result file is as follows:

- Line 1

  Indicates the number of time steps that are available.

- Line 2

  Lists the time that is associated with each time step. There must be the same number of values as are indicated in Line 1. This "line" can actually span several lines in the file. You do not have to have one very long line.

- Line 3

  Specified only if more than one time step is indicated in Line 1. The two values on this line indicate the file extension value for the first time step and the offset between files. If the values on this line are 0 5, the first time step available has a subscript of 0, the second time step available has a subscript of 5, the third time step has a subscript of 10, and so on.

- Line 4

  Contains the names of the universal file that will be used for the changing time step information. The universal file name must follow the EnSight5 wild card specification.

The generic format of the result file is as follows:

```
#_of_timesteps
time1 time2 time3 .....
start_file_# skip_by_value
universal_file_name***
```

FLUENT Example

This example illustrates a FLUENT result file

```
4
1.0 2.0 3.0 4.0
0 1
extwo**.uni
```

The following FLUENT universal files will need to exist for the result file:

```
extwo00.uni
extwo01.uni
extwo02.uni
extwo03.uni
```

11.6 Movie.BYU Results File Format

For transient cases, you must supply an EnSight result file. The result file for the Movie.BYU case is exactly the same as for EnSight5 (it is repeated below for your ease).

The Result file is an ASCII free format file that contains variable and time step information that pertains to a Particular geometry file. The following information is included in this file:

- Number of scalar variables
- Number of vector variables
- Number of time steps
- Starting file number extension and skip-by value
- Flag that specifies whether there is changing geometry
- Names of the files that contain the values of scalar and vector variables
- The names of the geometry files that will be used for the changing geometry.

The format of the Movie.BYU (EnSight5) result file is as follows:

- Line 1

  Contains the number of scalar variables, the number of vector variables and a geometry-changing flag. (If the geometry-changing flag is 0, the geometry of the model does not change over time. If it is 1, then there is connectivity changing geometry. If it is 2, then there is coordinate only changing geometry.)

- Line 2

  Indicates the number of time steps that are available.

- Line 3

  Lists the time that is associated with each time step. There must be the same number of values as are indicated in Line 2. This “line” can actually span several lines in the file. You do not have to have one very long line.

- Line 4

  Specified only if more than one time step is indicated in Line 2. The two values on this line indicate the file extension value for the first time step and the offset between files. If the values on this line are 0 5, the first time step available has a subscript of 0, the second time step available has a subscript of 5, the third time step has a subscript of 10, and so on.

- Line 5

  Contains the names of the geometry files that will be used for changing geometry. This line exists only if the flag on Line 1 is set to 1 or 2. The geometry file name must follow the EnSight5 wild card specification.

- Line 6 through Line [5+N] where N is the number of scalar variables specified in Line 1.

List **BOTH** the file names **AND** variable description that correspond to each scalar variable. There must be a file name for each scalar variable that is specified in Line 1.

If there is more than one time step, the file name must follow the EnSight5 wild card specification. See Note below.

- Lines that follow the scalar variable files.

  List the file names that correspond to each vector variable. There must be a file name for each vector variable that is specified in Line 1. If there is more than one time step, the file name must follow the EnSight5 wild card specification. See Note below.

Note: *Variable descriptions have the following restrictions:*

The maximum variable name length is documented at the beginning of this chapter.

Duplicate variable descriptions are not allowed.

Leading and trailing white space will be eliminated.

Variable descriptions must not start with a numeric digit.

Variable descriptions must not contain any of the following reserved characters:

(	[	+	@	!	*	$
)	]	-	space	#	^	/

The generic format of a result file is as follows:

```
#_of_scalars #_of_vectors geom_chang_flag
#_of_timesteps
time1 time2 time3 .....
start_file_# skip_by_value
geometry_file_name.geo**
scalar0_file_name**   description (19 characters max)
scalar1_file_name**   description
.
.
.
vector0_file_name**   description (19 characters max)
vector1_file_name**   description
.
```

Movie.BYU Result File Example 1

The following example illustrates a result file specified for a non-changing geometry file with only one time step:

```
2 1 0
1
0.0
exone.scl0 pressure
exone.scl1 temperature
exone.dis0 velocity
```

Movie.BYU Result File Example 2

This example illustrates a result file that specifies a connectivity changing geometry that has multiple time steps.

```
1 2 1
4
1.0 2.0 2.5 5.0
0 1
```

```
extwo.geom**
pres.scl** pressure
vel.dis** velocity
grad.dis** gradient
```

The following files would be needed for example 2:

```
extwo.geom00 pres.scl00 vel.dis00 grad.dis00
extwo.geom01 pres.scl01 vel.dis01 grad.dis01
extwo.geom02 pres.scl02 vel.dis02 grad.dis02
extwo.geom03 pres.scl03 vel.dis03 grad.dis03
```

11.7 PLOT3D Results File Format

PLOT3D input data consists of the following:

- Geometry file (required) (GRID file).
- Results file (optional).
- EnSight5 Measured/Particle Files (optional). The measured .res file references the measured geometry and variable files.

PLOT3D data files can be read as:

Workstation: ASCII, C Binary, or FORTRAN binary

Cray: ASCII, C Binary, or COS-Blocked FORTRAN binary

(see PLOT3D Reader, in Section 2.3)

Due to the different number of representations on a Cray Research vector system and workstations, binary files created on a Cray Research vector system can *not* be read on the workstation, and visa versa.

EnSight attempts to ensure that the format of the file being read matches the format you have selected in the Data Reader dialog. However, if you specify that the file is C binary, and it is really FORTRAN binary, this will not be detected and erroneous values will be loaded.

EnSight reads the geometry (xyz files) directly. However, an EnSight-like results file (described below) is needed in order to read the results, unless a “standard” Q-file is provided in its place.

PLOT3D Geometry file notes

The following information is required in order to read PLOT3D files correctly:

1. whether there is Iblanking information in the file
2. whether files are in ASCII, C Binary, or FORTRAN binary
3. whether the file is “Single Zone” or Multi-Zoned”
4. whether the model is 1D, 2D, or 3D in nature.

Iblanking can be one of the following:

0 = Outside (Blanked Out)

1 = Inside

2 = Interior boundaries

<0 = zone that neighbors

If single zone with Iblanking, you can build EnSight Parts from the inside portions, blanked-out portions, or internal boundary portions. If single zone, you can also specify I, J, K limiting ranges for Parts to be created.

If Multi-zoned with Iblanking, you can additionally build Parts that are the boundary between two zones. (For boundary you must specify exactly two zones.)

If Multi-zoned and not using the “between boundary” option, a Part can span several zones.

If Multi-zoned, the dimension of the problem is forced to be 3D.

There can be nodes in different zones which have the same coordinates. No attempt has been made to merge these. Thus, on shared zone boundaries, there will likely be nodes on top of nodes. One negative effect of this is that node labels will be on top of each other.

Currently EnSight only prints out the global conditions in the solution file, fsmach, alpha, re, and time. It does not do anything else with them.

Node and element numbers are assigned in a sequential manner. Queries can be made on these node and element numbers or on nodes by I, J, and K.

PLOT3D Result file format

The PLOT3D result file was defined by CEI and is very similar to the EnSight results file and contains information needed to relate variable names to variable files, step information, etc. There is a slight variation from the normal EnSight results file because of the differences between the solution (Q file) and function files. The difference lies on the lines which relate variable filenames to a description. These lines have the following format:

```
<filename> <type> <number(s)> <description>
```

See PLOT3D Result File below for the definition of each.

The following information is included in a PLOT3D result file:

- Number of scalar variables
- Number of vector variables
- Number of time steps
- Starting file number extension and skip-by value
- Flag that specifies whether there is changing geometry.
- Names of the files that contain the values of scalar and vector variables. An indication as to the type of the file being used for the variable, which variable in the file and the name given to that variable.
- The names of the geometry files that will be used for the changing geometry.

Generic PLOT3D Result File Format

The format of the Result file is as follows:

- Line 1

  Contains the number of scalar variables, the number of vector variables and a geometry changing flag. If the geometry changing flag is 0, the geometry of the model does not change over time. Only the coordinates can change for a PLOT3D file at present time.

- Line 2

  Indicates the number of time steps that are available.

Note: if this number is positive, then EnSight will assign the time values listed in Line 3 to the Analysis_Time constant variable. If this value is negative, the time values in Line 3 must be removed and EnSight will assign the TIME (or Tvref for OVERFLOW restart q-file(s)) value from the header of the q-file(s) to the Analysis_Time constant variable. This is noted under PLOT3D and/or OVERFLOW readers (see Section 2.3, Other Readers).

- Line 3

  Lists the time that is associated with each time step. There must be the same number of values as are indicated in Line 2. This “line” can actually span several lines in the file. (Exists if Line 2 is a positive, non-zero value).

- Line 4

  Specified only if more than one time step is indicated in Line 2. The two values on this line indicate the file extension value for the first time step and the offset between files. If the values on this line are 0 5, the first time step available has a subscript of 0, the second time step available has a subscript of 5, the third time step has a subscript of 10, and so on.

- Line 5

  This line exists only if the changing geometry flag on Line 1 has been set to 1. Line contains name of the PLOT3D xyz file. The file name must follow the EnSight wild card specification.

- Line 6 through Line [5+N] where N is the number of scalar variables specified in Line 1.

  List the file names that correspond to each scalar variable. There must be a file name for each scalar variable that is specified in Line 1. If there is more than one time step, the file name must follow the EnSight wild card specification.

  These lines also contain the type of file being used, solution or function, and the location of the variable value in the file. The contents are:

  ```
  <filename> <type> <number> <description>
  ```

  where filename is the name of solution file or function file containing the variable; type is “S” for solution file, or “F” for function file; numberis which variable in the file to use (specify just one number); and description is the Description of the variable.

  The solution file (“s”) is the traditional .q file in which normally the first variable is density, the second through fourth variables are the components of momentum, and the fifth variable is total energy.

- Lines that follow the scalar variable files.

  List the file names that correspond to each vector variable. There must be a file name for each vector variable that is specified in Line 1. If there is more than one time step, the file name must follow the EnSight wild card specification.

  These lines also contain the type of file being used, solution or function, and the location(s) of the variable values in the file. The contents are:

```
<filename> <type> <numbers> <description>
```

where filename is the name of solution file or function file containing the variable; type is “S” for solution file, or “F” for function file; numbersare which variables in the file to use (specify just three numbers); and description is the Description of the variable.

The generic format of the result file is as follows:

```
#_of_scalars #_of_vectors geom_chng_flag
#_of_timesteps
time1 time2 time3 .....
start_file_# skip_by_value
geometry_file_name.geo**
scalar0_file_name** type # description
scalar1_file_name** type # description
.
.
.
vector0_file_name** type # # # description
vector1_file_name** type # # # description
.
.
.
```

PLOT3D Example

This example illustrates a result file that specifies a non-changing geometry with only one time step.

```
3 2 0
1
0.0
block.sol S 1 Density
block.sol S 5 Total_Energy
block.scl F 1 Temperature
block.var F 1 2 3 Displacement
block.sol S 2 3 4 Momentum
```

Thus, this model will get two scalars from the solution file (block.sol). The first is Density in the first location in the file and the next is Total energy in the fifth location in the solution file. It will also get a Temperature scalar from the first location in the function file (block.scl).

It will get a Displacement vector from the function file called block.var. The three components of this vector are in the 1st, 2nd, and 3rd locations in the file. Finally, a Momentum vector will be obtained from the 2nd, 3rd, and 4th locations of the solution file.

Vectors can be 1D, 2D, or 3D. For a vector, always provide three numbers, but a zero will indicate that a component is empty, thus:

```
block.var F 1 0 3 XZ_Displacement
```

would be a 2D vector variable with components only in the X–Z plane.

If the above example had transient variables (but not geometry), with 3 time steps, it would appear as:

```
3 2 0
3
0.0 1.5 4.0
1 1
block.sol** S 1 Density
block.sol** S 5 Total_Energy
block.scl** F 1 Temperature
block.var** F 1 2 3 Displacement
block.sol** S 2 3 4 Momentum
```

The files needed would then be:

```
block.sol01    block.scl01    block.var01
block.sol02    block.scl02    block.var02
block.sol03    block.scl03    block.var03
```

And if the geometry changed as well as the variables, it would appear as:

```
3 2 1
3
0.0 1.5 4.0
1 1
block.geo**
block.sol** S 1 Density
block.sol** S 5 Total_Energy
block.scl** F 1 Temperature
block.var** F 1 2 3 Displacement
block.sol** S 2 3 4 Momentum
```

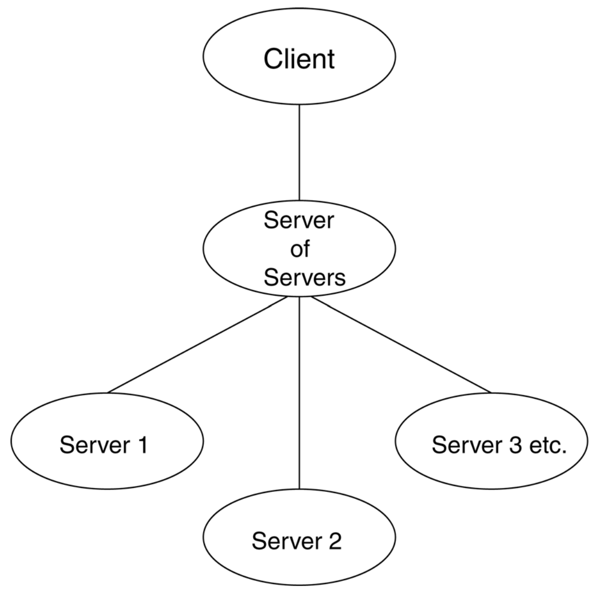
The files needed would then be:

```
block.sol01    block.scl01    block.var01    block.geo01
block.sol02    block.scl02    block.var02    block.geo02
block.sol03    block.scl03    block.var03    block.geo03
```

Note: A "standard" Q-file can be substituted for PLOT3D result file format if desired. A "standard" Q-file has 5 variable components (First is density, then the three components of momentum, and last is energy).

11.8 Server-of-Server Casefile Format

EnSight92 (with gold license key) has the capability of dealing with partitioned data in an efficient distributed manner by utilizing what we call a server-of-servers (SOS for short). An SOS server resides between a normal client and a number of normal servers. Thus, it appears as a normal server to the client, and as a normal client to the various normal servers.



This arrangement allows for distributed parallel processing of the various portions of a model, and has been shown to scale quite well.

Currently, EnSight SOS capability is only available for EnSight5, EnSight6, EnSight Gold, Plot3d, and any EnSight User-Defined Reader data. It is not directly available for Fidap Neutral, Fluent Universal, N3S, Estet, MPGS4, Movie, Ansys, Abaqus, or FAST Unstructured data.

Please recognize that your data must be partitioned in some manner (hopefully in a way that will be reasonably load balanced) in order for this approach to be useful. The exception to this is the use of the auto_distribute capability for structured or unstructured data. This option can be used if the data are readable by all servers. It will automatically distribute each portion of the data over the defined servers - without the user having to partition the data. ***Please note that currently only EnSight Gold, Plot3d, and any 2.06 (or greater) user-defined readers (which have implemented structured reader cinching) should be used for structured auto_distribute - and that only EnSight Gold and any 2.08 (or greater) user-defined readers (which have implemented the "*_in_buffers" routines) should be used for unstructured auto_distribute. Using an EnSight reader not written for SoS autodistribute will result in each server loading the entire model.***

Included in the EnSight distribution is an unsupported utility that will take most EnSight Gold binary unstructured datasets and partition it for you. The source for this utility (called "chopper") can be found in the `$CEI_HOME/ensight92/unsupported/partitioner` *directory.*

Note: If you do your own partitioning of data into EnSight6 or EnSight Gold format, please be aware that each part must be in each partition - but, any given part can be “empty” in any given partition. All that is required for an empty part is the “part” line, the part number, and the “description” line.

You should place each partitioned portion of the model on the computer that will access that portion. Each partitioned portion is actually a self contained set of EnSight data files, which could typically be read by a normal client - server session of EnSight. For example, if it were EnSight gold format, there will be a casefile and associated gold geometry and variable results file(s). On the machine where the EnSight SOS will be run, you will need to place the SOS casefile. This is a simple ascii file which informs the SOS about pertinent information need to run a server on each computer that will compute the various portions.

The format for the SOS casefile is as follows: (Note that [ ] indicates optional information, and a blank line or a line with # in the first column are comments.)

FORMAT (Required)
type: master_server *datatype* (Required)

where: *datatype* is required and is one of the formats of EnSight’s internal readers (which use the Part builder), namely:

gold ensight6 ensight5 plot3d

or it can be the string used to name any of the user-defined readers.
Note: For user-defined readers, the string must be exactly that which is defined in the USERD_get_name_of_reader routine of the reader (which is what is presented in the Format pulldown of the Data Reader dialog).
If *datatype* is blank, it will default to EnSight6 data type.

[auto_distribute: *server*/*reader*/*on*/*off*] (Optional for structured or unstructured data)

Set this option to “off” if you have moved the decomposed data file(s) into separate directories or machines. Each server will simply read all of the data as specified in its data_path and casefile below.

Set this option to “on” or “server” (these two options are the same) if you want the EnSight server to automatically distribute data to the servers specified below. This requires that each of the servers have access to the same data (or identical copies of it). ***For structured data: use the “ON” or “Server” option only if the datatype is gold, plot3d or a 2.06 or greater user-defined reader (which has implemented structured cinching). For unstructured data: use only if the datatype is gold, or a 2.* user-defined reader (which has implemented the “*_in_buffers” routines).***

Set this option to “reader” if you want EnSight to allow the readers to do the decomposition. This option requires that each of the servers have access to the same data (or identical copies of it). This is prefered over letting the server do the distribution. **However the “reader” option will only work if the reader has told EnSight that it can do the distribution by setting the return value to TRUE in the USERD_prefer_auto_distribute reader routine.**

[use_resources: on/off] (Optional, to use the specification of server machines from EnSight resources; this will override the machine names used in this file, see How To Use Resource Management.)

[plot3d_iblanked: *true*/*false*] (Required only if *datatype* is plot3d)

[plot3d_multi_zone: *true*/*false*] (Required only if *datatype* is plot3d)
[plot3d_dimension: *1d*/*2d*/*3d*] (Required only if *datatype* is plot3d)
[plot3d_source: *ascii*/*cbin*/*fortranbin*] (Required only if *datatype* is plot3d)
[plot3d_grid_double: *true*/*false*] (Required only if *datatype* is plot3d)
[plot3d_results_double: *true*/*false*] (Required only if *datatype* is plot3d)

where: iblanking, multi_zone, dimension, source type, grid file double precision, and results file double precision information should be provided. If it is not provided, it will default to the following (which is likely not to be correct):

plot3d_iblanked:	false
plot3d_muti_zone:	false
plot3d_dimension:	3d
plot3d_source:	cbin
plot3d_grid_double:	false
plot3d_results_double:	false

[do_ghosts: on/off] (Optional for unstructured auto_distribute - default is on)

Allows user to control whether ghost cells will be produced between the distributed portions.

[buffer_size: *n*] (Optional for unstructured auto_distribute and do_ghosts - default is 100000)

Allows user to modify the default buffer size that is used when reading node and element information of the model when producing ghost cells.

[want_metric: *on/off*] (Optional for unstructured auto_distribute and do_ghosts - default is on)

If set on, a simple metric will be printed in the shell window that can indicate the quality of the auto_distribution. The unstructured auto_distribute method relies on some coherence in the element connectivity - namely, that elements thal lie next to each other are generally listed close to each other in the data format.

The metric is simply the (#total_nodes / #nodes_needed_if_no_ghosts).

When no ghosts, the value will be 1.0. The more ghosts you must have, the higher this metric will be. If the number gets much more than 2.0, you may want to consider partitioning yourself.

