

Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [X](#) [Y](#) [Z](#)

- **2DPlot**

The type of Field that contains a 1-dimensional GridNodeBasedDomain and a NumericalFieldRange. It is a Plot whose *PlotType* Property has a value of “2DPlot”. A special tool exists for visualizing 2DPlots. A Property can have a 2DPlot as its Value, if the 2DPlot’s DomainType is the SingleValuedDynamicEntity type expected for that Property type.

Related glossary terms: *Field*, *Plot*, *PlotType*, *GridNodeBasedDomain*, *NumericalFieldRange*, *SingleValuedDynamicEntity*

- **3DPlot**

The type of Field that contains a 2-dimensional GridNodeBasedDomain and a NumericalFieldRange. It is a Plot whose *PlotType* Property has a value of “3DPlot”.

Related glossary terms: *Field*, *Plot*, *PlotType*, *GridNodeBasedDomain*, *NumericalFieldRange*

- **4DPlot**

The type of Field that contains a 3-dimensional GridNodeBasedDomain and a NumericalFieldRange. It is a Plot whose *PlotType* Property has a value of “4DPlot”.

Related glossary terms: *Field*, *Plot*, *PlotType*, *GridNodeBasedDomain*, *NumericalFieldRange*

[Go to start of Glossary](#)

A

- **ABAQUS[®]**

A commercial simulation tool, manufactured by ABAQUS, Inc. (a subsidiary of Dassault Systemes), which Comet supports through its AbaqusTask. ABAQUS performs finite element analysis, particularly structural and thermal analysis. See http://www.hks.com/products/products_standard.html.

Related glossary terms: *AbaqusTask*, *external tool*

- **AbaqusTask**

A type of SimulationTask that manages runs of the ABAQUS/Standard external tool. It is a type of Comet entity found within a Process. “AbaqusTask” is an abbreviation of “AbaqusStandardTask-6.4”. See http://www.hks.com/products/products_standard.html.

Related glossary terms: *ABAQUS*, *SimulationTask*, *Process*, *Comet entity*

- ***abstract modeling***

Comet's mechanism for specifying entities of interest through an abstraction, such as a name or a characterization, rather than through explicit enumeration.

Related glossary terms: *AbstractDomain*, *AbstractDomainSet*, *physicalDomainFilterType*, *BoundEntitySet*, *TransferredEntitySet*, *GeometricClosure*, *EntireDomain*, *BoundaryDomain*, *EmbeddedDomain*, *InterfaceDomain*, *AbstractDomain membership tag*, *CAD annotation*, *ActiveTag*

- ***AbstractDomain***

A type of Comet entity that defines a Comet entity set of interest through an abstraction, such as a name or a characterization, rather than through explicit enumeration. An AbstractDomain specifies a filter to be applied to the entities within the PhysicalModel during a search. Currently, AbstractDomains can filter on entity type (the types that can be filtered on are Component, ComponentRepresentation, Interaction, RefVolume, Face, Edge and Vertex), domain name tags, and context of use. Comet tracks and dynamically updates which entities pass the filter at any point during a Comet session. Wherever a UEM instance needs to specify a set of entities, it can specify an AbstractDomain that binds to a set of entities of compatible type in place of an explicit enumeration of entities. For example, a PhysicalDomain can contain an AbstractDomain whose *physicalDomainFilterType* is a type of BasicTopologyEntity, instead of directly containing particular BasicTopologyEntities of that type. An AbstractDomain can be assigned *transferred entities*, which are dynamically bound to the members of the AbstractDomain's EntireDomain for the duration of their membership. Typical transferred entities are Properties and Sections. The entities to be transferred are contained in the EntireDomain's TransferredEntitySet.

Related glossary terms: *Comet entity*, *abstract modeling*, *physicalDomainFilterType*, *BoundEntitySet*, *TransferredEntitySet*, *PhysicalDomain*, *GeometricClosure*, *EntireDomain*, *BoundaryDomain*, *EmbeddedDomain*, *InterfaceDomain*, *AbstractDomain membership tag*, *CAD annotation*, *ActiveTag*

- ***AbstractDomainSet***

A top-level Comet entity, which contains the AbstractDomains defined for the UEM instance. Every Project has exactly one AbstractDomainSet.

Related glossary terms: *Comet entity*, *UEM instance*, *top-level entity*, *abstract modeling*, *AbstractDomain*, *physicalDomainFilterType*, *BoundEntitySet*, *TransferredEntitySet*, *GeometricClosure*, *EntireDomain*, *BoundaryDomain*, *EmbeddedDomain*, *InterfaceDomain*, *AbstractDomain membership tag*, *CAD annotation*, *ActiveTag*

- ***AbstractDomain membership tag***

A string-valued CAD annotation of type "COMET_DOMAIN_NAME", which is attached to an assembly, a part, a geometric entity, or a feature (vertex or axis). When an AbstractDomain membership tag is imported into Comet, it is represented within the UEM as an ActiveTag that is attached to the corresponding Comet entity (a Part if the CAD entity was a part, an Assembly if the CAD entity was an assembly, an Interaction if the CAD entity was a feature that was part of a set of Comet-specific

features and annotations that specify a LowFidelityInteraction, and otherwise a BasicTopologyEntity), named “COMET_DOMAIN_NAME” and of the same value as the annotation. AbstractDomain membership tags distinguish entity groupings for which AbstractDomains can filter. The tag’s value is a comma-separated list of domain names, where white space is ignored. If an AbstractDomain’s “activeTagValue” Property matches any of a domain tag’s domain names, the AbstractDomain will bind to the Comet entity to which the relevant ActiveTag is attached.

Related glossary terms: *abstract modeling, AbstractDomain, CAD annotation, CAD feature, ActiveTag*

- **ACIS[®] (3D ACIS[®] Modeler)**

A commercial 3-D modeling and computational geometry library, manufactured by Spatial Corporation, which is linked into Comet. Comet supports the import and export of solid models in the ACIS format. See <http://www.spatial.com/components/acis>.

Related glossary terms: *solid model, component software*

- **ActiveTag**

A string-valued annotation to a Comet entity, which marks the entity as having a specific identity, membership or role. Five common uses of ActiveTags are as follows: (1) tagging geometric entities so that they can be easily identified, usually for the purpose of setting them to be members of a PhysicalDomain, *i.e.*, the geometric domain for a load or boundary condition; (2) providing entity groupings for which AbstractDomains can filter; (3) marking ComponentRepresentation, Vertex and Axis entities to have particular roles in LowFidelityInteractions; (4) setting the material type or surface treatment type of a Part, Component, Section or TransferredEntitySet; (5) marking Faces as corresponding to particular optical surfaces within an OpticalSystem. Many ActiveTags originate as CAD model annotations, which are converted to ActiveTags when the CAD model is imported into Comet.

Related glossary terms: *Comet entity, entity identification tag, PhysicalDomain, AbstractDomain membership tag, Interaction modeling specification, material specification tag, optical surface identification tag, CAD annotation*

- **ActiveTagEditor**

A Comet-specific plug-in to a CAD tool. It allows users to interactively attach annotations to CAD model elements, which are converted to ActiveTags when the CAD model is imported into Comet.

Related glossary terms: *CAD, CAD annotation, CAD model, ActiveTag, Pro/ENGINEER, SolidWorks*

- **Adams[®]**

See *MSC.Adams[®]*.

- **AdamsTask**

A type of SimulationTask that manages runs of the MSC.Adams external tool. It is a type of Comet entity found within a Process. “AdamsTask” is an abbreviation of “AdamsTask-005”. See <http://www.mscsoftware.com/products/adams.cfm>.

Related glossary terms: *MSC.Adams, SimulationTask, Process, Comet entity*

- **ANSYS®**

A commercial simulation tool, manufactured by ANSYS, Inc., which Comet supports through its AnsysTask. ANSYS performs finite element analysis, including structural and thermal analysis. See <http://www.ansys.com>.

Related glossary terms: *AnsysTask, external tool*

- **AnsysTask**

A type of SimulationTask that manages runs of the ANSYS external tool. It is a type of Comet entity found within a Process. “AnsysTask” is an abbreviation of “AnsysTask-10.0”. See http://www.hks.com/products/products_standard.html.

Related glossary terms: *ANSYS, SimulationTask, Process, Comet entity*

- **AppliedLightRayOrigin**

A type of EnvironmentEntity that models one member of the test set of field points (*i.e.*, origin and direction) of light rays applied to an optical system during analysis. A set of AppliedLightRayOrigins, together with a set of *AppliedLightWavelengths*, comprise the members of the BoundaryConditionSet of the Environment of a CodeVTask. The set of AppliedLightRayOrigins is represented in a CODE V *sequence file* with the “XAN” and “YAN” lines. (Sequence files have alternative ways of representing this information, which are not supported by Comet.)