NETWORK_INTERFACES (**Note: this whole section is optional**. It is needed only when more than one network interface to the sos host is available and it is desired to use them. Thus distributing the servers to sos communication over more than one network interface)

number of network interfaces: *num* (Required - if section used)

where: *num* is the number of network interfaces to be used for the sos host.

network interface: *sos_network_interface_name1*(Required - if section is used)
network interface: *sos_network_interface_name2*(Required - if section is used)
.
.
network interface: *sos_network_interface_namenum*(Required - if section used)

SERVERS (Required)
number of servers: *num* (Required)

where: *num* is the number of servers that will be started and run concurrently.

#Server 1	(Comment only)
machine id: *mid*	(Required)

where: *mid* is the machine id of the server.

executable: /.../ensight92.server	(Required, must use full path)
[directory: *wd*]	(Optional)

where: *wd* is the working directory from which ensight92.server will be run

[login id: *id*]	(Optional)

where: *id* is the login id. Only needed if it is different on this machine.

[data_path: /.../*dd*]	(Optional)

where: *dd* is the data where the data resides. Full path must be provided if you use this line.

casefile: *yourfile.case*	(Required, but depending on format, may vary as to whether it is a casefile, geometry file, neutral file, universal file, etc. Relates to the first data field of the Data Reader Dialog.)
[resfile: *yourfile.res*]	(Depends on format as to whether required or not. Relates to the second data field of the Data Reader Dialog.)
[measfile: *yourfile.mea*]	(Depends on format as to whether required or not. Relates to the third data field of the Data Reader Dialog.)
[bndfile: *yourfile.bnd*]	(Depends on format as to whether required or not. Relates to the fourth data field of the Data Reader Dialog.)
#Server 2	(Comment only)

--- Repeat pertinent lines for as many servers as declared to be in this file ---

Example

This example deals with a EnSight Gold dataset that has been partitioned into 3 portions, each running on a different machine. The machines are named joe, sally, and bill. The executables for all machines are located in similar locations, but the data is not. Note that the optional data_path line is used on two of the servers, but not the third.

```
FORMAT
type: master_server gold

SERVERS
number of servers:  3

#Server 1
machine id:  joe
executable:  /usr/local/bin/ensight92/bin/ensight92.server
data_path: /usr/people/john/data
casefile: portion_1.case

#Server 2
```

```
machine id:  sally
executable:  /usr/local/bin/ensight92/bin/ensight92.server
data_path: /scratch/sally/john/data
casefile: portion_2.case

#Server 3
machine id:  bill
executable:  /usr/local/bin/ensight92/bin/ensight92.server
casefile: /scratch/temp/john/portion_3.case
```

If we name this example sos casefile "all.sos", and we run it on yet another machine, one named george, the data would be distributed as follows:

On george: all.sos

On joe (in /usr/people/john/data): portion_1.case, and all files referenced by it.

On sally (in /scratch/sally/john/data): portion_2.case, and all files referenced by it.

On bill (in /scratch/temp/john): portion_3.case, and all file referenced by it.

By starting EnSight with the -sos command line option (which will autoconnect using ensight92.sos instead of ensight92.server), or by manually running ensight92.sos in place of ensight92.server, and providing all.sos as the casefile to read in the Data Reader dialog - EnSight will start three servers and compute the respective portions on them in parallel.

For examples of structured and unstructured auto_distribute, see:
How To Use SOS.

Spatially decomposed Case files

If you choose to read spatially decomposed case files using the same number of servers as case files, then this section is unnecessary. At some point you may encounter case files that are spatially decomposed and wish to read them with less servers than you have case files. At that point you will need to have this section with a keyword MULTIPLE_CASEFILES. There are several ways to specify the case files.

List them all by name:

```
MULTIPLE_CASEFILES
total number of cfiles:     n
cfiles global path:         global_path      Optional
cfiles:                     partition_1.case
                            partition_2.case
                            .
                            partition_n.case
```

Specify number, pattern, start and increment

```
MULTIPLE_CASEFILES
total number of cfiles:     n
cfiles global path:         global_path      Optional

cfiles pattern:             partition_*.case
cfiles start number:        #
cfiles increment:           #
```

Use a separate file to containing the case filenames

MULTIPLE_CASEFILES
total number of cfiles: n
cfiles file: all_together_cfilenames.txt

Please see **How To Load Spatially Decomposed Case Files**for detailed usage instructions.

Configuration using Environment Variables

By default, both the EnSight client and the server start up with an optimum number of threads (depending on the number of processors available and within the EnSight user license limits). Each executable of EnSight can be configured individually to limit the number of threads used. The following environment variables are used to specify the maximum number of threads that the executable should use for computation. These environment variables should be set in your shell startup script on the computers where the various EnSight executables run.

ENSIGHT9_MAX_THREADS

The maximum number of threads to use for each EnSight server. Threads are used to accelerate the computation of streamlines, clips, isosurfaces, some calculator functions and other compute-intensive operations.

ENSIGHT9_MAX_CTHREADS

The maximum number of threads to use for each EnSight client. Threads in the client are used to accelerate sorting of transparent surfaces.

ENSIGHT9_MAX_SOSTHREADS

The maximum number of threads to use on the server of server in order to start up server processes in parallel rather than serially.

Other Notes

The number of threads is limited to 2 (per client or server) with a Lite license, 4 (per client or server) with a Standard license, while the upper limit for a Gold license is 128. When setting these parameters it is a good idea to take into account the number of processors on the system. In general, you will not see benefit from setting the parameters higher than the number of total processors. Because the server, server-of-servers and client operate in a pipelined fashion, it is not necessary to limit one in order to apply more threads to another.

Compute intensive server operations that make use of shared memory parallel computations include isosurface, clipping, and particle trace computations. Client threaded operations include transparency resort and display list creation.

If the machine named george had more than one network interface (say it had its main one named george, but also had one named george2), we could add the section shown below to our casefile example:

```
NETWORK_INTERFACES
        number of network interfaces: 2
        network interface:  george
        network interface:  george2
```

This would cause machine joe to connect back to george, machine sally to connect back to george2, and machine bill to connect back to george. This is because the sos is cycling through its available interfaces as it connects the servers. Remember that this is an optional section, and most users will probably not use it. Also, the contents of this section will be ignored if the `-soshostname` command line option is used.

11.9 Periodic Matchfile Format

This is an optional file which can be used in conjunction with models which have rotational or translational computational symmetry (or periodic boundary conditions). It is invoked in the GEOMETRY section of the EnSight casefile, using the "match: filename" line. EnSight 6 Case file allows the use of the periodic matchfile which uses the matchfile in the border process to eliminate the matched symmetry faces at the shared boundary between the faces. EnSight Case Gold allows the optional [add_ghosts] parameter after the filename which will produce ghost cells across the match file boundary which will provide continuity for variable calculations across the boundary, and for computational symmetry/mirroring, etc. Only use the add_ghosts option if you can afford the penalty of the additonal ghost cells and you need the computational continuity that they provide.

When a model piece is created with periodic boundary conditions, there is usually a built-in correspondence between two faces of the model piece. If you transform a copy of the model piece properly, face 1 of the copy will be at the same location as face 2 of the original piece. It is desirable to know the corresponding nodes between face 1 and face 2 so border elements will not be produced at the matching faces. This correspondence of nodes can be provided in a periodic match file as indicated below. (Please note that if a periodic match file is not provided, by default EnSight will attempt to determine this correspondence using a float hashing scheme. This scheme has been shown to work quite well, but may not catch all duplicates. The user has some control over the "capture" accuracy of the hashing through the use of the command: "test: float_hash_digits". If this command is issued from the command dialog, the user can change the number of digits, in a normalized scheme, to consider in the float hashing. The lower the number of digits, the larger the "capture" distance, and thus the higher the number of digits, the smaller the capture distance. The default is 4, with practical limits between 2 and 7 or 8 in most cases.)

The transformation type and delta value are contained in the file. The periodic match file is an ASCII free format file. For unstructured data, it can be thought of as a series of node pairs, where a node pair is the node number of face 1 and its corresponding node number on face 2. For structured blocks, all that is needed is an indication of whether the i, j, or k planes contain the periodic face. The min plane of this "direction" will be treated as face 1, and the max plane will be treated as face 2.

The file format is as follows:

`rotate_x/y/z / translate`	The first line is either rotate_x, rotate_y, rotate_z or translate
`theta / dx dy dz`	The second line contains rotation angle in degrees or the three translational delta values.
np n_{11} n_{21} n_{12} n_{22} . . . . . . n_{1np} n_{2np}	If any unstructured pairs, the third line contains the number of these pairs (np). And the node ids of each pair follow. (The first subscript indicates face, the second is pair.)

`blocks` b_{min} b_{max} `i/j/k`	Last in the file comes as many of these “blocks” lines as needed. b_{min} and b_{max} are a range of block numbers. For a single block, b_{min} and b_{max} would be the same. Only one of i, j, or k can be specified for a given block.

Simple unstructured rotational example:

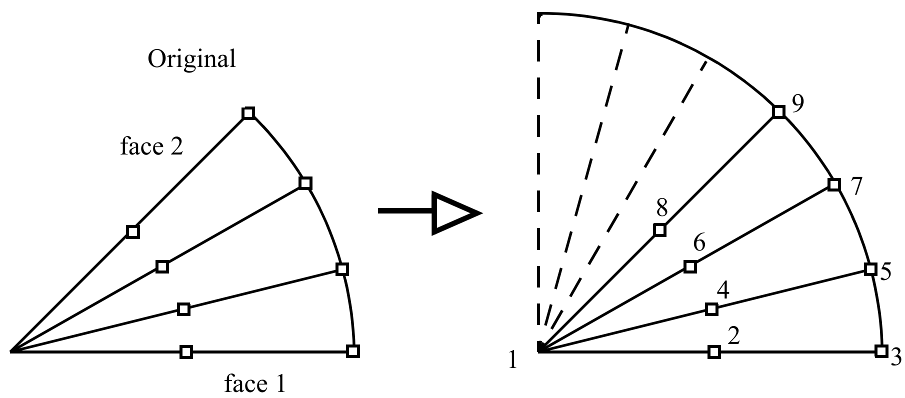


Figure 11-6
Model Duplication by rotational symmetry

The periodic match file for a rotation of this model about point 1 would be:

```
rotate_z
45.0
3
1  1
2  8
3  9
```

Thus, face 1 of this model is made up of nodes 1, 2, and 3 and face 2 of this model is made up of nodes 1, 8, and 9. So there are 3 node pairs to define, with node 1 corresponding to node 1 after a copy is rotated, node 2 corresponding to node 8, and node 3 corresponding to node 9.

Simple structured translational model:

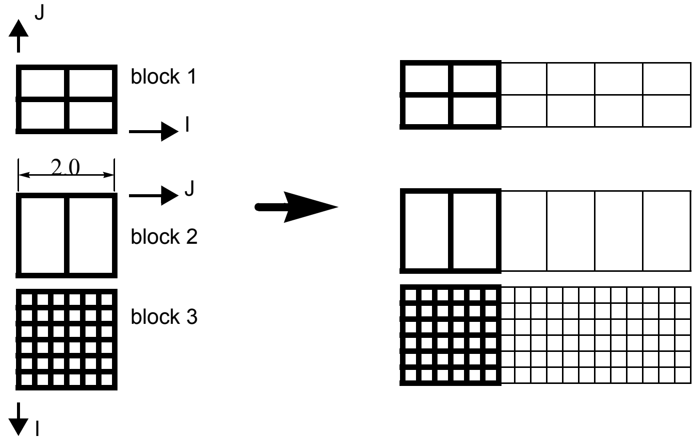


Figure 11-7
Model Duplication by translational symmetry
of structured blocks (3 instances)

```
translate
2.0 0.0 0.0
blocks 1 1 i
blocks 2 3 j
```

Since block 1 is oriented differently than blocks 2 and 3 in terms of ijk space, two "blocks" lines were needed in the match file.

Special Notes / Limitations:

1. This match file format requires that the unstructured node ids of the model be unique. This is only an issue with EnSight Gold unstructured format, since it is possible with that format to have non-unique node ids.

2. The model instance (which will be duplicated periodically) must have more than one element of thickness in the direction of the duplication. If it has only one element of thickness, intermediate instances will have all faces removed. If you have this unusual circumstance, you will need to turn off the shared border removal process, as explained in note 3.

3. The shared border removal process can be turned off, thereby saving some memory and processing time, by issuing the "test: rem_shared_bord_off" command in the command dialog. The effect of turning it off will be that border elements will be left between each periodic instance on future periodic updates.

4. The matching and hashing processes actually occur together. Thus, matching information does not have to be specified for all portions of a model. If no matching information is specified for a given node, the hashing process is used. By the same token, if matching information is provided, it is used explicitly as specified - even if it is has been specified incorrectly.

11.10XY Plot Data Format

This file is saved using the Save section of the Query Entity dialog. The file can contain one or more curves. The following is an example XY Data file:

Line	Contents of Line
1	2
2	Distance vs. Temperature for Line Tool
3	Distance
4	Temperature
5	1
6	5
7	0.0 4.4
8	1.0 5.8
9	2.0 3.6
10	3.0 4.6
11	4.0 4.8
12	Distance vs. Pressure for Line Tool
13	Distance
14	Pressure
15	2
16	4
17	0.00 1.2
18	0.02 1.1
19	0.04 1.15
20	0.06 1.22
21	3
22	1.10 1.30
23	1.12 1.28
24	1.14 1.25

Line 1 contains the (integer) number of curves in the file.

Line 2 contains the name of the curve.

Line 3 contains the name of the X-Axis.

Line 4 contains the name of the Y-Axis.

Line 5 contains the number of curve segments in this curve.

Line 6 contains the number of points in the curve segment.

Lines 7-11 contain the X-Y information.

Line 12 contains the name of the second curve.

Line 13 contains the name of the X-Axis

Line 14 contains the name of the Y-Axis

Line 15 contains the number of curve segments in this curve. (For the second curve, the first segment contains 4 points, the second 3 points.)

11.11 EnSight Boundary File Format

This file format can be used to represent boundary surfaces of structured data as unstructured parts. The boundaries defined in this file can come from sections of several different structured blocks. Thus, inherent in the file format is a grouping and naming of boundaries across multiple structured blocks.

Additionally, a delta can be applied to any boundary section to achieve the creation of repeating surfaces (such as blade rows in a jet engine).

Note: There is no requirement that the boundaries actually be on the surface of blocks, but they must define either 2D surfaces or 1D lines. You may not use this file to define 3D portions of the block.

The boundary file is read if referenced in the casefile of EnSight data models, or in the boundary field of the Data Reader Dialog for other data formats. Any boundaries successfully read will be listed in the Unstructured Data Part List of the Data Part Loader Dialog.

The format of the EnSight Boundary File is as follows:

Line (1): Required header keyword and version number. ENSBND is required exactly, but the version number could change in the future.

```
ENSBND 1.00
```

Line (2) through Line (NumBoundaries+1): The names of the boundaries to be defined. (Each name must be no greater than 79 characters.)

For example:

```
inflow
wall
Chimera Boundary
outflow
```

Line (NumBoundaries+2): Required keyword indicating the end of the boundary names and the beginning of the boundary section definitions.

```
BOUNDARIES
```

Line (NumBoundaries+3) through the end of the file: The boundary section definitions. Each line will have the following information:

```
bnd_num blk_num imin imax jmin jmax kmin kmax [di dj dk n_ins]
```

where:

bnd_num is the number of the boundary in the list of names above.
(For example: inflow is 1, wall is 2, Chimera Boundary is 3, etc.)

blk_num is the parent structured block for this boundary section.

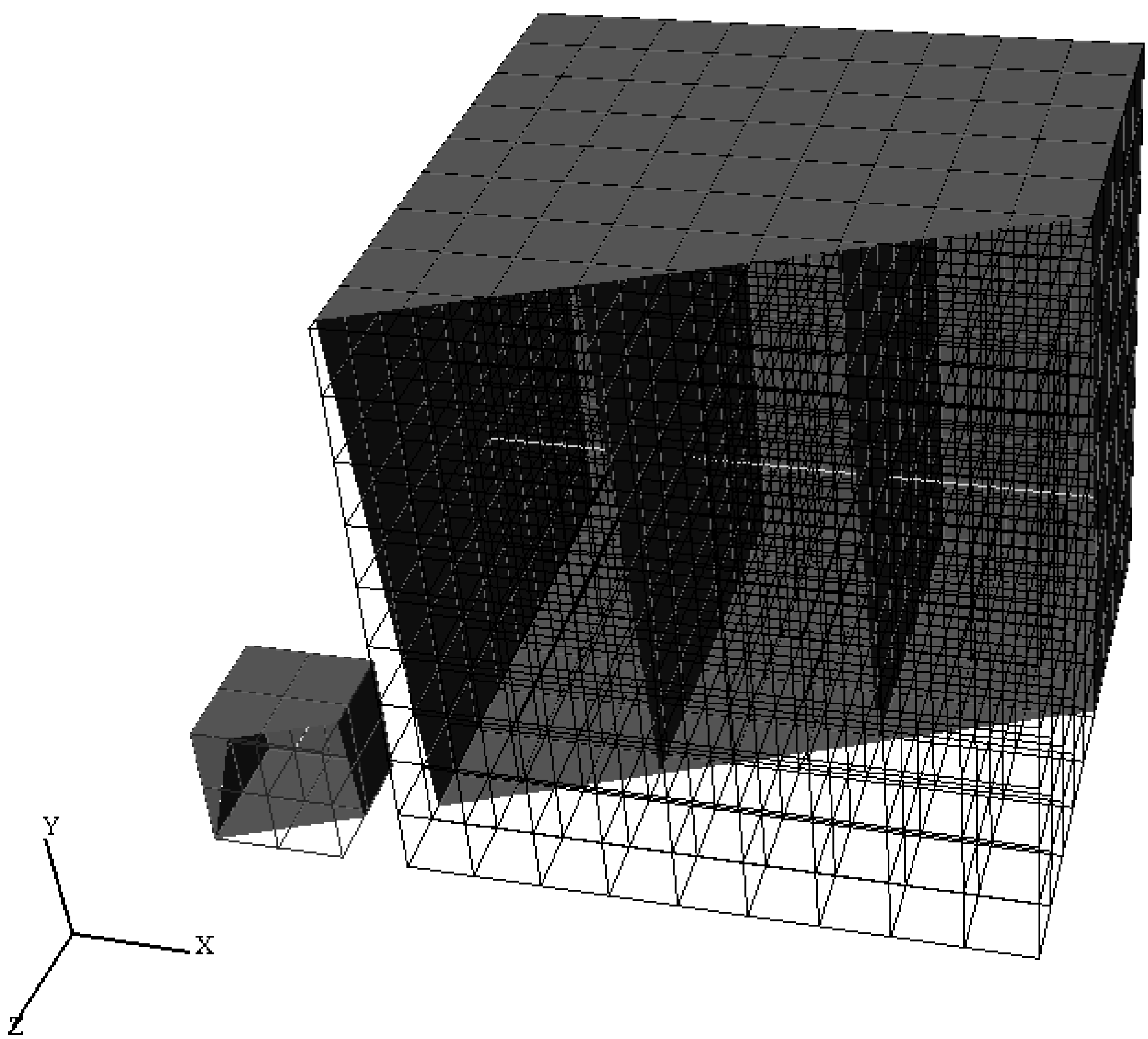
imin,imax, jmin,jmax, kmin,kmax are the ijk ranges for the boundary section. At least one of the min,max pairs must refer to the same plane. A wildcard ("$") can be used to indicate the maximum i, j, or k value of that block (the far plane). Additionally, negative numbers can be used to indicate plane values from the far side toward the near side. (-1 = far plane, -2 = one less than the far plane, etc.)

[di,dj,dk and **n_ins]** are optional delta information which can be used to extract repeating planes. The appropriate di, dj, or dk delta value should be set to the repeating plane offset value, and the other two delta values must be zero. The non-zero delta must correspond to a min,max pair that are equal. The n_ins value is used to indicate the number of repeating instances desired. A (“$”) wildcard can be used here to indicate that the maximum number of instances that fit in the block should be extracted.

All numbers on the line must be separated by spaces.

Finally, comment lines are allowed in the file by placing a “#” in the first column.

Below is a simple example of a boundary file for two structured blocks, the first of which is a 3 x 3 x 3 block, and the second is a 10 x 10 x 10 block. We will define a boundary which is the front and back planes of each block(K planes) and one that is the top and bottom planes of each block(J planes). We will also define some repeating x planes, namely, planes at i=1 and 3 for block one and at i=1, 4, 7, and 10 for block two. The image below shows the blocks in wire frame and the boundaries shaded and slightly cut away, so you can see the interior x-planes.



The file to accomplish this looks like:

```
ENSBND 1.00
front_back
top_bottom
x-planes
middle lines
BOUNDARIES
#bnd blk imin imax jmin jmax kmin kmax di dj dk n_ins
#--- --- ---- ---- ---- ---- ---- ---- -- -- -- -----
 1   1   1    3    1    3    $    $
 1   2   1    $    1    $    10   10
 2   1   1    $    $    $    1    $
 2   2   1    10   10   10   1    10
 1   1   1    3    1    3    1    1
 1   2   1    $    1    $    1    1
 2   1   1    3    1    1    1    $
 2   2   1    $    1    1    1    $
 3   1   1    1    1    $    1    $    2  0  0  2
 3   2   1    1    1    $    1    $    3  0  0  $
 4   1   2    2    2    2    1    3
 4   2   1    10   5    5    5    5
```

Interpreting the 12 boundary definition lines:

```
 1   1   1    3    1    3    $    $
```
defines a part of the boundary called front_back, on block 1, where I=1 to 3, J=1 to 3, and K=3. Thus, the far K plane of block 1.

```
 1   2   1    $    1    $    10   10
```
defines another part of the front_back boundary, on block 2, where I=1 to 10, J=1 to 10, and K = 10. Thus, the far K plane of block 2.

```
 2   1   1    $    $    $    1    $
```
defines a part of the boundary called top_bottom, on block 1, where I=1 to 3, J=3, and K=1 to 3. Thus, the far J plane of block 1.

```
 2   2   1    10   10   10   1    10
```
defines another part of the top_bottom boundary, on block 2, where I=1 to 10, J=10, and K=1 to 10. Thus, the far J plane of block 2.

```
 1   1   1    3    1    3    1    1
```
defines another part of the front_back boundary, on block 1, where I=1 to 3, J=1 to 3, and K=1. Thus, the near K plane of block 1.

```
 1   2   1    $    1    $    1    1
```
defines another part of the front_back boundary, on block 2, where I=1 to 10, J=1 to 10, and K=1. Thus the near K plane of block 2.

```
 2   1   1    3    1    1    1    $
```
defines another part of the top_bottom boundary, on block 1, where I=1 to 3, J=1, and K=1 to 3. Thus, the near J plane of block 1.

```
 2   2   1    $    1    1    1    $
```
defines another part of the top_bottom boundary, on block 2, where I=1 to 10, J=1,and K=1 to 10. Thus, the near J plane of block 2.

```
 3    1    1     1     1     $     1     $     2   0   0   2
```
defines a part of the boundary called x-planes, on block 1, where I=1, J=1 to 3, K=1 to 3, then again where I=3, J=1 to 3, and K=1 to 3. Thus, both the near and far I planes of block 1.

```
 3    2    1     1     1     $     1     $     3   0   0   $
```
defines another part of the x-planes boundary, on block 2, where I=1, J=1 to 10, and K=1 to 10, then again where I=4, J=1 to 10, and K=1 to 10, then again where I=7, J=1 to 10, and K=1 to 10, then again where I=10, J=1 to 10, and K=1 to 10. Thus, the I = 1, 4, 7, and 10 planes of block 2.

```
 4    1    2     2     2     2     1     3
```
defines a part of the boundary called middle lines, on block 1, where I=2, J=2, and K=1 to 3. Thus, line through the middle of block 1 in the K direction.

```
 4    2    1     10    5     5     5     5
```
defines another part of the middle lines boundary, on block 2, where I=1 to 10, J=5, and K=5. Thus a line through the middle of the block2 in the I direction.

Please note that the "$" wildcard was used rather randomly in the example, simply to illustrate how and where it can be used.

The use of negative numbers for ijk planes is indicated below - again rather randomly for demonstration purposes. This file will actually produce the same result as the file above.

```
ENSBND 1.00
front_back
top_bottom
x-planes
middle lines
BOUNDARIES
#bnd blk imin imax jmin jmax kmin kmax di dj dk n_ins
#--- --- ---- ---- ---- ---- ---- ---- -- -- -- -----
 1    1   1     3    1     3   -1   -1
 1    2  -3    -1    1    -1    10   10
 2    1   1    -1   -1    -1    1   -1
 2    2   1     10   10    10   1    10
 1    1   1     3    1     3    1    1
 1    2   1    -1    1    -1    1    1
 2    1   1     3    1     1    1   -1
 2    2   1    -1    1     1    1   -1
 3    1   1     1    1    -1    1   -1    2  0  0  2
 3    2   1     1    1    -1    1   -1    3  0  0  $
 4    1   2    -2   -2     2    1    3
 4    2   1     10   5     5   -6   -6
```

11.12 EnSight Particle Emitter File Format

This file can be used to specify the location of emitter points. It is most useful when the user has specific (and many) emit points to use for particle traces. Rather than type them all in one at a time as a cursor emitter, this file can be used.

Note the following:

1. The first two lines need to be exactly as shown.

```
Version 1.0
EnSight Particle Emitter File
```

2. #'s as the first character on a line, are comment lines. Comment lines can be used anywhere in the file after the first two mandatory lines.

3. `Time` lines contain the emitter release time (simulation time, NOT time step).

4. `Emit` lines contain the coordinates of an emitter. The emitter will be associated with the previously specified `Time`.

Sample File:

```
Version 1.0
EnSight Particle Emitter File
#
Time 0.249063
Emit 0.0669737 0.0134195 -0.013926
Emit 0.0669737 0.0131277 -0.0141079
Emit 0.0669737 0.0128642 -0.0143178
Emit 0.0669737 0.0155969 -0.0113572
Emit 0.0669737 0.0150344 -0.0122012
Emit 0.0669737 0.0146967 -0.0123587
#
Time 0.249113
Emit 0.0669737 0.0137097 -0.0136404
Emit 0.0669737 0.0134218 -0.0138284
Emit 0.0669737 0.0131628 -0.0140438
Emit 0.0669737 0.0158325 -0.0110264
Emit 0.0669737 0.0152879 -0.011882
Emit 0.0669737 0.0149536 -0.0120466
#
Time 0.249163
Emit 0.0669737 0.0139938 -0.0133488
Emit 0.0669737 0.01371 -0.0135428
Emit 0.0669737 0.0134555 -0.0137636
Emit 0.0669737 0.0160612 -0.0106906
Emit 0.0669737 0.0155347 -0.0115575
Emit 0.0669737 0.0152039 -0.0117291
#
Time 0.249213
Emit 0.0669737 0.0142718 -0.0130512
Emit 0.0669737 0.013992 -0.0132511
Emit 0.0669737 0.0137423 -0.0134773
Emit 0.0669737 0.0162827 -0.0103501
Emit 0.0669737 0.0157745 -0.0112279
Emit 0.0669737 0.0154475 -0.0114064
#
```

```
Time 0.249263
Emit 0.0669737 0.0145433 -0.0127479
Emit 0.0669737 0.0142679 -0.0129536
Emit 0.0669737 0.014023 -0.013185
Emit 0.0669737 0.0164969 -0.0100051
Emit 0.0669737 0.0160074 -0.0108933
Emit 0.0669737 0.0156842 -0.0110787
```

11.13 EnSight Rigid Body File Format

This file can be used to specify the rigid body Euler Parameter file (see Section 11.14, Euler Parameter File Format) and which transformations within said file to associate with the parts in EnSight Gold format. It can also be used to specify a units scaling factor, center of gravity offsets, and initial yaw pitch roll rotations, if needed. Please note that the order of transformations will be first) the yaw pitch roll rotations, if any specified, then second) the center of gravity offsets, if any specified, and third) the euler parameter rotations and translations of the euler parameter file. While not required, `.erb` is the normal extension given to this file. It is referenced via the `rigid_body` line in an EnSight Gold Casefile.

Note the following:

1. The first line needs to be exactly as shown.

   `EnSight Rigid Body`

2. The second line must be the .erb version number line. (Version is 1.1, previous to EnSight 8.2 this line was not present and thus defaulted to version 1.0)

   `version 1.1`

2. The third line must either be exactly

   `names`

   or

   `numbers`

   This indicates whether part names or part numbers will be used in the association of ensight parts to the rigid body information in the Euler Parameter File.

3. The fourth line must be a single integer `n` indicating the number of lines that follow in the file.

4. The remaining `n` lines contain either 3, 4, 7, or 11 tokens which must each be enclosed in quotes. When no unit scaling or center of gravity offset is needed the lines have the form of either

   `"ens_part_name" "eet_filename" "eet_title"`

   or

   `"ens_part_#(s)" "eet_filename" "eet_title"`

   depending on the state of the second line in the file.

   If unit scaling is specified, then the lines have the form of either

   `"ens_part_name" "eet_filename" "eet_title" "u_scale"`

   or

   `"ens_part_#(s)" "eet_filename" "eet_title" "u_scale"`

   If center of gravity offsets are needed (without rotations), the lines are either

   `"ens_part_name" "eet_filename" "eet_title" "u_scale" "xoff" "yoff" "zoff"`

   or

   `"ens_part_#(s)" "eet_filename" "eet_title" "u_scale" "xoff" "yoff" "zoff"`

If yaw pitch roll rotations are needed (without offsets), the lines are either

```
"ens_part_name" "eet_filename" "eet_title" "u_scale" "rot_order" "xrot" "yrot" "zrot"
```

or

```
"ens_part_#(s)" "eet_filename" "eet_title" "u_scale" "rot_order" "xrot" "yrot" "zrot"
```

If both offsets and initial yaw pitch roll rotations are needed, the lines are either (note difference in order of 3 offset offset columns and 4 rotation columns).

```
"ens_part_name" "eet_filename" "eet_title" "u_scale" "xoff" "yoff" "zoff" "rot_order" "xrot" "yrot" "zrot"
```

or

```
"ens_part_name" "eet_filename" "eet_title" "u_scale" "rot_order" "xrot" "yrot" "zrot" "xoff" "yoff" "zoff"
```

or

```
"ens_part_#(s)" "eet_filename" "eet_title" "u_scale" "xoff" "yoff" "zoff" "rot_order" "xrot" "yrot" "zrot"
```

or

```
"ens_part_#(s)" "eet_filename" "eet_title" "u_scale" "rot_order" "xrot" "yrot" "zrot" "xoff" "yoff" "zoff"
```

Where:

`ens_part_name` = the actual part description given in the EnSight geometry file

`ens_part_#(s)` = a single number, comma separated list, or list of dash separated ranges of part numbers given in the EnSight geometry file

`eet_filename` = the actual name of the Euler Parameter file

`eet_title` = the actual name of the title in the columns of the Euler Parameter File which correspond to the rigid body euler values to apply to the ensight parts indicated

`u_scale` = the optional units scaling value which can be applied to the translation values in the Euler Parameter file. This is only needed if the translations of the Euler Parameter file are not in the same units as the EnSight geometry. (Note, if you use the xoff, yoff, zoff offset below, you must use this value as well - but it can be 1.0) Note: this does not scale the geometry. This is used to scale the translational values in the .eet file only. For example if your .eet file has units of millimeters but your model is in meters, putting a 1000 here will scale up all your translations appropriately. This does not scale values in the .erb file.

`xoff,yoff,zoff` = optional additional offsets to the center of gravity that can be applied. These are applied before the rotation and translations contained in the Euler Parameters file, and as such are generally not needed

`rot_order` = one of the following, indicating the order in which to apply the rotation angles. xyz yxz zxy xzy yzx zyx

`xrot,yrot,zrot` = rotation angles (in degrees) about each axis. These will be applied in the order specified with rot_order.

Simple Sample File (using part numbers):

```
EnSight Rigid Body
version 1.1
```

```
numbers
3
“1,5”        “my.eet” “BUCKET”
“2-4,6-9,11” “my.eet” “BOOM”
“10”         “my.eet” “LINK”
```

Note, the above file would require that the ensight geometry file contain parts 1 thru 11 and that the my.eet file contain titles BUCKET, BOOM, and LINK.

Sample File (using part names and optional scaling and offsets - no initial rotations):

```
EnSight Rigid Body
version 1.1
names
11
“bucket_property_1” “my.eet” “BUCKET” “1000.0” “383.7” “105.23” “0.0”
“bucket_property_2” “my.eet” “BUCKET” “1000.0” “383.7” “105.23” “0.0”
“boom_property_1”   “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“boom_property_2”   “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“boom_property_3”   “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“boom_enda”         “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“boom_endb”         “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“boom_top”          “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“boom_bot”          “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“boom_cen”          “my.eet” “BOOM”   “1000.0” “-35.4” “76.85”  “45.6”
“link”              “my.eet” “LINK”   “1000.0” “102.4” “80.0”   “0.0”
```

Sample File (using part names and optional scaling and initial rotations - no offsets):

```
EnSight Rigid Body
version 1.1
names
11
“bucket_property_1” “my.eet” “BUCKET” “1000.0” “yzx” “0.0” “90.0” “0.0”
“bucket_property_2” “my.eet” “BUCKET” “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_property_1”   “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_property_2”   “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_property_3”   “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_enda”         “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_endb”         “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_top”          “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_bot”          “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“boom_cen”          “my.eet” “BOOM”   “1000.0” “yzx” “0.0” “90.0” “0.0”
“link”              “my.eet” “LINK”   “1000.0” “yzx” “0.0” “90.0” “0.0”
```

Note, the two files above would require that the ensight geometry file contain parts labeled bucket_property_1, bucket_property_2, boom_property_1, boom_property_2, boom_property_3, boom_enda, boom_endb, boom_top, boom_bot, boom_cen, and link, and that the my.eet file contain titles BUCKET, BOOM, and LINK.

Sample File (using part numbers and optional scaling, offsets and rotations - with the offsets before the rotations. Also has some comments in the file):

```
# This is the same file as above, but has comments in it
EnSight Rigid Body
version 1.1
numbers
```

```
11
# Here come the parts registered against the labels in the motion file and
# assigned initial rotations and cg offsets
"1"  "my.eet" "BUCKET" "1000.0" "383.7" "105.23" "0.0"  "yzx" "0.0" "90.0" "0.0"
"2"  "my.eet" "BUCKET" "1000.0" "383.7" "105.23" "0.0"  "yzx" "0.0" "90.0" "0.0"
"3"  "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"4"  "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"5"  "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"6"  "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"7"  "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"8"  "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"9"  "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"10" "my.eet" "BOOM"   "1000.0" "-35.4" "76.85"  "45.6" "yzx" "0.0" "90.0" "0.0"
"11" "my.eet" "LINK"   "1000.0" "102.4" "80.0"   "0.0"  "yzx" "0.0" "90.0" "0.0"
```

Sample File (using part numbers and optional scaling, offsets and rotations - with the rotations before the offsets. Also has some comments in the file):

```
EnSight Rigid Body
version 1.1
numbers
11
"1"  "my.eet" "BUCKET" "1000.0" "yzx" "0.0" "90.0" "0.0" "383.7" "105.23" "0.0"
"2"  "my.eet" "BUCKET" "1000.0" "yzx" "0.0" "90.0" "0.0" "383.7" "105.23" "0.0"
"3"  "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"4"  "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"5"  "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"6"  "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"7"  "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"8"  "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"9"  "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"10" "my.eet" "BOOM"   "1000.0" "yzx" "0.0" "90.0" "0.0""-35.4" "76.85"  "45.6"
"11" "my.eet" "LINK"   "1000.0" "yzx" "0.0" "90.0" "0.0""102.4" "80.0"   "0.0"
```

Note, the two files above would require that the ensight geometry file contain 11 parts and that the my.eet file contain titles BUCKET, BOOM, and LINK.

11.14 Euler Parameter File Format

This contains rigid body transformation values (center of gravity translations and euler parameters) over time. Please note that these transformations are applied after any yaw pitch roll rotations and/or any center of gravity offsets specified in the EnSight Rigid Body file. The order of application of these transformations is first) the euler parameter rotations, then second) the translations. While not required, `.eet`, is the normal extension given to this file. It is referenced for EnSight parts via the EnSight Rigid Body file. It can also be referenced from readers. The Nastran reader .mop file, and the STL reader .xct are examples of this.

For a concise description of euler parameters, see the following, some of which is discussed below.

Eric W. Weisstein. "Euler Parameter." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/EulerParameters.html

Note the following:

1. The first line needs to be exactly as shown.

   `Ens_Euler`

2. The second section consists of two lines. The first of which must be exactly:

   `NumTimes:`

   and the second of which must contain one integer indicating the number of times, `nt`, contained in the file.

3. The third section consists of two lines. The first of which must be exactly:

   `NumTrans:`

   and the second of which must contain one integer indicating the number of transformations, `ntx`, contained in the file.

4. The fourth section must begin with a line that is exactly:

   `Titles:`

   and must have `ntx` number of lines - each containing the title associated with a transformation. These are the titles that .erb, .mob, and .xct files reference as they associate parts with rigid body transformations.

5. The rest of the file consists of `nt` time step sections with the first line being exactly:

   `Time Step:`

   the next line containing only a single float which represents the time value, and the next `ntx` lines containing 7 floats representing the 3 translations in x, y, z and the 4 euler parameters, e_o, e_1, e_2, and e_3 which describes a finite rotation angle of ϕ as follows:

$$e_0 = \cos\left(\frac{\phi}{2}\right)$$

and a normalized, scaled unit vector, *n*, about which the rotation occurs.

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \hat{n}\sin\left(\frac{\phi}{2}\right)$$

and e_o, e_1, e_2, and e_3 form a quaternion in scalar-vector representation

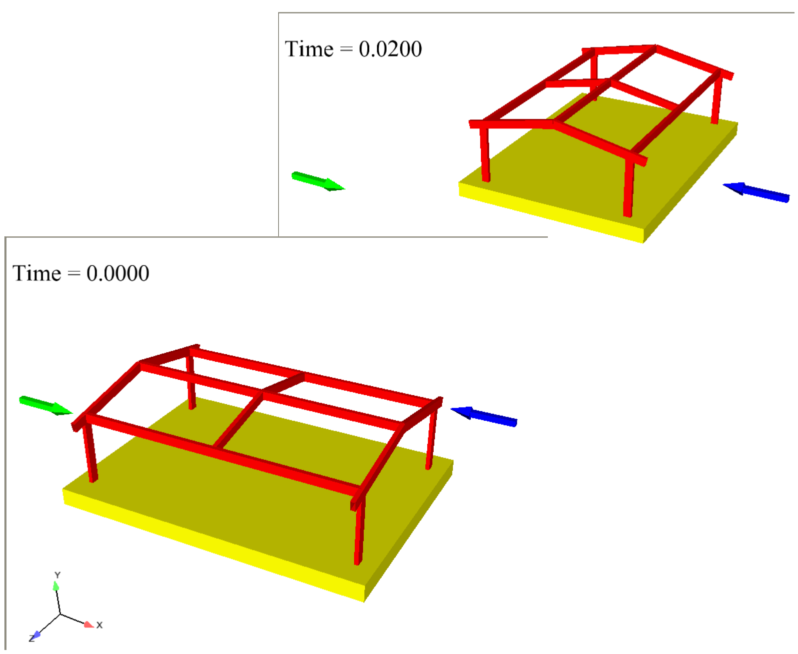
$$(e_0,e) = e_o^2 + e_1^2\hat{i} + e_3^2\hat{j} + e_3^2\hat{k}$$

and, since, Euler's rotation theorem states that an arbitrary rotation may be described by only three parameters, the four quantities must be related

$$e_o^2 + e \bullet e = e_o^2 + e_1^2 + e_3^2 + e_3^2 = 1$$

Note: If your four euler parameters don't satisfy these equations you may get odd non-linear rotational behavior and scaling of your geometry.