Related glossary terms: *EnvironmentEntity, optical system, OpticalSystem, AppliedLightWavelength, BoundaryConditionSet, Environment, CodeVTask, CODE V, sequence file, optics*

- **AppliedLightWavelength**

A type of EnvironmentEntity that models one member of the test set of wavelengths of light rays applied to an optical system during analysis. A set of AppliedLightWavelengths, together with a set of *AppliedLightRayOrigins*, comprise the members of the BoundaryConditionSet of the Environment of a CodeVTask. The set of AppliedLightWavelengths is represented in a CODE V *sequence file* with the “WL” line. Note that it always gives the wavelengths in units of “nm”, rather than in the length unit specified by the “DIM” line. Comet uses LocalSimulationControls to represent the relative weights of the test wavelengths and the identity of the reference wavelength. Sequence files represent the same information using “WTW” and “REF” lines, respectively.

Related glossary terms: *EnvironmentEntity, optical system, OpticalSystem, AppliedLightRayOrigin, BoundaryConditionSet, Environment, CodeVTask, CODE V, sequence file, optics*

- **Assembly**

A type of ComponentRepresentation that hierarchically groups Parts, Interactions and other Assemblies in a way that defines their spatial locations. The Parts and Interactions comprise the hierarchy’s leaf nodes, while the Assemblies comprise its internal nodes. Each Part, Interaction and Assembly in a hierarchy has a CoordinateSystem; the CoordinateSystems are themselves linked into a hierarchy that exactly parallels the *Part/Interaction/Assembly hierarchy* (abbreviated as *Assembly*)

hierarchy), where the CoordinateSystems indicate the spatial location of each Part, Interaction or Assembly relative to its parent. Each Part, Interaction or Assembly can belong to only one Assembly hierarchy; that is, a Part, Interaction or Assembly that composes the root of an Assembly hierarchy is not contained in any other Assembly, while all the other Parts, Interactions and Assemblies in that hierarchy are each directly contained (*i.e.*, parented) by exactly one Assembly.

Related glossary terms: *PhysicalModel, Component hierarchy, ComponentRepresentation, Part, Interaction, Assembly, CoordinateSystem, CAD model*

- **Assembly hierarchy**

A tree that hierarchically groups Parts, Interactions and Assemblies in a way that defines their spatial locations. The Parts and Interactions comprise the tree's leaf nodes, while the Assemblies comprise its internal nodes. Each Part, Interaction and Assembly in an Assembly hierarchy has a CoordinateSystem; the CoordinateSystems are themselves linked into a hierarchy that exactly parallels the Assembly hierarchy, where the CoordinateSystems indicate the spatial location of each Part, Interaction or Assembly relative to its parent and the root ComponentRepresentation's CoordinateSystem is linked to the PhysicalModel's CoordinateSystem—the reference coordinate system for the entire PhysicalModel.

Each Part, Interaction or Assembly can belong to only one Assembly hierarchy; that is, a Part, Interaction or Assembly that composes the root of an Assembly hierarchy is not contained in any other Assembly, while all the other Parts, Interactions and Assemblies in that hierarchy are each directly contained (*i.e.*, parented) by exactly one Assembly.

Each element in an Assembly hierarchy is linked to the corresponding element in a Component hierarchy. The Assembly hierarchy would tend to represent one particular consistent computational model for that Component hierarchy. For example, when a CAD model of an engineering system is imported into Comet, the UEM represents it as three parallel interlinked hierarchies—a Component hierarchy to represent the compositional relationships among the subsystems as abstract entities, an Assembly hierarchy to represent the system geometry, and a CoordinateSystem hierarchy to represent the spatial relationships between subsystems (and, at the leaf level, to provide a local coordinate system with respect to which Part geometry is defined).

Related glossary terms: *PhysicalModel, Component hierarchy, ComponentRepresentation, Part, Interaction, Assembly, CoordinateSystem, CAD model, reference coordinate system, local coordinate system*

- **AuxiliaryGeometry**

A type of Comet entity that contains BasicTopologyElements used to specify some geometric aspect of the UEM other than MeshingDomains and SimulationDomains. Examples of the contained geometry include the Vertex within a CoordinateSystem, and members of the domains of BoundaryConditions and Loads for which appropriate BasicTopologyEntities did not exist within the ComponentGeometry.

Related glossary terms: *Comet entity, PhysicalModel, ComponentRepresentation, CAD model, solid model, ComponentGeometry, ConstructionGeometry, BasicTopologyEntity*

B

- ***base coordinate system***

See *reference coordinate system*.