Suppose we want to rotate frame and base in the frame dataset (included in your EnSight install at $CEI_HOME/ensight92/data/frame) about the Y-axis in several timesteps. The frame dataset is static, that is, it has no timesteps. But, using the rigid body rotation files, we will make the parts transient in time.



The frame case file is modified by adding one line after the model line which indicates the EnSight rigid body (.erb) filename.

```
model:                  frame.geo
rigid_body: motion.erb
```

The motion.erb file might look as follows:

```
EnSight Rigid Body
version 1.1
names
2
"3d space frame" "simple_euler_y.eet" "Frame" "1.0" "0.0" "0.0" "0.0" "xyz" "0.0" "0.0" "0.0"
"frame base" "simple_euler_y.eet" "Frame" "1.0" "0.0" "0.0" "0.0" "xyz" "0.0" "0.0" "0.0"
```

This erb file names the euler (.eet) file: simple_euler_y.eet which is as follows:

```
Ens_Euler
NumTimes:
5
NumTrans:
1
Titles:
Frame
Time Step:
 0.0
 0.0 0.0 0.0 1.0 0.0 0.0 0.0
Time Step:
 0.01
 0.0 0.0 0.0 0.92388 0.0 0.38268 0.0
Time Step:
 0.02
 0.0 0.0 0.0 0.707 0.0  0.707 0.0
Time Step:
 0.03
 0.0 0.0 0.0 0.3827 0.0  0.92390 0.0
Time Step:
 0.04
 0.0 0.0 0.0 0.0 0.0 1.0 0.0
```

Notice that the Euler parameters are for 5 rotations about the Y-axis, 0, 45, 90, 135, and 180 degrees. The four euler parameters are

$\cos(\phi / 2)$, 0.0, $1.0*\sin(\phi / 2)$, 0.0

In this simple example we now have changed a static dataset into a transient dataset with 5 timesteps. This is useful for showing the translational and rotational motion of a rigid body. This is also useful for steady-state flows that include one solution that may have a rotational part(s) that you wish to animate over time. You can create an euler file that accurately rotates the part over time as well as the flow field around it, and then trace pathlines through this rotating flowfield, because the velocity and other vector variables are rotated along with the geometry.

Another, more complex sample file might look like the following:

```
Ens_Euler
NumTimes:
  6
NumTrans:
  3
Titles:
  Boom
  Bucket
  Link
```

```
Time Step:
  0.0000
  1600.5009  852.6449 -444.21389 0.9920 -0.0065 -0.05263 0.1144
  3111.1418 -355.9743 -282.1282  0.0443 -0.5412  0.8391 -0.0291
 -1463.1949  765.1186   -0.7573  0.9999 -0.0001 -0.0001  0.0035
Time Step:
  0.1900
  1600.6779 852.2065 -444.19329 0.9920 -0.0065 -0.05263 0.1143
  3093.7031 -378.0978 -284.0403 0.0454 -0.5085  0.8594 -0.0273
 -1463.3661  765.2241   -0.7464 0.9999 -0.0001 -0.0001  0.0035
Time Step:
  0.3900
  1600.8939 852.2266 -444.23309 0.9920 -0.0065 -0.05268 0.1137
  3065.6582 -404.1990 -286.7536 0.0471 -0.4527  0.8900 -0.0245
 -1462.6159  765.4865   -1.4728 0.9999 -0.0002 -0.0003  0.0034
Time Step:
  0.5900
  1621.0400 808.7615 -446.20608 0.9947 -0.0054 -0.05139 0.0881
  3053.7280 -510.7401 -292.5969 0.0464 -0.4105  0.9103 -0.0230
 -1463.4859  765.3441   -3.0230 0.9999 -0.0003 -0.0002  0.0034
Time Step:
  0.7901
  1640.6300 771.1768 -500.67977 0.9974 -0.0017 -0.02703 0.0659
  3135.5610 -562.3977 -420.8722 0.0245 -0.4325  0.9012 -0.0112
 -1464.9549  764.2155   -0.6192 0.9999 -0.0003 -0.0001  0.0037
Time Step:
  0.9900
  1654.2860 736.9480 -533.92132 0.9988 -0.0004 -0.01302 0.0461
  3217.4958 -596.2341 -493.7609 0.0117 -0.4585  0.8885 -0.0057
 -1464.9670  761.1839   -1.0681 0.9999 -0.0010 -0.0001  0.0051
```

11.15 Vector Glyph File Format

This file contains vector glyph information. Vector glyphs can represent static or transient forces or moments. These vectors can be located at a node id, an element id, or an x,y,z location. Transient vector glyphs can change value and/or location over time.

This file format is referenced from and EnSight Casefile, with a line in the `GEOMETRY` section, as such:

```
vector_glyphs:  filename.vgf
```

See EnSight Gold Case File Format for more detail on the EnSight Casefile.

General Comments:

1. This is an ascii file.

2. Comment lines are allowed in the file. To be a comment, the line must begin with a # sign, or be a blank line.

3. Each non-comment line begins with a keyword. When multiple lines are needed for a given keyword entry, the keyword is repeated on each line.

4. The various tokens on a given line must be separated by spaces. Do NOT allow the number of space separated tokens to exceed 15 on a line. *Note: this restriction really only applys to the TIMELINE times: line. All others have a set number of tokens per line.*

5. Transient timelines are specified separately, so they can easily be applied to multiple vector glyphs. Static vector glyphs do not reference a timeline.

File description:

1. The first non-comment line must be exactly:		`EnSight Vector Glyphs`
2. The second line must be the version number, like: Currently the only valid version is 1.0		`Version 1.0`
3. The third line must be the number of Vector Glyphs in the file, like: where `int` is the number of vector glyphs		`NumVectorGlyphs: int`
4. The fourth line should be the number of Transient timelines in the file, if any. The line should look like: where `int` is the number of vector glyph timelines. (If all vector glyphs are static, the line is not needed, or should indicate that the number is 0).		`NumTimeLines: int`
5. Each vector glyph will start with a line containing the single capitalized word, `VECTOR`. Under each `VECTOR` section, several lines are needed/ possible:		`VECTOR`
	This is a required 2 token line, where `int` is an id number. (Note: this must be first line of the section.)	`id: int`
	This is a required 2 (or more) token line.	`description: string`
	This is a required 2 token line. `constant` must be `FORCE` or `MOMENT`	`type: constant`
	This is a required 2 token line. `constant` must be `STATIC` or `TRANSIENT`	`time_condition: constant`
	This is required if `TRANSIENT` time_condition. It is a 2 token line. `int` will be the id of the associated timeline. (Note: the `time_line` must be specified after the `time_condition`, and before any `xyzloc` or `values` lines)	`time_line: int`
	This is a required 2 token line. `int` is the associated part number	`part: int`
	Use this 2 token line if attaching to a node id. `int` is the node id to use.	`nidloc: int`
	Use this 2 token line if attaching to an element id. `int` is the element id to use.	`eidloc: int`
	Use this 4 token line(s) if specifying an xyz location to act at. The three floats are the x,y,z locs. (There must be one of these lines for each time. Thus, for a STATIC glyph there will be just one line, but for a TRANSIENT glyph, there will be multiple - namely the `numtimes` of the associated timeline.)	`xyzloc: float float float`
	(Note: one of the three location methods must be specified)	

	This is a required 4 token line(s). The three floats are the x,y,z components of the vector. (There must be one of these lines for each time. Thus, for a STATIC glyph there will be just one line, but for a TRANSIENT glyph, there will be multiple - namely the `numtimes` of the associated timeline.)	`values: float float float`
6. Each vector glyph timeline (if any) will start with a line containing the single capitalized word, `TIMELINE`. Under each `TIMELINE` section, several lines are needed/ possible:.		`TIMELINE`
	This is a required 2 token line. `int` is an id number, and is the number that will be referenced by the `VECTOR time_line` *(Note: Must be first line of the section.)*	`id: int`
	This is a required 2 token line. `int` is the number of time steps in the timeline. *(Note: Must follow the id line, and precede the times line.)*	`numtimes: int`
	This is a required multi-token line. The `floats` are the time values. We must read in `numtimes` floats, however the number of these on a given line is somewhat arbitrary. Just make sure that you do not do more than 15 times per line.	`times: float float ...`
	This is an optional 2 token line. `constant` is `UNDEF` or `NEAREST`. This controls what will happen if EnSight specifies a time before the first time in this timeline. `UNDEF` will make the vector undefined. `NEAREST` will treat it as if it is at the first time.	`before: constant`
	This is an optional 2 token line. `constant` is `NEAREST` or `INTERPOLATE`. This controls what will happen if EnSight specifies a time between times in this timeline. `NEAREST` will cause the nearest time to be used. `INTERPOLATE` will interpolate between the bounding times.	`amidst: constant`
	This is an optional 2 token line. `constant` is `UNDEF` or `NEAREST`. This controls what will happen if EnSight specifies a time after the last time in this timeline. `UNDEF` will make the vector undefined. `NEAREST` will treat it as if it is at the last time.	`after: constant`

	(Note: The defaults for `before`, `amidst`, *and* `after` *are:*
	`before: UNDEF`
	`amidst: INTERPOLATE`
	`after: UNDEF`)	

Example:

The following fictitious example shows a few variations for 4 vector glyphs, 2 static and 2 transient. Three are FORCE vectors, and one is a MOMENT vector. You will note that one vector is located with a node id, one with an element id, and 2 with xyz locations.

We used some blank lines and some lines starting with #, for comments.

Note the use of the appropriate number of xyzloc and/or values lines.

Note that for the example, we did the TIMELINES `times` lines differently. The first TIMELINE put all times on the same line - which we can easily do without violating the 15 token limit because there are only 4 times. The second TIMELINE put each time on its own `times` line.

In this case we did not re-use the transient timelines, but we easily could have more than one vector glyph reference the same timeline.

```
EnSight Vector Glyphs
Version 1.0
#--------------------
NumVectorGlyphs: 4
NumTimeLines: 2
#--------------------
VECTOR
id: 1
description: dead load
type: FORCE
time_condition: STATIC
part: 1
nidloc: 173
values: 0.5 1.0 0.75

VECTOR
id: 2
description: wind load
type: FORCE
time_condition: TRANSIENT
time_line: 1
part: 1
eidloc: 13
values: 1.0 2.0 0.0
values: 0.0 2.0 1.0
values: 0.5 7.0 3.0
values: 2.0 5.0 8.0

VECTOR
id: 4
description: snow load
type: FORCE
time_condition: STATIC
part: 1
xyzloc: 1.5 1.5 1.0
values: 0.0 0.0 -5.25

VECTOR
id: 5
description: wrench
type: MOMENT
```

```
time_condition: TRANSIENT
time_line: 3
part: 72
xyzloc: 1.0 1.0 0.0
xyzloc: 1.1 1.8 0.2
xyzloc: 1.2 1.7 0.5
values: 10.0 15.0 2.0
values: 20.0 30.0 4.0
values: 100.0 150.0 20.0

#---------------------------

TIMELINE
id: 1
numtimes: 4
before: UNDEF
amidst: INTERPOLATE
after:  NEAREST
times: 0.0 1.0 2.0 3.0

TIMELINE
id: 3
numtimes: 3
before: NEAREST
amidst: INTERPOLATE
after:  UNDEF
times: 0.5
times: 1.5
times: 2.5
```

11.16 Constant Variables File Format

This file contains constant variables information. Constant variables are defined two ways. One way is via the EnSight Gold Case file where constant variables can be defined per case under the VARIABLE section, i.e

```
VARIABLE
constant per case: [ts] Density 0.92
```

(see subsection EnSight Gold Case File Format under EnSight User Manual Chapter 11.1).

The other way to define constant variables is by loading them using this Constant Variables File Format. The advantage of this format is that all the constant variables defined in this file will be created if they do not exist or updated if they already exist, each time they are loaded at any stage during the EnSight session.

This file is specified via the Feature Detailed Editor (Calculator) or (Variables) dialog under File > Load constants from file...

General Comments:

1. This is an ASCII file.

2. The file must start with the following two lines:

```
#Version 1.0
#EnSight Constant Variables
```

3. Any following line that begins with a "#" is recognized as a comment line. Also, text following a "#" symbol is ignored as comments.

4. Each non-comment line has the following syntax

```
Constant_description1 Constant_value1
Constant_description2 Constant_value2
. . .
Constant_descriptionN Constant_valueN
```

where:
Constant_description = the variable name of the constant

Constant_value = the value of the constant

5. This file format only supports static and not transient (or changing time) constant variables. Although this file can be loaded at any time step.

6. This file may also be loaded at anytime with no regard to data file format.

Example:

```
#Version 1.0
#EnSight Constant Variables
#
# Comment lines are supported and ignored during loading.
# But, the 1st two lines must be as indicated above.
#
Density .98
Pressure 1.66
Viscosity 1.965e-05  # comment appending text line is permitted
                     # and ignored during loading
```

Line 1: Required version line; verbatim as indicated above,
i.e. "#Version 1.0"

Line 2: Required file type line; verbatim as indicated above,
i.e. "EnSight Constant Variables"

Line 3 and on: Variable description, white-space, and variable value
(free format) for each variable.

11.17Point Part File Format

This file format is for a series of points that can be used to place points in a certain location to be used as probes within a volume part (see Chapter 7.22, Point Parts Create/Update). The point data in this file is not loaded into EnSight as geometry. The format to load points as geometry is a different section (see Chapter 11.1, EnSight Gold Casefile Format).

Point part points can be loaded from a text file with the following format:

Line 1 is the version number in the form "#Version <version number>", where version number is a floating point value. The version number is currently 1.0.

Line 2 is a keyword indicating the file type: “#EnSight Point Format”.

lines 3 through n contain the floating point values for the x, y and z coordinates, respectively, of each point. The values must be separated by commas or spaces and each line may contain the x, y, and z values of only one point, terminated by a carriage return. The number can be exponential (2 or 3 exponent values) or floating. Any lines 3 through n that begin with a “#” are considered comments and are ignored.

example:

```
#Version 1.0
#EnSight Point Format
0.00000e+00,0.00000e+00,0.00000e+00
# This is a comment line and is ignored
1.00000e-01,1.00000e-001,3.00000e-001
2.0,3.00000e-01,6.00000e-001
3.00000e+00,3.50000e-01,1.70000e+00
4.20000e+000,5.00000e-001,1.80000e+000
5.40000e+000,6.00000e-001,1.90000e+000
6.50000e+000,8.00000e-001,1.95000e+000
7.00000e+000,1.00000e+000,2.00000e+000
```

11.18Spline Control Point File Format

A Spline Control Point file is used to define one or more splines that can be used in EnSight for positioning entities such as the camera or the plane tool. This file is a text file with the following format:

Line 1 is the version number in the form "#Version <version number>", where version number is a floating point value. The version number is currently 1.0.

Line 2 is a keyword indicating file type: "#EnSight Spline Control Points".

Line 3 is the keyword to start a new spline and provide the text description of the spline in the form "DESCRIPTION: <spline name text>" where the spline name text is at most a 49 character text description.

lines 4 through n contain the floating point values for the x, y and z coordinates, respectively, of each point. The values must be separated by commas and each line may contain the x, y, and z values of only one point, terminated by a carriage return. The number can be exponential (2 or 3 exponent values) or floating.

Lines beginning with a "#" are considered comments and ignored.

Multiple splines can be contained in one file. The keyword "DESCRIPTION: <spline name>" begins a new spline point listing.

example:

```
#Version 1.0
#EnSight Spline Control Points
DESCRIPTION: First Spline
0.00000e+00,0.00000e+00,0.00000e+00
1.00000e-01,1.00000e-001,3.00000e-001
2.00000e-01,1.00000e-001,3.00000e-001
3.0,3.50000e-01,1.70000e+00
4.20000e+000,5.00000e-001,1.80000e+000
5.40000e+000,6.00000e-001,1.90000e+000
6.50000e+000,8.00000e-001,1.95000e+000
7.00000e+000,1.00000e+000,2.00000e+000
# This is a comment line and is ignored
DESCRIPTION: Second Spline
0.00000e+00,0.00000e+00,0.00000e+00
9.00000e-01,1.00000e-001,3.00000e-001
2.0,3.00000e-01,6.00000e-001
3.00000e+00,3.50000e-01,1.70000e+00
```

11.19EnSight Embedded Python (EEP) File Format

The EnSight Embedded Python file format is a mechanism that allows a user to create portable archives of data files and scripts that can be opened and read by EnSight through its Python interpreter. The file format itself is the standard ZIP format used by many tools (e.g. WinZip, 7-Zip, PKZIP, zip, built into Windows and OSX, etc). To change an existing ZIP file into a EEP file, simply change its name to end in .eep. EnSight will recognize these files when passed on the command line, dropped on the running application, run by the play: command or opened via the command dialog. When presented with an EEP file, EnSight scans its contents looking for the presence of a file named "__init__.py" or "autoexec.py". If files by both names are found in the EEP archive, it treats the file as if only the "autoexec.py" file was found. The two cases are handled as outlined below:

The "module" case ("__init__.py"):

If EnSight detects a file named "__init__.py", it treats the contents of the file as an EnSight module. It will extract all of the files in the EEP archive into a new, temporary directory located in the same directory as the EEP file. If EnSight cannot create a directory in that location, the operation will fail. Note: when EnSight exits, the temporary directory and all of its contents will be deleted. The directory which contains the temporary will be added to the EnSight Python interpreter sys.path and the temporary directory will be imported as a module using the 'import' command. As a result, the "__init__.py" file will be executed by EnSight and the '__file__' variable will point to the '__init__.py' file. If the same EEP file is opened a second time in a session, the files will not be re-extracted, but reload() will be called on the module and "__init__.py" executed a second time.

The "installer" case ("autoexec.py"):

If EnSight detects a file named "autoexec.py", it will extract the contents of the file as a string and execute itas Python code (similar to the exec() function). One special variable will be set up before the string is executed. The Python variable '__file__' will be set to the (string) name of the EEP file. Thus, the Python code in "autoexec.py" can access the EEP file contents using the name in the '__file__' variable and Python modules such as 'zipfile'. An advantage of this approach is that it does not require write access to the filesystem or the space to uncompress the data. This approach requires a more complex user written Python script, but should be used when the data in the EEP file is large, the filesystem is read only (e.g. CDROM) or only a portion of the data in the file is needed.

Usage notes:

The EEP file is a very general mechanism for packaging data and scripts in a portable fashion and getting EnSight to perform scripted operations. It can be useful for creating demos, building installers for EnSight enhancements, performing automated testing and other tasks. The use of the Python interpreter inside of the EnSight application itself greatly simplifies the run-time environment for such applications.

11.20 Camera Orientation File Format

A camera orientation file is used to define the location, orientation, and lens field of view angles. The orientation file can be restored in the Transformation Editor (Camera) dialog via the File->Restore camera position option. The restored values will apply to the camera selected and the aspect ratio information from the field of view angles will be applied to modify any viewport using the camera.

The file is keyword delimited. The keywords can appear anywhere in the file. Any line with a # sign in the first column is considered a comment line. Any blank line is also considered a comment line.

FORMAT: Must be set to CAMERA_SETUP

VERSION: Must be set to 1.0

CAMERA LOCATION: Three space deliminted floating point values indicating the camera origin

CAMERA LOOK AT: Three space deliminted floating point values indicating a point along the camera view axis. The combination of the look at and the camera location creates a camera view vector.

CAMERA ROTATION: Currently (version 1.0) ignored.

CAMERA FOV HORIZONTAL: A floating point value specifying the horizontal field of view angle of the camera lens

CAMERA FOV VERTICAL: A floating point value specifying the vertical field of view angle of the camera lens

Example:

```
FORMAT:  CAMERA_SETUP
VERSION:  1.0
#
#  camera location is the coordinate location of the camera
#
CAMERA LOCATION:  -1500 50 150
#
#  camera look at is the 3D coordinate of the center pixel of the IR image
#
CAMERA LOOK AT:  0. 50 150
#
# camera rotation in degrees about the axis from the camera location to the look at
point.
# Right hand rule.
#
CAMERA ROTATION:  0.
#
# field of view values in degrees for the horizontal and vertical axis of the lens
#
CAMERA FOV HORIZONTAL:  21.0
CAMERA FOV VERTICAL: 15.75
```

11.20 Example:

12 Utility Programs

This chapter describes the utility programs that accompany EnSight. The Server utility programs are located in `$CEI_HOME/ensight92/server_utilities` and the Client utility programs are located in `$CEI_HOME/ensight92/client_utilities`.

Utility programs are supplied on an "as is" basis and are unsupported. CEI will, however, try to assist in problem resolution.

Each utility program is presented below and accompanied with a brief overview that describes the function of the utility.

12.1 EnSight Case Gold Writer

EnSight Case Gold Output API

The EnSight Case Gold Output API will allow users to make simple calls to our API to write out EnSight's Case Gold format. This is currently under development. If you have a need to output our format, please contact support@ensight.com for the latest information.

13 Parallel and Distributed Rendering

EnSight supports several types of parallel rendering for increased performance, increased display resolution, and virtual reality. EnSight Gold supports shared-memory parallel rendering, which is limited to a single EnSight client and up to 36 screens. EnSight DR supports distributed-memory parallel rendering, which allows multiple EnSight clients running on one or more workstations to combine rendering results to support high-resolution and/or high-performance rendering.

Combined with support for 6 DOF (degree-of-freedom) input devices, EnSight Gold and DR provide an immersive virtual reality interface. This chapter describes the configuration file formats and command-line parameters required for parallel rendering and 6D input. Hardware accelerated parallel rendering is supported on all platforms, provided that the OpenGL graphics driver is thread-safe. Six DOF input is supported for all platforms, with pre-compiled trackD support for Windows (32-bit and 64-bit) and Linux (32-bit and 64-bit).

Configuration File formats

Proper settings for the various configurations are discussed throughout this chapter.

Resource File format

Details for the Resource file format, which is also discussed in this chapter can be found in How To Use Resource Management.

13.1 Shared-memory parallel rendering

In order to make use of shared-memory parallel rendering with EnSight, the user must create a configuration file. This file is specified on the command line using the argument `-dconfig <file>`. If `<file>` is not a fully-qualified path EnSight will search for the file in the following directories:

1. `~/.apex21/dconfig`
2. `$CEI_HOME/apex21/site_preferences/dconfig`

These options allow for user-level and site-level configurations, respectively. There are two logical displays that can be configured in EnSight. The *GUI display* is always active, and consists of the main rendering window embedded in the user-interface. The *detached display* is external to the user-interface, and may consist of 1-36 regions configured to form a large continuous display. The configuration file contains information about both the GUI display and the detached display, as well as tracking calibration information and options for using 6D input devices. The following sections will address each of the capabilities related to parallel rendering and VR. The sample configurations described in this chapter can be found in the directory `$CEI_HOME/ensight92/doc/dconfig`. There are also examples of 'simulated' configuration files, which allow you to simulate display to multiple graphics pipes on a single display.

CONFIGURATION FILE FORMAT

Configuration files are text-based beginning with the line:

```
CVFd 1.0
# after the first line, anything following a '#' is a comment
```

The remainder of the file consists of one or more sections describing the displays and options. In describing the format of the file, portions which are optional will be surrounded by [].

DISPLAY WALLS

Shared-memory parallel rendering in EnSight allows for the use of multiple graphics pipes to create large, flat tiled displays. Commonly referred to as display walls, this is an example of a "*detached display*" supported by EnSight. Detached displays are supported on all platforms, including Windows. The advantage of a display wall configuration is that the file specification is easy to create. The disadvantage is that display walls cannot be used for tracking and 6d input. In order to use tracking, it will be necessary to use the more general immersive configuration format described later.

The specification for a display wall consists of:

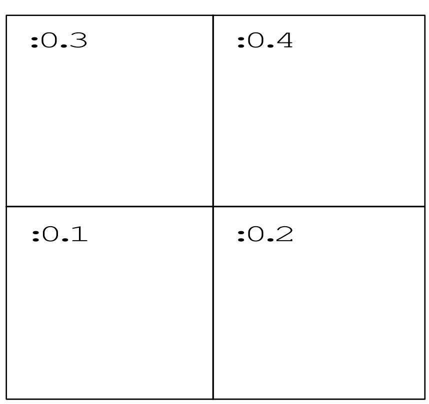
```
display
wallresolution
   <x-res> <y-res>
```

```
screen
   displayid <p1>
   resolution <x-res> <y-res>
   walloriging <wall-x> <wall-y>
   [ displayorigin <xo> <yo> ]
   [ lefteye
       or
     righteye
   ]
[ repeat 'screen' section for each additional screen ]
```

The `wallresolution` section gives the total pixel resolution of the display wall. For each graphics pipe, there will be a screen section that describes the size (`resolution`) and position (`wallorigin`) of the region within the global display. The `displayid` parameter specifies the X display (i.e. :0.1). This parameter is ignored on Windows, because there is only a single “screen” regardless of the number of graphics cards or video ouputs. The `displayorigin` is an optional parameter to specify the origin of the window on the given pipe (default (0,0)). Note that `displayorigin` is a position relative to the origin of a given `displayid`, while `wallorigin` is a position relative to the origin of the global display. Changing `wallorigin` will change the region of the wall that is visible in a given window, while changes to `displayorigin` simply move the window on the screen without changing the contents. Example 4a will demonstrate a situation when the use of `displayorigin` is useful. The `lefteye/righteye` optional designation can be used for passive stereo displays, in which separate graphics pipes render the left and right images. Note that each pipe in a detached display can have one or more `worker` pipes configured to accelerate the rendering, just as described in the previous section.

Example 1

In this example there is one X server with five graphics pipes. The GUI is displayed on pipe :0.0, with the other four pipes used for the detached display. Four projectors are configured in a 2x2 array to form a large continuous wall as illustrated:



```
CVFd 1.0
#
# conference room display wall
#
display
wallresolution
   2560 2048
screen # lower-left
   displayid :0.1
   resolution 1280 1024
   wallorigin    0    0
screen # lower-right
   displayid :0.2
   resolution 1280 1024
   wallorigin 1280    0
screen # upper-left
   displayid :0.3
   resolution 1280 1024
   wallorigin    0 1024
screen # upper-right
   displayid :0.4
   resolution 1280 1024
   wallorigin 1280 1024
```

Example 2

It is not uncommon for displays walls to use overlapping images with *edge-blending* to smooth the otherwise sharp transition between projector images. The edge-blending is performed by the projectors directly. This is easily configured as a detached display by specifying pipes with overlapping pixel regions. Consider an example of two pipes at 1280x1024 resolution each, with an overlap of 128 pixels.

```
CVFd 1.0
#
# edge-blending example
#
display
wallresolution
   2432 1024
screen # left
   displayid :0.1
   resolution 1280 1024
   wallorigin    0    0
screen # right
   displayid :0.2
   resolution 1280 1024
   wallorigin 1152    0
```

Note that in this case the total resolution of the wall in the x direction is decreased by the amount of overlap.

Example 3

Passive stereo displays achieve stereo by projecting overlapping polarized images from multiple projectors. This can be achieved using detached displays with a distinct rendering region for each screen and eye. Consider for this example a single screen with two projectors. For illustration purposes we will assume that we have five graphics pipes. One pipe (:0.0) renders the GUI and is not listed.

```
CVFd 1.0
#
# passive stereo display
#
display
wallresolution
   1280 1024
screen # left-eye
   displayid :0.1
   resolution 1280 1024
   wallorigin    0    0
   lefteye
screen # right-eye
   displayid :0.3
   resolution 1280 1024
   wallorigin    0    0
   righteye
```

Note that the `lefteye/righteye` parameters are NOT necessary when using traditional quad-buffered stereo to drive the projectors. Some systems have a signal splitter which takes the frame-sequential stereo signal and generates separate signals for left and right eye. In this case a conventional configuration file without the “eye” designations will work fine. Passive stereo displays are always in stereo mode.

IMMERSIVE DISPLAYS

True immersive display requires more information than is present in the display wall configuration files previously described. The key factors are that (1) immersive displays are often not flat and (2) the rendered images must be co-registered with the coordinates of a 6d input tracking system.

The basic syntax of the immersive display configuration is going to look very similar to the display wall format:

```
display
screen
   displayid <p1>
   resolution <x-res> <y-res>
```

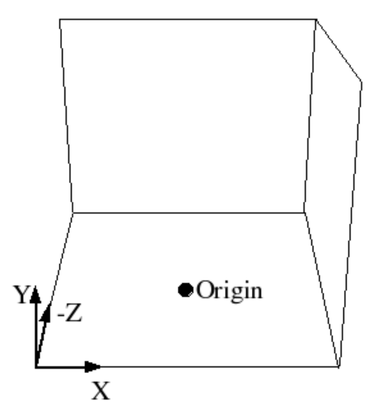
```
    [ displayorigin <xo> <yo> ]
    [ bottomleft  <x> <y> <z>
      bottomright <x> <y> <z>
      topleft     <x> <y> <z>
    ]
    [ lefteye
        or
      righteye
    ]
   [ repeat 'screen' section for each additional screen
]
```

The important difference is that the position of the screen is measured in 3d physical coordinates, rather than 2d pixel coordinates. Note that all 3d coordinates given in the file are unit-less, but they must be consistent and in the same frame of reference, which is referred to as "display coordinate space".

The keywords bottom/top refer to the minimum Y/maximum Y of the region, and left/right refer to the minimum X/maximum X of the region. In some cases 'bottom' may be near the ceiling, and 'top' may be near the floor, such as when a projector is mounted in an inverted position.
When determining the proper coordinates to use it is invaluable to sketch out the display environment, label the corners of each screen, and mark the location of the origin of the coordinate system. When using 6D input, the display coordinate system and the tracking coordinate system must be the same.

Example 4

For the purpose of illustration consider the following example. Two projectors are pointed at screens which form a right angle, as illustrated below. The projected images are 10 feet wide by 7.5 feet high. The tracking system is calibrated in units of feet with the origin on the floor in the middle of the room.




```
CVFd 1.0
display
```

```
screen
   displayid :0.2
   resolution 1024 768
   bottomleft  -5 0.0 -5
   bottomright  5 0.0 -5
   topleft     -5 7.5 -5
screen
   displayid :0.1
   resolution 1024 768
   bottomleft   5 0.0 -5
   bottomright  5 0.0  5
   topleft      5 7.5 -5
```

Without head-tracking, this example is not yet very useful. The default position of the viewer is at (0,0,0), which is on the floor in the chosen coordinate system. There is an optional `view` section that can be inserted before the first screen of the configuration file to change these defaults:

```
view
   [ origin <x> <y> <z> ]
   [ zaxis  <nx> <ny> <nz> ]
   [ yaxis  <nx> <ny> <nz> ]
   [ center <x> <y <z> ]
   [ scale  <factor> ]
   [ eyesep  <d> ]
```

The `origin` specifies the position of the viewer, and is only used if head-tracking has not been enabled. The `zaxis` and `yaxis` are unit vectors that allow the specification of a default orientation for objects placed in the scene. The default values are (0,0,-1) for `zaxis` and (0,1,0) for `yaxis`. From the `origin` vantage point, it is useful to think of `zaxis` as the direction that the viewer is looking and `yaxis` as the 'up' direction.

The `center` and `scale` parameters allow you to position and size the scene for your display. If these parameters are not given, EnSight will compute a bounding box from the 3d coordinates given in the `bottomleft`, `bottomright`, and `topleft` parameters for the screens. The default center will be at the center of this box and the default scale will be computed so that your EnSight scene will fill the 3d space. Specifying a scale factor of 1.0 may be useful if your display coordinates were designed to coincide with your model coordinates. This will allow you to view your models life-sized.

The `eyesep` parameter allows an exact setting of the stereo separation between the eyes.

Example 4a

Extending our example, we can position the viewer at the opposite corner of the room at a height of 5.75 feet:

```
CVFd 1.0
display
view
   origin -5 5.75 5
screen
   displayid :0.2
   resolution 1024 768
   bottomleft -5 0.0 -5
   bottomright 5 0.0 -5
   topleft    -5 7.5 -5
screen
   displayid :0.1
   resolution 1024 768
   bottomleft   5 0.0 -5
   bottomright  5 0.0  5
   topleft      5 7.5 -5
```

Example 4a-sim

It is relatively straightforward to test large displays and VR environments on a smaller system with a different number of graphics pipes. This can be accomplished by creating a configuration file that maps the pipes to smaller regions on a single monitor. As an example we will take the immersive configuration from Example 4a and modify it to run on a single display, with the modified regions shown in bold text.

```
CVFd 1.0
display
view
   origin -5 5.75 5
screen
   displayid :0.0
   resolution 320 240
   bottomleft  -5 0.0 -5
   bottomright  5 0.0 -5
   topleft     -5 7.5 -5
screen
   displayid :0.0
   displayorigin 320 0
   resolution 320 240
   bottomleft   5 0.0 -5
   bottomright  5 0.0  5
   topleft      5 7.5 -5
```

Note that this method makes use of the `displayorigin` parameter so that the resulting windows do not overlap. The default value for `displayorigin` is (0,0) for each pipe. In a similar manner it is also possible to simulate large display walls on a single pipe.

TRACKING

EnSight supports tracking and input with 6 DOF devices through a defined API. Pre-built libraries are provided to interface with trackd ((C) VRCO, Inc., www.vrco.com) on Windows (32-bit and 64-bit) and Linux (32-bit and 64-bit), or the user may write a custom interface to other devices or libraries. The tracking library is specified with the `CEI_INPUT` environment variable. To select trackd, use:

`setenv CEI_INPUT trackd` (for csh or equivalent users)

The value of `CEI_INPUT` can either be a fully-qualified path and filename or simply the name of the driver, in which case EnSight will load the library libuserd_input-$(CEI_INPUT).so from directory:

```
$CEI_HOME/apex21/machines/$CEI_ARCH/udi/
```

For the trackd interface you will also need to set:

```
CEI_TRACKER_KEY <num>
CEI_CONTROLLER_KEY <num>
```

in order to specify the shared-memory keys for the input library to interact with trackd. You should be able to find these values in your trackd.conf configuration file.   For information on the API which allows you to interface to other tracking libraries or devices, please see the `README` file in `$CEI_HOME/ensight92/src/udi`.

With the environment variables set, you are ready to activate tracking. There are two parts to this. First, trackd operates as a daemon that is run independent of EnSight. If your input interface includes a separate program, you can run it at this time. For trackd users, it is often useful during configuration to invoke trackd with the `–status` option, so that you can see the information on your input devices. Once any external programs are started, you can enable tracking in EnSight. From the ‘`Preferences->User Defined Input`’ menu, there is a toggle button which turns tracking on and off.

The trackd driver shipped with EnSight also has a debug mode that can be activated as follows:

```
setenv CEI_TRACKD_DEBUG  1
```

This is similar to the trackd -status option, but it reports the input as seen by the EnSight trackd interface.

Once the EnSight client has been correctly interfaced to a tracking system you can add a section to the configuration file in order to calibrate the tracking with the display frame and customize the behavior of various interactions. The syntax for the section is:

```
tracker
   [ headtracker <i> ]
```

```
[ cursortracker <i> ]
[ selectbutton <i> ]
[ rotatebutton <i> ]
[ transbutton <i> ]
[ zoombutton <i> ]
[ xformbutton <i> ]
[ transxval <i> ]
[ transyval <i> ]
[ transzval <i> ]
[ auxbutton <i> <j> ]
[ motionfilter <i> <p> <r> ]
```

The `headtracker` and `cursortracker` parameters allow you to specify which tracking device is tracking head position and which is tracking the controller. At this time only two devices can be tracked by EnSight – one for the head position and one for the position of the controller. All button/valuator input is interpreted as having come from the controller. Note that the EnSight API for input devices uses 0-based indices for trackers, buttons, and valuators. Trackd uses 1-based indices, and other libraries may differ as well.

The remaining options allow you to customize the behavior of buttons and valuators on the 6D input device. The input device can be used for:

1. Selecting items from the 3D GUI, which includes the heads-up macro (HUM) panel, the part list, variable list, and value slider.
2. Performing transformations on the geometry in the scene.
3. Manipulating the cursor, line, plane, and quadric tools.

The input device has a local coordinate system which is relevant for some forms of 6D interaction:

The default mode defines button 0 as the select button. When the 3D GUI is visible, you can point at the 3D buttons and the item that you are pointing at will be displayed in a highlight color. When you press the select button you will activate the current selection. For the HUM panel, this means that you will activate the macro that is defined for the selected button. See the “User Defined Input Preferences” found in Section 6.2, Edit Menu Functions for more instructions on configuring and showing the HUM panel and part panel. Clicking on an item in the part list will select or unselect the item in the list. Combined with macros in the HUM, this will allow you to modify visibility or other attributes on a part or collection of parts. If there are many parts in the part list, you can also select the scrollbar and move the controller up and down to scroll through the list. Similarly, the part-value slider can be used to modify part attributes for certain part types. For isosurfaces you can select the part slider and move left to right to change the isovalue. When no parts are selected, the part-value slider can be used to modify the time in a transient simulation.

The `rotatebutton`, `transbutton`, and `zoombutton` allow you to perform the selected transformations using gestures with the 6d input device. The `xformbutton` allows you to link a button to the current transformation mode, similar to the mouse button configurations for the main GUI interactions. You may want to add buttons on the heads-up-macro (HUM) panel to switch between modes. This is useful for 6D input devices with a smaller number of buttons. Note that it is possible (and encouraged) to re-use the `selectbutton` for a transformation. The `selectbutton` is only used when you are pointing at a heads-up menu. When you are not pointing at a menu, the same button could be used as the `xformbutton`, for example.

All 6d transformations have a ‘sensitivity’ which can be set to control the speed at which the transformation occurs. These values can be set from the ‘Edit->Preferences->User Defined Input’ dialog. There are also two forms of rotation available. In ‘Mixed Mode’, the 6d device acts similar to a mouse for rotation. Once you click the `rotatebutton`, your movement is tracked in the X-Y plane of the input device. Your translation in this space is mapped to a rotation in the 3D space. In ‘Direct Mode’ it is the orientation of the device, rather than the position of the device, which controls the rotation.

The `transxval`, `transyval`, and `transzval` parameters configure the valuators to allow for translation of the scene by pressing the valuator in a given direction. The 'x', 'y', and 'z' designations refer to a local coordinate system which is fixed to the controller input device. As you hold the device in your hand, positive x is to the right, positive y is up, and positive z is toward the viewer. This local coordinate system depends on the orientation of the tracking device attached to the input device. It may be necessary to align the tracking device properly or modify the trackd (or other tracking library) configuration to achieve the proper orientation.

The auxbutton parameters configure additional buttons that can be used to control various options. The first parameter "<i>" is the number of the auxbutton to define. The second parameter "<j>" is the physical device button to bind to auxbutton <i>. Currently, auxbutton 0 is used by EnSight as the "Menu" button. This button can be used to bring up the "User defined menu" at the cursor point on the annotation plane. Pressing the button a second time will either reposition the menu or it will pop up to the next level of the menus if a submenu has been selected.