- ***base quantity types***

A set of dimensionally independent units, which form the bases for a unit system (or space). Comet follows the SI unit system in having seven base quantity types—length, mass, time, thermodynamic temperature, luminous intensity, quantity of matter (molarity), and electrical current. Each has a base unit. For the SI unit system, these are meter, kilogram, second, degrees Kelvin, candela, mole and ampere, respectively.

Related glossary terms: *unit of measure, SI unit system, SI base unit*

- ***BasicTopologyEntity***

A type of Comet entity that represents the basic topological elements of a solid model. The types of BasicTopologyEntity are the 3-dimensional RefVolume, the 2-dimensional Face, the 1-dimensional Edge, and the 0-dimensional Vertex.

BasicTopologyEntities are linked into Boundary Representation (BREP) models of geometry.

Related glossary terms: *Comet entity, ComponentGeometry, AuxiliaryGeometry, ConstructionGeometry, RefVolume, Face, Edge, Vertex, solid model*

- ***BeamSection***

The type of Section used for “beam analysis”, in which the ComponentGeometry is topologically 1-dimensional and the Section geometry is topologically 2-dimensional. Comet currently limits its support for BeamSections to homogeneous material composition and pre-defined parameterized shapes, represented by PrimitiveBeamSections. The currently supported types of PrimitiveBeamSection are RectangleBeamSection and IBeamSection. Other types that will be supported soon are TBeamSection, LBeamSection, ZBeamSection, CircularSolidBeamSection, CircularTubeBeamSection, HollowRectangleBeamSection, HatBeamSection, QuadrilateralBeamSection, and ChannelBeamSection. The material properties of the ComponentRepresentation being simulated are represented as Properties of the SolidSection.

Related glossary terms: *PhysicalModel, Part, ComponentGeometry, Section, ShellSection, SolidSection, PrimitiveBeamSection, Property, material property*

- ***behavioral property***

A characteristic of the behavior of an engineering system.

Related glossary terms: *physical property, material property*

- ***border***

A band running along the perimeter of a Session window. By dragging a point on the border, the user can interactively re-size the Session window.

Related glossary terms: *GUI, Session window*

- ***boundary condition***

For a given simulation method that involves solving a set of physics equations, the values of the solution variables at a user-selected subset of the PhysicalModel over the duration of the analysis. The UEM represents a boundary condition as an EnvironmentEntity that is a member of some Environment's BoundaryConditionSet.

Related glossary terms: *solution variable, PhysicalModel, UEM, EnvironmentEntity, Environment, BoundaryConditionSet*

- ***Boundary Representation solid model***

A solid modeling representation in which a solid object is represented by boundary surfaces and then filled to make solid.

Related glossary terms: *BREP, Constructive Solid Geometry, CSG, solid model*

- ***BoundaryConditionSet***

A type of Comet entity whose parent is an Environment and that contains the set of EnvironmentEntities playing the role of boundary condition within that Environment.

Related glossary terms: *Comet entity, Environment, EnvironmentEntity, boundary condition*

- ***BoundaryDomain***

The type of GeometricClosure that contains all BasicTopologyEntities that lay on the **boundary** of the EntireDomain. These entities have a dimensionality that is one less than that of the highest-dimensionality BasicTopologyEntities found in the EntireDomain. If the *physicalDomainFilterType* value is "Vertex", "Component" or "Interaction", there is no BoundaryDomain. This GeometricClosure type is currently disabled.

Related glossary terms: *abstract modeling, AbstractDomain, GeometricClosure, EntireDomain, EmbeddedDomain, InterfaceDomain*

- ***BoundEntitySet***

The type of Comet entity that, given a particular GeometricClosure within a particular AbstractDomain, contains the set of Comet entities currently bound to that AbstractDomain and belonging to that GeometricClosure. The BoundEntitySet is a child of the GeometricClosure.

Related glossary terms: *Comet entity, abstract modeling, AbstractDomain, GeometricClosure, physicalDomainFilterType, TransferredEntitySet, PhysicalDomain*

- ***BREP***

See *Boundary Representation solid model*.

C

- **CAD**

See *Computer Aided Design*.

- **CAD annotation**

A string-valued annotation (or “tag”) that is attached to a part, assembly or feature within a CAD model. Pro/ENGINEER calls such annotations “parameters”, while SolidWorks calls them “attributes”.

Related glossary terms: *CAD model, entity identification tag, AbstractDomain membership tag, Interaction modeling specification, ActiveTag, Parameter, label*

- **CAD feature**

An auxiliary point or axis that is attached to a CAD model. Pro/ENGINEER calls such features “Datum Points” and “Datum Axes”.