The motionfilter parameter allows the user to select a filter threshold for each tracker (head or cursor). If the motion of the tracker exceeds this threshold, and

EnSight has the Fast Display mode global toggle enabled, it will display all the parts using their fast representation until the motion drops below the threshold. The <i> parameter selects the tracker to filter. The <p> value specifies a threshold for the tracker position. This is computed as the variance of the distance between the current position and the last 10 tracker positions. Similarly, the <r> value specifies a threshold for the tracker direction vector. This is computed as the variance of the angles between the current direction and the last 10 directions (in radians).

Example 5

For the most basic configuration with head-tracking and a 6d input device, there are only three lines added to Example 4 to create the `tracker` section:

```
CVFd 1.0
display
view
   origin -5 5.75 5
tracker
   headtracker 0
   cursortracker 1
screen
   displayid :0.2
   resolution 1024 768
   bottomleft  -5 0.0 -5
   bottomright  5 0.0 -5
   topleft     -5 7.5 -5
screen
   displayid :0.1
   resolution 1024 768
   bottomleft   5 0.0 -5
   bottomright  5 0.0  5
   topleft      5 7.5 -5
```

Example 6

There are many different input devices available, and some have additional buttons and valuators that can be used for navigation and selection in immersive environments. In this example the configuration file is extended to use different buttons for rotation, translation, zoom, and selection. We also configure a ‘thumbwheel’ input to provide translation in the X-Z plane.

```
CVFd 1.0
display
view
   origin -5 5.75 5
tracker
   headtracker 0
   cursortracker 1
   selectbutton 4
```

```
   rotatebutton 0
   transbutton 1
   zoombutton 2
   xtransval 0
   ztransval 1
screen
   displayid :0.2
   resolution 1024 768
   bottomleft  -5 0.0 -5
   bottomright  5 0.0 -5
   topleft     -5 7.5 -5
screen
   displayid :0.1
   resolution 1024 768
   bottomleft   5 0.0 -5
   bottomright  5 0.0  5
   topleft      5 7.5 -5
```

OTHER CONTROLLERS

EnSight supports simple rotation and translation of the current view using external input devices such as joysticks, game controllers and the 3DConnexion SpaceNavigator ™ (http://www.3dconnexion.com). This is controlled via the configuration file 'spacedevice.defaults' located in the system site_preferences directory or users resource directory. EnSight will automatically detect and use devices like the SpaceNavigator, but the configuration file can be used to fine-tune the sensitivity of the device. The default file looks something like:

```
VERSION 1.0
#
# Comments must have the '#' as the first character.
#
# ranges are xmin xmax ymin ymax zmin zmax
#
# these are the values for sensitivity of 50
translate_ranges -400 400 -400 400 -400 400
rotate_ranges -400 400 -400 400 -400 400
```

The values set the minimum and maximum ranges for the translation and rotations. Adjusting them can correct for asymmetry in the output values from the device and set the device's sensitivity. Narrowing the range has the effect of increasing the sensitivity to smaller motions of the device.

On Windows and Linux platforms, EnSight also supports standard PC joysticks and game controllers using native joystick APIs on those platforms. To use these devices, three additional fields can be used: controller_id, controller_config and single_axis.

`single_axis {0|1}`

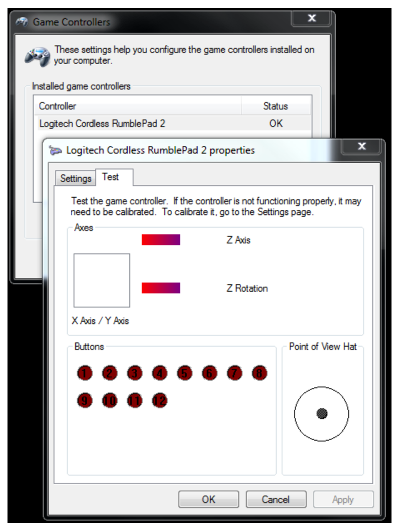
By default, the spacedevice interface only allows transformation by a single axis at a time. If this value is set to 0, transformations over multiple axes (either translations or rotations) are allowed at the same time.

`controller_id "name"`

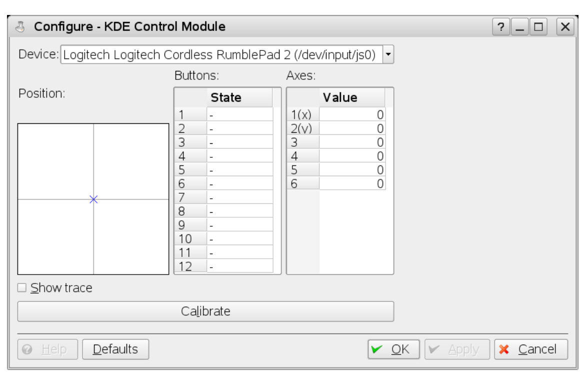
The presence of this field enables the use of joysticks/game controllers. The value must be a string in quotes. Under Windows, the value is an integer (as a string) which is the index of the game controller in the system. The first controller is "0". Under Linux, this is the device name of the joystick in question. An example is "/dev/input/jp0". The actual value is dependent on the specific Linux distribution.

`control_config "configstring"`

This field maps device controls (axes and buttons) to the transformation axes. The configuration string consists of some number of axis transformation expressions separated by spaces. Each expression is in the form {transformation axis}={device axis}{u}{device button number}. The axes of transformation are: X: translation in screen X, Y: translation in screen Y, Z: translation in screen Z, A: rotation over the screen X axis, B: rotation over the screen Y axis and C: rotation about the screen Z axis. The device axes follow the Windows game controller naming conventions. The legal values are 'X', 'Y', 'Z', 'R', 'U', 'V'. The mapping of various device controls to these axes varies from controller to controller. Under Windows, the game controller control properties panel allows the user to see the layout of the controls and see what buttons are mapped to which button numbers interactively. The rumble pad uses the X, Y, Z and R axes.



Under Linux, the 'X', 'Y', 'Z', 'R', 'U', 'V' axes map to the first though sixth analog axes respectively.



If no button number is specified, the game controller always controls the selected EnSight transformation. Each axis mapping can be qualified by a button state. For example Y=Z5 means that EnSight translation in Y will be controlled by game controller Z axis, but only while button 5 is pressed. Putting a lowercase 'u' between the controller axis and the button number specifies the link is only made when the named button is not pressed (up state). Thus, B=Ru6 specifies that rotation over the screen Y axis is controlled by the game controller R axis, but only when the button 6 is not pressed (up).

An example using a Logitech RumblePad 2 on a Windows system might be:

```
VERSION 1.0
#
# Comments must have the '#' as the first character.
#
# ranges are xmin xmax ymin ymax zmin zmax
#
# these are the values for sensitivity of 50
translate_ranges -400 400 -400 400 -400 400
rotate_ranges -400 400 -400 400 -400 400
#
controller_id "0"
controller_config "X=X Y=Yu5 Z=Y5 A=Ru6 B=Z C=R6"
single_axis 0
```

This file selects the first controller in the system (controller_id "0"), sets the configuration string and enables transformations in multiple axes at the same time. The configuration string maps translation in the X axis to the left stick horizontal axis ("X=X"). It maps translation in the Y axis to the left stick vertical axis, when button 5 is "up" ("Y=Yu5"). Translation in the Z axis is mapped to the left stick vertical axis when button 5 is "down" ("Y=Y5"). Similarly, the right stick axes (R and Z) are mapped to the rotation axes (A, B, C) conditioned by the status of button 6.

Note: the default configuration string maps well to the Xbox style controllers and amounts to: "X=X Y=Yu5 Z=Y5 A=R B=Uu6 C=U6".

ANNOTATIONS

Annotations in EnSight include the heads-up macro panel, text, lines, logos, legends, and plots. In the GUI display these items appear as an overlay which is fixed in screen space. In an immersive display environment it is useful to be able to specify the locations of these objects. In EnSight, these items continue to occupy a plane in the 3D world. By default, this plane will coincide with the first pipe in the configuration file. The user may choose to specify the position and orientation of this plane with the following addition to the configuration file:

```
Annot
   [ screen <n> ]
         OR
   [
     center <x> <y> <z>
     zaxis  <x> <y> <z>
     yaxis  <x> <y> <z>
     xscale <float>
     yscale <float>
   ]
```

Example 6

To continue with Example 5, suppose that the user would prefer for the annotations to appear on the right wall instead of the left wall. The following configuration file defines an annot section with the appropriate parameters to do this:

```
CVFd 1.0
display
view
   origin -5 5.75 5
tracker
   headtracker 0
   cursortracker 1
   selectbutton 4
   rotatebutton 0
   transbutton 1
   zoombutton 2
   xtransval 0
   xtransval 1
annot
   screen 1
screen
   displayid :0.2
   resolution 1024 768
   bottomleft  -5 0.0 -5
   bottomright  5 0.0 -5
   topleft     -5 7.5 -5
screen
   displayid :0.1
   resolution 1024 768
```

```
   bottomleft   5 0.0 -5
   bottomright  5 0.0  5
   topleft      5 7.5 -5
```

Fixing the annotations to a pipe is merely provided as a convenience. Internally this is identical to using the explicit form:

```
annot
   center 5 3.75 0
   zaxis 1 0 0
   yaxis 0 1 0
   xscale 10
   yscale 7.5
```

STEREO DISPLAY

When using a detached display (either a wall or an immersive configuration) the created windows will be monoscopic by default. If you want the display to be initialized in stereo, you can simply add the keyword to the configuration file between any of the other sections.

```
stereo
```

This keyword is not necessary for passive stereo displays, which are always in stereo.

TIPS FOR SHARED MEMORY PARALLEL RENDERING

1. Use the -bbox command-line option when using a detached display. This will cause EnSight to draw only bounding boxes in the GUI window, which will improve performance
2. Make sure that the displays that are in your configuration files are valid. An X display identifier looks like: `<host>:<server>.<screen>`. If you have constructed a valid X display you should be able to set the `$DISPLAY` environment variable with the given string and run `xterm` or another X11 application.
3. Tracking is the most difficult part of configuring a system. Make sure that you are confident in the display configuration before you activate tracking. It may be useful to manually position the view origin in several different locations in order to verify that the display coordinate system is as you expected. In the examples installed with EnSight example 4 is extended with several different view origins to demonstrate this technique

13.2 Distributed Memory Parallel Rendering

Distributed-memory parallel rendering allows the user to combine the capability of multiple high-performance graphics pipes (in either a shared-memory machine or cluster configuration) in various ways. The following brief descriptions of the distributed rendering methods will help you decide which approach is best for your needs. It is quite likely that a user of EnSight DR will decide to use different parallel rendering configuration in different circumstances.

Compositing

Parallel compositing enables users with an extremely large amount of visible geometry to distribute the client-side computation and rendering among multiple CPUs and GPUs in either a shared-memory machine or a cluster of workstations. The final result is an image in the EnSight rendering window that is indistinguishable from a standard EnSight client running on a single workstation, but at much higher performance.

Pros:

- very-high polygon rates (Billions of triangles / sec)
- scalable client memory and computation
- ability to render remotely and view locally

Cons:

- frame-rate upper-bound determined by network bandwidth
- resolution limited to single workstation display

Compositing is for users working in a desktop environment who have large data which overwhelms the capability of a single workstation, in terms of memory, processing power, and/or rendering performance. Note that compositing is not an application for ganging together dozens of old workstations that have no other use - compositing itself requires fast processors and a fast network in order to achieve any measure of scalability.

Replication

Replication is a distributed rendering strategy for users with data of moderate size which can be rendered on a single workstation at acceptable frame-rate, who are looking to increase display resolution or render to multi-screen virtual reality and/or stereo displays.

In replication mode each render node loads the full visible geometry, and each node is configured to render a different portion of the display. Replication has the absolute highest possible frame rate for distributed rendering, because the rendering performed by each node is completely independent.

Pros:

- highest frame-rate option
- unlimited resolution for multi-screen walls and VR

Cons:

- not scalable in compute or rendering performance
- aggregate memory requirement are higher than for compositing

Replication with Compositing

Replication can be combined with compositing when the features of replication and the performance of compositing are required. As in the basic replication scenario, multiple render nodes are configured to render to multiple screens. For each 'N' screens, some number of additional render nodes 'M' are designated to increase performance and lower the per-node memory requirements. The total amount of replicated data is the same as in the basic replicated case - the data is replicated N times, but each copy is distributed over M nodes, for an effective memory requirement of 1/M per node.

Pros:

- high polygon rates
- scalable client memory and computation

Cons:

- higher hardware costs (M x N nodes)

Configuring EnSight DR

Distributed rendering requires the use of the collaboration hub to coordinate the efforts of multiple EnSight clients, whether they run on one machine or many. When using a DR mode that requires data decomposition, the SOS (server-of-servers) is required and performs this service. SOS is not required in 'replicate' mode because all clients receive all of the data.

While there are many individual processes involved, starting EnSight for parallel rendering can be achieved automatically using a few configuration files. Depending on the rendering mode, either a 'prdist' or a 'dconfig' file specifies the parameters for the individual worker clients, as shown in the following sections. The example configuration files from this section can be found in $CEI_HOME/ensight92/doc/ in the prdist and dconfig subdirectories. EnSight resources and command line options can be used in place of or in conjunction with ‘prdist’ files.

Note that EnSight DR requires that a cluster be configured so that remote launching works with correct X11 permissions. Before trying any of the examples, please see the tips at the end of this chapter.

Compositor configuration

The compositor and Chromium modes share a configuration file format, referred to as the *prdist* file. The prdist file begins with an identifying string:

```
ParallelRender EnSight 1.0
```

This line is followed by configuration information for the collaboration hub and the rendering nodes:

```
router [hostname] [options]
```

where `[hostname]` is either a network routable hostname or IP address, or `hostname` is `‘MANUAL’` which indicates that the user will start the EnSight Collabhub (`ensight92.collabhub`) manually. Additionally, `hostname` can be left blank when using resources. In this case the options will apply to the resource host.

And where [options] may include:

```
Executable=”/path/to/ensight92.collabhub <cmdline>”
Directory=”/start/directory”
AltLoginId=”UserId”
RSH=”rsh”
ConnectBackToHost=””
UsingSSH=”FALSE”
OutputPrefix=”/dev/null”
```

```
client [hostname] [options]
```

where [hostname] is either a network routable hostname or IP address, or hostname can be left blank when using resources. In this case the options will apply to the resource hosts. Furthermore, if left blank for resource defaults, only the first client line will be used. The number of resource hosts specified will dictate the number of rendering clients used.

And where [options] may include:

```
Executable="/path/to/ensight92.client"
Directory="/start/directory"
AltLoginId="UserId"
RSH="rsh"
ConnectBackToHost=""
UsingSSH="FALSE"
Display=":0.0"
```

Example 1

The example file prdist.template contains an annotated example configuration file for cluster rendering. This is a fallback example - it does not use compositing or Chromium, it simply illustrates how to start up multiple clients in parallel. The functional parts of the file are:

```
ParallelRender EnSight 1.0
router localhost
client localhost
client localhost
```

To test this configuration run:

```
ensight92 -prdist prdist.template
```

This command will start 3 clients, the "Queen Bee" client for user input and interaction and two "Worker" clients that load data and perform rendering. The Queen Bee client displays only bounding boxes of the data.

This is a simple test that verifies the installation and user environment. Once this test is working, you can make a new copy of this file and test the same setup on a cluster. Simply change the two worker hosts to the names of two hosts in the cluster, for example:

```
client node1
client node2
```

This describes exactly the same configuration, except that the three clients will all be running on different machines. On each of the worker clients you will see *half* of the geometry. This illustrates what EnSight is actually doing - the data computed by the servers is being load-balanced to the clients for rendering in parallel. Using EnSight by itself, you see multiple images with partial renderings.

Compositing

Compositing is enabled by adding a line to the prdist file after the router:

```
pc [options]
```

where `[options]` may include:

```
guicomposite = "NONE" "NOCOMP" "COMP"
compression = "NONE" "LOW" "MED" "HIGH" "AUTO"
guicompression = "{RAW|RLE|GZ} {final quality}
{interactive quality}" example: "RLE 0 2"
offscreen = "TRUE" "FALSE"
host = "hostname"
```

guicomposite

The parallel compositor (PC) needs to make a full set of TCP/IP connections between all of the render processes to exchange data between them. The guicomposite options provide the user some level of control over what machines require full TCP/IP connectivity. This option might be used to configure DR around a port firewall or to compensate for asymmetric bandwidth issues specific to a given site. The master client (often called the GUI-client) is the EnSight client that actually displays a GUI and with which the user interacts. The guicomposite option has three possible values that determine the extent to which the master client participates in the compositing.

- COMP: If the client is physically running on the same cluster as the other rendering nodes, use guicomposite="COMP". This causes PC to use the master node as part of the composite and as a side effect, leave the final image on that node. It is the most efficient way to get an image from PC to the screen, but the master node needs to be on the same high-performance network as the other render nodes for this to work well as it assumes symmetric bandwidth between the renderer clients and the master client. "TRUE" is the same as "COMP".
- NOCOMP: The next level of performance is guicomposite="NOCOMP". In this mode, the master still makes TCP/IP connections to each of the renderer nodes, but it does not participate in the actual compositing. The various pieces of the final image are sent back to the master node in parallel over the N sockets to the render nodes. This mode works well if the master node is not located on the cluster interconnect, but can still make N connections to the cluster over a fairly high performance interface. (e.g. the asymmetric bandwidth situation). In practice, few people use this mode as the number of off-cluster TCP/IP connections to the master can be a problem for things like firewalls.
- NONE: The last level is guicomposite="NONE". In this mode, the PC system runs entirely on the cluster and the final image is passed to the first renderer node. That node can optionally compress the image and then sends it via TCP/IP to the collabhub which then sends it to the master

client for display. This requires two network hops, but ensures that all of the TCP/IP connections of PC remain inside the cluster. It is the most firewall friendly mode as no new ports need to be opened up, but it is also the slowest mode.

compression

The `compression` option determines how the render clients communicate during compositing. If the rendered images are large and the CPUs are fast enough, enabling compression can increase the frame rate. These compression methods are all lossless. There are very few situations where the best value us *not* “HIGH”.

guicompression

The `guicompression` option only has an effect when `guicomposite` is set to `"NONE".` This option allows you to specify a compression level for images transmitted through the collaboration hub. This option is most useful when the main client is remote from the rendering cluster, perhaps over a lower-bandwidth and/or higher-latency network.

There are three compression methods each followed by two numbers: a final quality number and an interactive quality number. The higher the number, the lower the image quality. A quality value of 0 is lossless. The final number represents the quality of the still image when no more transforms are occuring, and the interactive number represents the quality of the image while undergoing transforms (e.g. rotation or translation). Often the interactive number is higher (lower quality) than the final to accelerate transformations.

RAW - uncompressed pixels. In this mode, the final and interactive quality numbers represent spatial decomposition (elimination of some spatial fraction of the pixels). The final and interactive quality numbers can range from 0 to 8, where 0=all pixels, 1=every other pixel, 2=every third pixel, etc. Visually, this effectively looks like larger pixels.

RLE - the evo RLE scheme, which has a nice balance of speed vs compression. The final and interactive quality numbers represent the number of lower-order bits of a color that are set to zero before the RLE scheme is applied. For example, the value 2 means that the color of a pixel is first reduced to 6 (8-2) bits per pixel before the compression. Therefore these quality numbers should ranges between 0 (highest quality) and 7 (lowest quality). Visually, this looks like quantized colors/ banding.

GZ - use the "gzip" algorithm is slower, but has good compression. The final and transition quality numbers work as in the RLE case.

offscreen

The `offscreen` option specifies that the render workers should create offscreen (pbuffer) rendering windows. By default they will be onscreen, which is useful for viewing and confirming the partial results. One advantage of using offscreen windows is that your render nodes do not have to run at the same resolution as your main client. Note: if guicompression = “NONE” and offscreen = “FALSE” then the first renderer node in the prdist file may not display any image. It is rendering and participating properly, however.

example

If the master client (where the GUI is) is on the cluster, use "COMP". In all other situations, use "NONE". You can try "NOCOMP", but it is not always a win. In house, we typically use the following pc line in our prdist files:

pc compression="HIGH" guicomposite="COMP" offscreen="TRUE" guicompression="RLE 0 2" (all on one line).