Related glossary terms: *CAD model, Interaction modeling specification, Vertex, CoordinateSystem*

- **CAD model**

A virtual 3-dimensional representation of components for machine design and analysis. Comet supports two CAD model formats: Pro/ENGINEER and SolidWorks. These are both 3-dimensional, parametric feature-based solid model formats.

Related glossary terms: *Computer Aided Design, CAD, solid model, Pro/ENGINEER, SolidWorks, CAD annotation, CAD feature*

- **CAE**

See *Computer Aided Engineering*.

- **Canvas**

A background surface filling the Workspace onto which movable but non-dockable windows, such as Process Schematics, Viewports and Field Visualizers, can be drawn. These windows can be resized and moved, but they are confined to the Canvas. Note that dockable windows occlude the Canvas; that is, they are drawn in front of it. The dark grey area within a Session window is the currently exposed portion of the Canvas.

Related glossary terms: *GUI, Workspace, Scene window, Process Schematic, Viewport, Field visualizer, dockable window, Session window*

- **CML**

See *Computational Modeling Language*.

- **CmtBaseEntityPointerValue**

A type of Value that contains a pointer to a Comet entity.

Related glossary terms: *Value, Comet entity*

- **CmtPosition**

A type of Value that represents a position in space as three components, each of which always contains a numerical quantity and can optionally also contain a unit. If the unit is specified as “X” or not provided, the unit of the currently in-scope CoordinateSystem is used.

Related glossary terms: *Value, CoordinateSystem*

- **CODE V[®]**

A commercial optical design tool, manufactured by Optical Research Associates (ORA[®]), that CoMeT supports through its *CodeVTask*. CODE V performs lens modeling, analysis, optimization and tolerancing. See http://www.opticalres.com/cv/cvprodds_f.html.

Related glossary terms: *CodeVTask, optics, external tool, integrated optomechanical analysis*

- **CodeVTask**

A type of SimulationTask that manages runs of the *CODE V* external tool. It is a type of CoMeT entity found within a Process. “CodeVTask” is an abbreviation of “CodeVTask-9.60”. See http://www.opticalres.com/cv/cvprodds_f.html.

Related glossary terms: *CODE V, SimulationTask, Process, CoMeT entity, optics, integrated optomechanical analysis, optical system, OpticalSystem, optical surface, optical surface identification tag, optical entity, LeafOpticalEntity, mirror, Mirror, lens, Lens, non-physical optical surface, NonPhysicalOpticalSurface, RayTransmissionInteraction, AppliedLightRayOrigin, AppliedLightWavelength, SigFit, SigfitTask, OpticalEntityDisturbance, MirrorDisturbance, LensDisturbance, SimulationComponentDisturbanceSet, Zernike polynomial*

- **ColorValue**

A type of Value that represents a color as three real numbers in the range [0,1], where the numbers indicate the intensities of the color’s red, green and blue components.

Related glossary terms: *Value, drawable object, graphics property*

- **Comet (Computational Modeling Toolkit)**

A CAE work environment for Simulation-Driven Engineering (SDE), developed and distributed by Comet Solutions, Inc. Comet provides a single, consistent environment for using any CAE application for simulation (*e.g.*, structures, thermal, optics, multi-body dynamics, fluid dynamics, spreadsheets, and systems engineering tools), capturing knowledge of product behavior, creating and automating engineering analysis processes, and driving changes to any CAD model based on simulation results. See <http://www.cometsolutions.com>.

Related glossary terms: “*Comet Solutions, Inc.*”, *Computer Aided Engineering, Simulation Driven Engineering*

- **Comet entity**

An object in an instance of the Universal Engineering Model. Comet entities are linked into graphs. They are of type *CmtBaseEntity*, and can be queried for Properties.

Related glossary terms: *Comet, Universal Engineering Model, UEM instance, Property*

- ***Comet Solutions, Inc.***

A startup software company, based in Albuquerque, New Mexico, which develops and distributes engineering software products based upon the Comet framework. See <http://www.cometsolutions.com>.

Related glossary terms: *Comet*

- ***Command Log window***

A dockable GUI window, usually represented with tabs, which displays the stream of CML (Scheme plus Comet-specific extensions) commands sent from the GUI process to the Job process for execution.