If the master client is not on the cluster, change "COMP" to "NONE". For 99% of the cases, that is the best setup.

Example 2

The example file prdist.pc2 is a sort-last rendering example in which the two EnSight clients each render their half of the geometry and the result is composited by the PC compositor and display in the EnSight GUI window. The functional lines of the file are:

```
ParallelRender EnSight 1.0
pc
client node1
client node2
```

.

See How To Use Resource Management for an example using resources

Data replication modes

In replication mode, unlike compositing mode, EnSight itself needs to know about the screen positions and sizes. The 'dconfig' file format used by EnSight for shared-memory parallel rendering includes all of this information, and has been extended to provide additional information for distributed-memory rendering

Replication mode

To modify a shared-memory dconfig file for distributed-memory rendering, it is only necessary to add a `'hostid'` line for each screen in the file. The example file ‘dconfig1’ describes a on-panel wall:

```
CVFd 1.0
display
wallresolution
   1280 1024
screen
   hostid node
   displayid :0.0
   resolution 1280 1024
   wallorigin 0 0
```

This mode does not necessarily require SoS, because each client will load the entire visible geometry.

Replication with compositing

As noted earlier, replication sacrifices memory by storing the entire client-side geometry and variables on each of the clients. For small to moderate sized geometry, this can result in very fast framerate with tiled display wall capability. For extremely large geometry, it is possible to be memory-, compute-, or render constrained on the individual nodes. With additional hardware and one more line in the dconfig file, it is possible to add a compositing step, very similar to the single-screen compositing described in the previous section. Example file 'dconfig2' extends the previous example to distribute the rendering among four nodes:

```
CVFd 1.0
display
wallresolution
   1280 1024
screen
   hostid node1
   displayid :0.0
   resolution 1280 1024
   wallorigin 0 0
   workerhosts "node2 :0.0 node3 :0.0 node4 :0.0"
```

In a multi-screen configuration, it is required that each display node has an equal number of "worker" nodes for the compositing step.

TIPS FOR DISTRIBUTED MEMORY RENDERING

1. The rendering nodes must have OpenGL enabled X11 servers running on them and they must be configured for direct rendering by remotely executed user processes. To test this, log into a rendering node remotely and run "glxgears -display :0". If this does not bring up glxgears on the display, the X11 server will need to be reconfigured.
2. You may need to add options like '-ac' to the /etc/X11/xmd/Xservers file or otherwise configure X11 authorization mechanisms (we know it will run with "-ac -s 0", but site security implications must be carefully considered). It may be useful to check that the rendering is "direct" as well (glxgears should run > 8000 fps on modern graphics cards with proper hardware accelerated OpenGL drivers). The output of glxinfo on the rendering node can be helpful in diagnosing issues as well. You may need to modify the permissions of the /dev/nvidiactl file(s) to allow a non-console app to access the graphics system.
3. All X11 screen savers and blanking functions should be turned off on the rendering nodes for your PC to work properly.
4. The cluster needs to have passwordless ssh or rsh set up on it.
5. You need to be able to run the command:
   `ssh <hostname> /usr/X11R6/bin/glxgears -display :0.0`
   (rsh will work as well as ssh if you have it configured).
6. If your system prompts for a password when launching remote programs via

ssh or rsh, EnSight DR will not run properly.

7. If using ssh, make sure all the interactive prompts for node identification and authorization between rendering nodes have been suppressed for initial connections. This should be done for all possible connections between the rendering nodes. If you have any firewalls set up inside of the cluster (or between the rendering and interactive nodes), the firewall will need to leave ports 8739 to 8789 open for EnSight's own internal TCP/IP connections which do not utilize the rsh/ssh connections.
8. Set the DISPLAY environmental variable to ':0.0', set CEI_HOME to the proper path and put $CEI_HOME/bin in the search path for remote connections (rsh or ssh). For example, you should be able to run the following without any errors or interactive prompts:
   `ssh <hostname> cei_apex21_arch`
9. Many clusters have multiple TPC/IP address and hostnames for each node. Usually, one for the high-performance interconnect (e.g. InfiniBand) and one for administration. For the highest performance EnSight needs to use the TCP/IP addresses associated with the high-performance interconnect.

   A cluster configured to use the highest performance interconnect as its default is the simplest to configure for use with EnSight. For example, since by default EnSight will use the TCP/IP address resolved by 'hostname', this name should be the highest performance interconnect.

   It is possible to run EnSight DR on a secondary TCP/IP interface. To do this, add the following option:

   `ConnectBackToHost="guihostname"` to the "router" line
   `Host="guihostname"` to the "pc" line
   `ConnectBackToHost="collabhubhostname"` to each "client" line.

   All the other hostnames should also use the higher performance interconnect names.

Once the above are working on your cluster, contact CEI Support if you run into any problems running EnSight on it. Please send us your cluster(s) details and the config files you have tried.

Some general advice:

To simplify debugging, start small and scale up.

For example, we have seen problems with some TCP/IP over IB implementations, especially at scale, so start small (2-3 nodes) and if you have gigE in addition to IB, try that as well. Also, implement DR with all the EnSight processes running on rendering nodes of a single cluster. Configure one rendering node on the cluster as an "interactive" node (with mouse/keyboard).

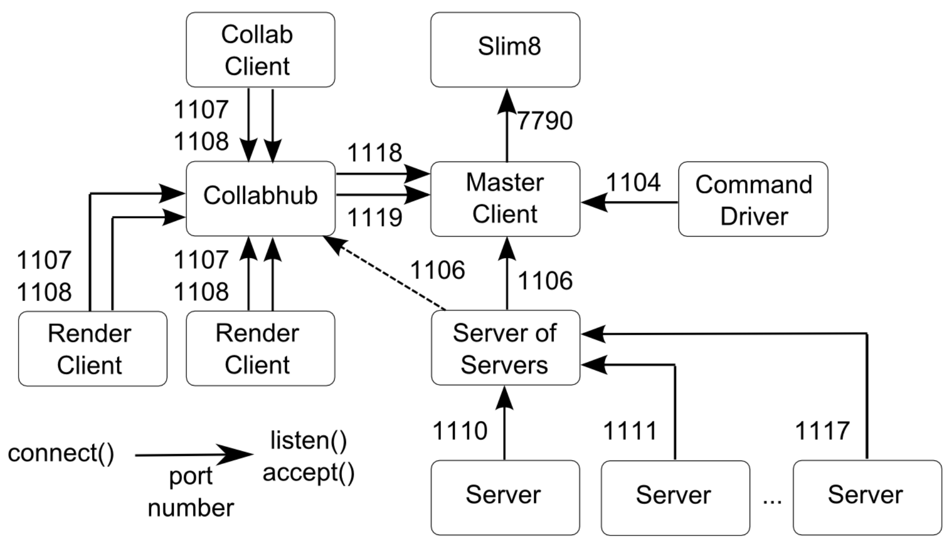
Some of these items can be simplified using prologue - epilogue scripts that can be used to do things like run the X11 servers as the user and with custom configs, which can eliminate a number of the potential issues in step (2). In step (3), we can use any script you provide instead of 'ssh' as long as it has the same API/semantics as rsh/ssh. We only use that system to launch executables on the remote systems.

13.3 EnSight Networking Considerations

The EnSight application suite consists of a number of different applications (Client, Server, Collabhub, Server of Servers, etc) that communicate over TCP/IP (socket) connections. The collection of connections are fairly complex and in secure environments, the specifics of those connections are critical for advanced client/server usage of EnSight with remote components. In this section, we document the various connections that can be made by the suite so that specific firewall and VPN/tunnel configurations can be customized for use with EnSight.

Default TCP/IP Port Usage

The following diagram documents all of the TCP/IP connections that EnSight can make as well as the default port numbers. In normal EnSight operation, the only connections are between Slim8 (the license manager), the Master Client and a single Server (which replaces the Server of Servers box in the diagram). EnSight uses rsh or ssh to launch the various processes, but it does not use the rsh/ssh channel for communication. Instead, the various processes make their own TCP/IP connections after they have been launched. The arrows in the diagram demonstrate the direction of the connection. The process at the arrow end of the lines is making the accept() socket call while the other end is making the connect() socket call. After the connection has been established, all communication is bi-directional over the various socket connections.



By default, EnSight may use any of the ports in the range 1104 to 1119 (excepting 1105 and 1109) and port 7790 for bi-directional communication. Command line options exist to change all of the default ports. It should be noted that these connections may be within the same machine as ports usage has been constructed to make it possible to run the entire system inside of a single computer.

At least one 'Client' and one 'Server' are necessary to run EnSight. The 'Server of Server' (SOS) looks to EnSight as a Server (it uses the same protocols and ports as a Server does to talk to the client). In standard collaboration mode, the "Master Client" launches a 'Collabhub' process and one or more external EnSight Clients (shown in the diagram as 'Collab Client's) can attach to the 'Collabhub'. In EnSight DR mode, there is a 'Collabhub' and one or more 'Render Client's (and there can be no ‘Collab Client’ components). The ‘Command Driver’is a seldom used component, but is included here for the sake of completeness.

One connection not in this diagram is the connection between a Server and the Master Client when the Server of Servers (SOS) is not being used. In this situation, the server connects to the Master client over port 1106, using exactly the same mechanism as the SOS uses to connect to the Master Client.

Connections between Clients and the Collabhub use two ports as data and control channels. These connections are always made in the same direction, with the connection from the Collabhub to the Master Client in the opposite direction than the other Clients.

When running with the Server of Servers, the Servers may use ports from 1110 to 1117 depending on the number of threads specified for the SOS (up to 8 threads maximum). For example if ENSIGHT9_MAX_SOSTHREADS is set to 2, then ports 1110 and 1111 are used. These ports are re-used if more than two Servers are specified. If ENSIGHT9_MAX_SOSTHREADS is set to 8 then ports 1110 to 1117 are used (and reused if more than 8 Servers is specified). By default, a single thread and only port 1110 is used to establish all of the SOS to Server connections.

Initially, the Server (or SOS) attaches to the client on port 1106. If a Collabhub is introduced (in Collaboration Mode or with DR), the SOS or Server moves its connection to 1106 over to the collabhub.

ENSIGHT ® VERSION 9 END USER LICENSE AGREEMENT

UNLESS A SEPARATE LICENSE OR TRIAL AGREEMENT DOCUMENT EXISTS BETWEEN THE LICENSEE AND CEI OR AN AUTHORIZED CEI DISTRIBUTOR, THE TERMS AND CONDITIONS OF THIS AGREEMENT SHALL GOVERN YOUR USE OF VERSION 9 OF THE EnSight SOFTWARE. READ THIS LICENSE CAREFULLY BEFORE USING THE EnSight SOFTWARE. BY USING THE EnSight SOFTWARE YOU AGREE TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT ACCEPT OR AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE EnSight SOFTWARE UNUSED WITHIN THIRTY (30) DAYS OF PURCHASE FOR A REFUND.

1. LICENSE GRANT. The Licensee is hereby granted by Computational Engineering International, Inc. ("CEI") a single, non-transferable and non-exclusive license to use the EnSight Software under the terms and conditions set forth below, in binary form only, and at the major release level indicated on the media used to deliver the licensed EnSight Software.

2. DEFINITIONS.

   "Ancillary Software" includes translators, user defined data readers, tools, or other software which may from time to time be delivered with, but separate from, the "EnSight Software".

   "End User" means one individual running an EnSight process.

   "EnSight Purchase Agreement" means the document provided by Licensee to CEI which describes the type of installation Licensee is authorized to make.

   "EnSight Documentation" means manuals, release or installation notes related to the EnSight Software, including electronic versions thereof.

   "EnSight Product" means the computer program specified in the EnSight Purchase Agreement licensed by the user and authorized for installation named EnSight, EnSight Gold, EnSight DR, EnSight Lite, EnSight CFD, or any other product which is now available or may become available from CEI with the same family name.

   "EnSight Software" means all of the CEI computer programs that constitute the EnSight Product, the EnSight Documentation, and any backups or other copies.

   "Maximum Seats" means the maximum authorized number of End Users who may simultaneously use the licensed copy of the EnSight Software as identified in the EnSight Purchase Agreement.

   "Major Release Level" means all versions of the EnSight Software which are denoted by the same integer number to the left of the first decimal point in the specification of the release. For example, all versions denoted as EnSight 8.x are regarded as the same Major Release Level.

3. USAGE LIMITATIONS. Licensee acknowledges that the EnSight Software is proprietary and shall remain the property of CEI. This License is not a sale. Title to and ownership of the EnSight Software and any copies thereof, any patents, trademarks or copyrights incorporated therein and to the CEI Proprietary information shall at all times remain with CEI. No rights to rent, loan, transfer, relicense, distribute, or otherwise assign any or all of Licensee's rights in the EnSight Software are granted. All rights not specifically granted to Licensee by this license shall remain in CEI. CEI may include features in the EnSight Software which prevent unlicensed use or use after license expiration. The license granted herein does not include provision of support and maintenance service or future releases or versions of the EnSight Software.

   The Ancillary Software may contain technology licensed to CEI from a technology partner. Licensee acknowledges that the Ancillary Software is proprietary and shall remain the property of CEI and technology partners. Title to and ownership of the Ancillary Software and any copies thereof, any patents, trademarks, or copyrights incorporated therein shall at all times remain with CEI and it's technology partners as applicable. Licensee is authorized to use the Ancillary Software only with CEI products.

4. INSTALLATION. EnSight Software may be installed and used by the Licensee only in the manner indicated in the EnSight Purchase Agreement. End User may use EnSight Software only at his immediate workplace, unless otherwise authorized by CEI. Subject to request of and approval by CEI, Licensee may permanently or temporarily change the systems and /or networks upon which Licensee is authorized to use the EnSight Software. In either case, such change may result in additional charges to the Licensee. Licensee may duplicate the EnSight Software for backup, archiving, or security, provided that each copy includes all of the copyright or proprietary notices of the original. Licensee shall not clone, reverse assemble, or reverse compile any part of the EnSight Software or adopt any part of the EnSight Software as its own. CEI reserves the right to charge an upgrade fee for future Major Release versions of the EnSight Software.

5. PATENTS AND COPYRIGHTS. EnSight Software is copyrighted under the laws of the United States and international treaty provisions. Notwithstanding the copyright, the EnSight Software contains trade secrets and proprietary information of CEI. Licensee acknowledges that CEI owns these copyrights and has the following exclusive rights with regard to the EnSight Software: to reproduce it; to adapt, transform or rearrange it; to prepare derivative works from it; and to control its distribution. Licensee agrees not to act in contravention of any

of CEI's intellectual property rights.

6. User defined data readers delivered with the Ancillary Software may contain licensed software technology from Visual Kinematics Incorporated ("VKI"). This data reader technology is proprietary and copyrighted containing confidential trade secret information which remains the sole property of VKI. CEI warrants that the End User is authorized to use these readers together with CEI products without further license requirements.

7. EnSight DOCUMENTATION. Licensee may use the EnSight Documentation only in support of its use of the licensed EnSight Software and may print or duplicate the Documentation (except any marked as proprietary), but only for its internal use and provided that each copy includes all of the copyright or related notices of the original.

8. WARRANTY; DISCLAIMER. CEI warrants that for a period of thirty (30) days after delivery of the EnSight Software, it will substantially conform in all material respects to the specifications set forth in CEI's published EnSight Documentation. If the EnSight Software does not meet this requirement, and Licensee notifies CEI within thirty (30) days after delivery, CEI will, at its option, repair or replace the effected EnSight Software. This warranty excludes problems caused by acts of Licensee, unauthorized use of or causes external to the EnSight Software, and items which would be covered under Support and Maintenance service. CEI specifically does not warrant, guarantee, or make any representations regarding the use, or the results of the use, of the EnSight Software or Documentation in terms of correctness, accuracy, reliability, currentness, or otherwise. The entire risk as to the result produced by and the performance of the EnSight Software is assumed by the End User. CEI does not warrant that the execution of the EnSight Software will be uninterrupted or error free.

   CEI warrants the ancillary software according to the terms found in the LICENSE_AND_WARRANTY_AND_SUPPORT file of the src directory of the software installation.

   THE ABOVE WARRANTIES ARE CEI'S ONLY WARRANTIES AND ARE IN LIEU OF ALL IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND THE ABOVE REMEDIES ARE THE EXCLUSIVE REMEDIES FOR ANY BREACH OF WARRANTY. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY CEI OR ITS EMPLOYEES SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY, AND LICENSEE MAY NOT RELY ON SUCH INFORMATION OR ADVICE. LICENSEE'S SOLE REMEDY AND CEI'S SOLE OBLIGATION SHALL BE GOVERNED BY THIS AGREEMENT. LICENSEE EXPRESSLY AGREES THAT IN NO EVENT SHALL CEI BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, OR SPECIAL DAMAGES ARISING FROM THE BREACH OF WARRANTY, NEGLIGENCE, OR ANY OTHER LEGAL THEORY, WHETHER IN TORT OR CONTRACT, EVEN IF CEI HAS BEEN APPRISED OF THE LIKELIHOOD OF SUCH DAMAGES OCCURRING, INCLUDING WITHOUT LIMITATION DAMAGES FROM INTERRUPTION OF BUSINESS, LOSS OF PROFITS OR BUSINESS OPPORTUNITIES, LOSS OF USE OF SOFTWARE, LOSS OF DATA, COST OF RECREATING DATA, COST OF CAPITAL, COST OF ANY SUBSTITUTE PRODUCTS, OR LOSSES CAUSED BY DELAY. FURTHER, CEI SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES OR EXPENSES RESULTING FROM ALTERATION OR UNAUTHORIZED USE OF THE EnSight SOFTWARE OR FROM THE UNINTENDED AND UNFORESEEN RESULTS OBTAINED BY LICENSEE AND RESULTING FROM SUCH USE. IN ANY EVENT, CEI'S MAXIMUM LIABILITY UNDER ANY LICENSE OR OTHERWISE FOR ANY REASON SHALL NOT EXCEED THE TOTAL AMOUNT PAID BY LICENSEE FOR THE CEI EnSight SOFTWARE, EXCLUSIVE OF ANY SUPPORT AND MAINTENANCE FEES PAID. THE EXISTENCE OF MORE THAN ONE CLAIM WILL NOT ENLARGE OR EXTEND THE LIMIT.

9. LICENSE MANAGER SOFTWARE. If Licensee is authorized to install and use License Manager Software to enable the EnSight Software to execute on multiple computer systems on a network, it will be so noted in the EnSight Purchase Agreement. All terms related to the EnSight Software, including and without limitation, terms related to limitation of remedies and restrictions on use, apply to the License Manager Software. Licensees authorized to use License Manager Software may install multiple redundant license servers on the same network subnet node. Installation of multiple redundant license servers on different subnet nodes requires Licensee to submit a written request to CEI, stating that the use of multiple servers will not enable more than the maximum seats of the software to be used. Furthermore, Licensee agrees to submit License Manager log files upon request to CEI for verification. Use of EnSight on a network is restricted to one geographic location, unless otherwise authorized by CEI.

10. U. S. GOVERNMENT RESTRICTED RIGHTS. If the Licensee is the USA Department of Defense ("DOD"), the licensed EnSight Software is subject to "Restricted Rights" as that term is defined in the DOD Supplement to the Federal Acquisition Regulations ("DFARS") section 252.227-7013(c). If the Licensee is any unit or agency of the U.S. Government other than the DOD, the Government's rights in the EnSight Software will be defined in paragra ph 52.227-19(c)(2) of the Federal Acquisition Regulations ("FAR"). Use, duplication, reproduction, or disclosure by the U.S. Government

is subject to such restrictions. Contractor/ Manufacturer is: Computational Engineering International, Inc., 2166 N. Salem Street, Suite 101, Apex, NC 27523, USA.

11. INVOICES; PAYMENT; TAXES. Licensee agrees to pay the License Fee(s) owed for this Software License as shown on a separate Invoice(s) from CEI. All Payments will be in U. S. Dollars. Payment is due within thirty (30) days after the date of each Invoice. Payment is considered made when good funds are received by CEI. To those Licensees leasing an EnSight Software License, future Invoices for the EnSight Software Lease Fee will be issued yearly in the month prior to that in which the initial payment of the EnSight Software Lease Fee was made to CEI. Licensees located in the State of North Carolina, USA are responsible for payment to CEI of all applicable State and Local sales taxes. Licensees located in any State other than North Carolina in the USA are directly responsible for payment of all State and Local sales taxes applicable at their location. Licensees located in any Country other than the USA are directly responsible for payment of all import and sales taxes to the responsible government(s) at their location. Licensees who have obtained a License through an authorized Distributor of CEI agree to comply with the Distributor's payment terms and conditions.

12. SEVERABILITY. If any provision of this Agreement shall be held by a court of competent jurisdiction to be contrary to law or not enforceable, the remaining provisions of this Agreement shall remain in full force and effect.

13. EXPORT. This License and the rights granted hereunder are subject to compliance by Licensee with all laws, regulations, orders, or other regulations relative to export or redistribution of Licensed Software that may now or in the future be imposed by the government of the United States or any agency thereof or of any other country into which licensed EnSight Software may be transported and any act of noncompliance shall immediately terminate this License. If the Licensee imports the licensed copy of the EnSight Software into a country in which CEI has an authorized Distributor, then if the License was purchased the Licensee shall pay to CEI a Penalty Fee equal to one half the cost of the original Purchased License Fee.

14. TERM. If a License for the EnSight Software was purchased by Licensee, then this license is valid upon the date of initial payment of the Purchase License Fee to CEI by Licensee and shall continue until an event of Termination. If a License for the EnSight Software was leased by Licensee, then this license is valid for twelve (12) months from the date of initial payment of the Lease License Fee to CEI by Licensee and shall renew for subsequent twelve (12) month periods from the date Licensee pays the amounts indicated on each yearly invoice, according to Section11, unless an event of Termination occurs.

15. TERMINATION. This Software License Agreement shall terminate upon occurrence of any of the following events: (i) the failure of Licensee to observe or perform any of the material covenants, terms, and conditions of this Agreement where such nonperformance is not fully remedied by Licensee within thirty (30) days after written notice by CEI; (ii) any breach of Sections 3, 4, 6, 7, 8, 9, or 10 hereof (effective immediately); (iii) the filing of a petition for Licensee's bankruptcy, whether voluntary or involuntary, or an assignment of Licensee's assets made for the benefit of creditors, or the appointment of a trustee or receiver to take charge of the business of Licensee for any reason, or Licensee's becoming insolvent or voluntarily or involuntarily being dissolved; or (iv) the use of a license purchased at an academic discount for purposes other than teaching or academic research. Termination of this Agreement under Section 15 shall be in addition to, and not a waiver of, any remedy at law or in equity. Notwithstanding the foregoing, the provisions of Sections 4, 6, 7, 8, 9, and 10 shall survive the termination of this agreement. Upon termination of this Software License Agreement, Licensee shall promptly cease to use and shall return to CEI all copies of the EnSight Software, the Ancillary Software, and EnSight Documentation.

16. GENERAL. Licensee may not assign or transfer its rights or obligations under this Agreement without the prior written consent of CEI. The parties agree that this Agreement shall be governed and construed by the laws of the State of North Carolina, USA, and that no conflict-of-laws provision shall be invoked to permit the laws of any other state or jurisdiction. Any legal action must be filed within one (1) year after the cause for such action arises with the court of jurisdiction in the State of North Carolina, USA. All preprinted additional or different terms on any purchase order forms or other documents received from Licensee are deemed deleted and Licensee agrees that such terms shall be void even if the Licensee's documentation indicates the terms therein take precedence over other documents. This Software License Agreement, together with the most recent EnSight Purchase Agreement delivered to CEI by Licensee, constitutes the entire agreement of the parties and supersedes any prior understandings relating to the subject matter, and may be amended or supplemented only in a written agreement signed by both an officer of CEI and the Licensee.

17. RESTRICTION ON USE OF TrackdAPI SOFTWARE, developed and owned by VRCO Inc., an Delaware corporation, with principal place of business at 192 Ballard Ct. Suite 300, Virginia Beach, VA 23462; here after referred to as "VRCO". Licensee is prohibited from the distribution, transfer, modification, or alteration of the TrackdAPI software and associated written materials and/or

documentation (“TrackdAPI Software”) and shall abide by the following:

I. PROPRIETARY RIGHTS. Licensee agrees that its use of TrackdAPI Software is a license only and VRCO owns all right, title, and interest in the TrackdAPI Software including any patents, trademarks, trade names, inventions, copyrights, know-how and trade secrets relating to the design, manufacture, operation or service of the TrackdAPI Software. Nothing in this agreement should be construed as transferring any aspects of such rights to Licensee or any third party.

II. WARRANTY DISCLAIMER. THE LICENSED TrackdAPI Software IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LICENSED TrackdAPI Software IS ON THE LICENSEE.

III. LIMITATION OF LIABILITY. IN NO EVENT SHALL VRCO BE LIABLE FOR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES OR LOST PROFITS ARISING OUT OF THIS AGREEMENT OR USE OF THE TrackdAPI Software, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY, AND EVEN IF VRCO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. LICENSEE AGREES VRCO’S LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED ANY AMOUNTS PAID BY LICENSEE FOR THE TrackdAPI Software IDENTIFIED ABOVE.

18. LICENSED USE OF SIEMENS PARASOLID CAD READER SOFTWARE

EnSight customers may license CAD READERS from CEI. Some CAD READERS require Siemens' Parasolid Communicator product, hereafter SIEMENS CAD READERS. Siemens requires such SIEMENS CAD READER customers to be informed of the license requirements, which are:

I. Authorization for use of Parasolid Products in the Territory to process only data associated with Licensee's own business, including providing professional services for Licensee's customers, but only while embedded within EnSight.

II. PROPRIETARY RIGHTS. Licensee agrees that its use of SIEMENS CAD READERS is a license only and Siemens owns all right, title, and interest in the SIEMENS CAD READERS including any patents, trademarks, trade names, inventions, copyrights, know-how and trade secrets relating to the design, manufacture, operation or service of the SPATIAL CAD READERS. Nothing in this agreement should be construed as transferring any aspects of such rights to Licensee or any third party.

III. Prohibition against reverse engineering the SIEMENS CAD READERS and creating any derivative works, compilations or collective works thereof;

IV. LIMITATION OF LIABILITY. IN NO EVENT SHALL SIEMENS BE LIABLE FOR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES OR LOST PROFITS ARISING OUT OF THIS AGREEMENT OR USE OF THE SIEMENS CAD READERS, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY, AND EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. LICENSEE AGREES SIEMENS LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED ANY AMOUNTS PAID BY LICENSEE FOR THE SIEMENS SOFTWARE IDENTIFIED ABOVE.

19. LICENSED USE OF SPATIAL CORP. 3D INTEROP CAD READER SOFTWARE

EnSight customers may license CAD READERS from CEI. Such customers are hereby informed that such software is in part licensed from SPATIAL Corporation 3D Interopt product hereafter SPATIAL CAD READERS . Spatial requires SPATIAL CAD READER customers to be informed of the license requirements, which are:

Hereafter referred to as "SPATIAL CAD READERS". Licensee is prohibited from the distribution, transfer, modification, or alteration of the SPATIAL CAD READERS software and associated written materials and/or documentation ("SPATIAL CAD READERS") and shall abide by the following:

I. PROPRIETARY RIGHTS. Licensee agrees that its use of SPATIAL CAD READERS is a license only and SPATIAL Corp owns all right, title, and interest in the SPATIAL CAD READERS including any patents, trademarks, trade names, inventions, copyrights, know-how and trade secrets relating to the design, manufacture, operation or service of the SPATIAL CAD READERS. Nothing in this agreement should be construed as transferring any aspects of such rights to Licensee or any third party.

II. WARRANTY DISCLAIMER. THE LICENSED SPATIAL CAD READERS Software IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE LICENSED SPATIAL CAD READERS IS ON THE LICENSEE.

III. LIMITATION OF LIABILITY. IN NO EVENT SHALL SPATIAL BE LIABLE FOR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES OR LOST PROFITS ARISING OUT OF THIS AGREEMENT OR USE OF THE SPATIAL CAD READERS, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY, AND EVEN IF SPATIAL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. LICENSEE AGREES SPATIAL'S LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED ANY AMOUNTS PAID BY LICENSEE FOR THE SPATIAL SOFTWARE IDENTIFIED ABOVE.

20. LICENSED USE OF JT OPEN

EnSight customers may license the JT Export from CEI. Such customers are hereby informed that such software is in part licensed from Siemens Corporation, hereafter SIEMENS JT EXPORT. Siemens reserves rights and restrictions to the use of SIEMENS JT EXPORT software.

I. Authorization for use of SIEMENS JT EXPORT in the Territory to process only data associated with Licensee's own business, including providing professional services for Licensee's customers, but only while embedded within EnSight.

II. PROPRIETARY RIGHTS. Licensee agrees that its use of SIEMENS JT EXPORT is a license only and Siemens owns all right, title, and interest in the SIEMENS JT EXPORT including any patents, trademarks, trade names, inventions, copyrights, know-how and trade secrets relating to the design, manufacture, operation or service of the SIEMENS JT EXPORT . Nothing in this agreement should be construed as transferring any aspects of such rights to Licensee or any third party.

III. Prohibition against reverse engineering the SIEMENS JT EXPORT and creating any derivative works, compilations or collective works thereof;

IV. LIMITATION OF LIABILITY. IN NO EVENT SHALL SIEMENS BE LIABLE FOR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES OR LOST PROFITS ARISING OUT OF THIS AGREEMENT OR USE OF THE SIEMENS JT EXPORT , HOWEVER CAUSED, ON ANY THEORY OF LIABILITY, AND EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. LICENSEE AGREES SIEMENS LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED ANY AMOUNTS PAID BY LICENSEE FOR THE SIEMENS SOFTWARE IDENTIFIED ABOVE.

EnSight® Version 9 Support and Maintenance Service Agreement

UNLESS A SEPARATE AGREEMENT FOR SUPPORT AND MAINTENANCE SERVICES EXISTS BETWEEN THE LICENSEE AND CEI OR AN AUTHORIZED CEI DISTRIBUTOR, THE TERMS AND CONDITIONS OF THIS AGREEMENT SHALL GOVERN CEI'S PROVISION OF AND LICENSEE'S USE OF AND PAYMENT FOR SUCH SERVICES.

1. GENERAL. The EnSight Software Licensee and Computational Engineering International, Inc., (“CEI”) agree that the following terms and conditions apply to the provision of Software Support and Maintenance Service.

2. DEFINITIONS.

   “Ancillary Software” includes translators, user defined data readers, tools, or other software which may from time to time be delivered with, but separate from, the “EnSight Software”.

   “EnSight Purchase Agreement” means the document provided by Licensee to CEI which describes the type of installation Licensee is authorized to make.

   “EnSight Documentation” means manuals, release or installation notes related to the EnSight Software including electronic versions thereof.

   "EnSight Product" means the computer program specified in the EnSight Purchase Agreement licensed by the user and authorized for installation named EnSight, EnSight Gold, EnSight DR, EnSight Lite, EnSight CFD, or any other product which is now available or may become available from CEI with the same family name.

   "EnSight Software" means all of the CEI computer programs that constitute the EnSight Product, the EnSight Documentation, and any backups or other copies.

   “Major Release Level” means all versions of the EnSight Software which are denoted by the same integer number to the left of the first decimal point in the specification of the release.

3. SOFTWARE SUPPORT SERVICE. During the term of this Agreement CEI will provide technical support as described below to the EnSight Software Licensee in its use of the EnSight Software, provided that Licensee has an active License to use the EnSight Software from CEI.

   Exception for Academic Campus Licensees: Support is provided to 3 named individuals specified in the license profile. All other software users of the Campus License will receive support via these individuals. CEI recommends the campus IT department or main department users experienced with EnSight be named as these indivuals. These named individuals can be updated each semester.

   CEI will provide telephone consultation on problems encountered in using the EnSight Software during the Prime Time period defined as Monday through Friday, from 8:00 AM to 5:00 PM, USA Eastern Standard Time or Eastern Daylight Time, excluding U. S. Holidays. Licensee is limited to five (5) hours per month of telephone consultation. Alternatively, support requests can be made to CEI via FAX or Electronic mail.

   EnSight U.S. Support Telephone Number:(800) 551-4448
   Outside U.S. Support Telephone Number:(919) 363-0883
   EnSight Support FAX Number: (919) 363-0833
   EnSight Support Email address: support@ensight.com

4. SOFTWARE MAINTENANCE SERVICE. During the term of this Agreement CEI will maintain the EnSight Software provided that Licensee has an active License to use the EnSight Software from CEI. CEI will distribute to Licensee one copy of new minor releases of the major release listed on the media used to deliver the EnSight Software including EnSight Documentation as they become available, subject to the terms and conditions of this Agreement. CEI may modify the terms, conditions, and prices for services applicable to future major releases.

5. EXCLUDED SOFTWARE. NO SUPPORT OR MAINTENANCE SERVICE FOR THE ANCILLARY SOFTWARE IS OFFERED OR PROVIDED BY THIS AGREEMENT EXCEPT AS NOTED IN THE LICENSE_AND_WARRANTY_AND_SUPPORT FILE OF THE src DIRECTORY OF THE SOFTWARE INSTALLATION.

6. LICENSEE AGREEMENTS FOR SOFTWARE SUPPORT AND SOFTWARE MAINTENANCE. In order to permit CEI to supply the Software Support and Software Maintenance specified above:

   (i). If third party graphic libraries accompanied the EnSight Software delivered to Licensee, as noted in the EnSight Purchase Agreement, and such third party issues a new version of such library software which CEI incorporates into the Software, then Licensee must comply with any additional licensing or fee requirements imposed by such third party.

   (ii). Licensee agrees to install minor releases, fixes, circumventions, and corrective code to the EnSight Software in a reasonable time after receipt thereof.

   (iii). Licensee agrees to be responsible for the installation and administration of the EnSight Software.

   (iv). Upon request by CEI, Licensee will provide the name, address, telephone and FAX number, and Email address (if available) of the Licensee’s contact individual for communicating problems and solutions.

7. FEES. If Licensee is Leasing EnSight Software, then the cost of the Software Support and Maintenance Service is included in the annual EnSight Software License Lease Fee.

   If Licensee has Purchased an EnSight Software license, then there is no separate fee for Software Support and Maintenance Service for a period of twelve (12) months from the date of the initial payment of the EnSight Software License Purchase Fee to CEI. Licensee agrees to subsequently pay the annual EnSight Software Support and Maintenance Service Fee(s) for subsequent twelve (12) month periods at the then current rates when invoiced.

Licensees located in any State other than North Carolina in the USA are directly responsible for payment of all State and Local taxes applicable at their location for this Service. Licensees located in any Country other than the U.S. are directly responsible for payment of all taxes to the government(s) at their location for this Service. Licensees who have obtained a License through an authorized Distributor of CEI agree to comply with the Distributor's payment terms and conditions.

8. LIMITATION OF REMEDY AND DISCLAIMER. CEI WILL USE ITS DILIGENT EFFORTS TO PROVIDE THE SUPPORT AND MAINTENANCE SERVICES SPECIFIED HEREIN. CEI MAKES NO OTHER WARRANTY OF ANY KIND OR NATURE WITH REGARD TO THE SERVICES TO BE PERFORMED BY CEI UNDER THE TERMS OF THIS AGREEMENT AND ANY IMPLIED WARRANTIES, INCLUDING THE IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE AND MERCHANTABILITY, ARE HEREBY DISCLAIMED. THE REMEDIES SET FORTH IN THIS AGREEMENT ARE LICENSEE'S EXCLUSIVE REMEDIES FOR ANY BREACH OF THE TERMS OF THIS AGREEMENT. CEI WILL NOT BE LIABLE IN ANY EVENT FOR LOSS OF OR DAMAGE TO REVENUES, PROFITS, OTHER ECONOMIC LOSS OR GOODWILL OR OTHER CONSEQUENTIAL, SPECIAL, INCIDENTAL OR INDIRECT DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE PERFORMANCE OF ITS OBLIGATIONS HEREUNDER, INCLUDING ANY LIABILITY FOR NEGLIGENCE WITH RESPECT TO SERVICE PROVIDED UNDER THIS AGREEMENT EVEN IF CEI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY AND ALL CASES, CEI'S MAXIMUM LIABILITY IN CONNECTION WITH OR ARISING OUT OF THIS AGREEMENT SHALL NOT EXCEED THE EQUIVALENT OF ONE (1) YEAR OF CHARGES FOR THE RELEVANT SERVICE. THE EXISTENCE OF MORE THAN ONE CLAIM WILL NOT ENLARGE OR EXTEND THE LIMIT.

9. INVOICES AND PAYMENT. To those Licensees who have purchased an EnSight Software License, future Invoices for the EnSight Software Support and Maintenance Fee(s) will be issued yearly in the month prior to that in which the initial payment of the EnSight Software License Purchase Fee was made to CEI. Invoices shall be due and payable within thirty (30) days of date of invoice. All payments will be in U. S. Dollars.

   Payment of invoice is considered made when good funds are received by CEI.

10. OBSOLETE PRODUCTS. CEI will continue to provide Software Support Service for the release prior to the most current release for a period of twelve (12) months after the release date of the most current release. At that time, the previous release is designated as Obsolete and Software Support Service for the Obsolete release shall thereafter be discontinued.

11. TERM. The Term of this Agreement shall be from the date payment from Licensee is first received by CEI for the EnSight Software License Purchase Fee or the EnSight Software License Lease Fee, whichever is applicable, until an event of Termination.

12. TERMINATION. EnSight Software Support and Software Maintenance Service hereunder may be terminated as follows: (A) by Licensee or CEI to be effective during or after the first year of such Service upon giving ninety (90) days written notice; (B) by CEI with respect to Obsolete Products; (C) by CEI immediately and without notice, with respect to the EnSight Software, the license for which has expired or been terminated; (D) by CEI after ten (10) days subsequent to providing notice to Licensee upon either (i) nonpayment by Licensee of any Invoiced amount due under this Agreement or the EnSight End User License Agreement; or (ii) nonperformance by Licensee of any other material term or condition of this Agreement.

13. ASSIGNMENTS. The Licensee may not assign or transfer its rights or obligations under this Agreement, in whole or in part, without the written consent of CEI.

14. APPLICABLE LAW. The parties agree that this Agreement shall be governed and construed by the laws of the State of North Carolina, USA, and that no conflict-of-laws provision shall be invoked to permit the laws of any other state or jurisdiction. Any legal action must be filed within one (1) year after the cause for such action arises with the court of jurisdiction in the State of North Carolina, USA.

15. GENERAL. The terms and conditions stated in this Agreement constitute the complete and exclusive statement of the Agreement between Licensee and CEI and supersede all prior oral and written statements of any kind whatsoever made by either party or their representatives.

    All preprinted additional or different terms on any purchase order forms or other documents received from Licensee are deemed deleted and Licensee agrees that such terms shall be void even if the Licensee's documentation indicates the terms therein take precedence over other documents.

    Any waivers of or amendments to the terms and conditions of this Agreement, to be effective, must be in writing and signed by an officer of CEI and Licensee.

CEI Address:
Computational Engineering International, Incorporated
2166 N. Salem Street, Suite 101, Apex, North Carolina
27523
USA