Related glossary terms: *GUI, dockable window, CML, Job*

- ***Component***

A type of Comet entity that represents a subsystem of an engineering system. Components can contain other Components, down through as many levels of decomposition as you care to model. A Component is a placeholder for the physical entity at a logical or abstract level; it does not contain data characterizing the physical entity. In contrast, a ComponentRepresentation is a limited representation of the Component for computational purposes, and stores the numerical data that will represent the Component to some category of simulation code.

Related glossary terms: *Comet entity, PhysicalModel, Component hierarchy, ComponentRepresentation*

- ***Component hierarchy***

A tree whose nodes are Components, capturing the compositional relationships among the subsystems of the engineering system being modeled.

Related glossary terms: *PhysicalModel, Component, ComponentRepresentation, CoordinateSystem, CAD model*

- ***component mode synthesis***

The analysis technique used by MSC.Adams in modeling a system of flexible bodies. In this technique, “an efficient modal basis is used to superimpose linear deformations onto a component’s large overall motion. The modal basis is constructed from a blend of normal and static correction modes to accurately capture the component’s dynamic response while assuring the correct interaction with the rest of the mechanical system. The modal basis can be easily modified by disabling individual modes which do not significantly contribute to the component’s motion, thereby reducing computational overhead and speeding up solution times.” This quote is from <http://www.mscsoftware.com/assets/ADAM2004NOVZFLEXZLTDAT.pdf>.

Related glossary terms: *MSC.Adams, AdamsTask, flexible-body dynamics analysis, ModalNeutralConstraintNode, ModalNeutralFile, MNF file, SpiderMeshControl, AuxiliaryVertexMeshControl*

- ***component software***

Software that is not developed at Comet Solutions, but that is linked into the Comet executable and provides some predefined, encapsulated service. Component software currently used by Comet includes the HOOPS 3D Application Framework for graphics, ACIS and Parasolid for 3-D modeling and computational geometry, ACIS again for history management, CORBA for inter-process communication, and VTK for Field visualization.

Related glossary terms: *HOOPS 3D Application Framework, ACIS, Parasolid, CORBA, VTK*

- ***ComponentGeometry***

A type of Comet entity that represents the geometric domain of the Part to which this entity belongs. It always directly contains one Body if the geometry is 3-dimensional, one Face if it is 2-dimensional, one Edge if it is 1-dimensional, or one Vertex if it is 0-dimensional. If a ComponentRepresentation is the MeshDomain of a MeshingTask, its ComponentGeometry (or, if it is an Assembly or UserDefinedComponentRepresentation, the union of the ComponentGeometries of its Parts) defines the geometry that will be meshed. If the ComponentRepresentation is the SimulationComponent of the SimulationDomain of a SimulationTask, its ComponentGeometry defines the geometry that will be simulated.

Related glossary terms: *Comet entity, PhysicalModel, ComponentRepresentation, CAD model, solid model, AuxiliaryGeometry, ConstructionGeometry, BasicTopologyEntity*

- ***ComponentMesh***

A type of Comet entity that holds the meshes that have been constructed to fit the ComponentGeometry of the Part to which this entity belongs. The types of meshes currently supported in Comet are PrimitiveUnstructuredMeshes and CompositeUnstructuredMeshes.

Related glossary terms: *Comet entity, PhysicalModel, Part, ComponentGeometry, PrimitiveUnstructuredMesh, CompositeUnstructuredMesh, MeshingTask*

- ***ComponentMeshControlSet***

A type of Comet entity that belongs to a MeshProcedure and contains the meshing controls that are specific to particular Components.

Related glossary terms: *Comet entity, MeshingTask, MeshProcedure, Component*

- ***ComponentRepresentation***

A type of Comet entity that contains a limited numerical representation, for computational purposes, of the Component to which it belongs. The ComponentRepresentation stores the numerical data that will represent the Component to some category of simulation code. The leaf types of ComponentRepresentations are Parts and Interactions. The types of ComponentRepresentations that group other ComponentRepresentations in various ways are Assemblies, UserDefinedComponentRepresentations and (still to be implemented) AggregateParts.

Related glossary terms: *Comet entity, PhysicalModel, Component, Part, Interaction, Assembly, UserDefinedComponentRepresentation, AggregatePart, Assembly*

hierarchy, CoordinateSystem, LowFidelityComponentRepresentation, ComponentGeometry, AuxiliaryGeometry, ConstructionGeometry, ComponentMesh

- ***CompositeUnstructuredMesh***

A type of Comet entity that represents the composition of any number of PrimitiveUnstructuredMeshes, where the PrimitiveUnstructuredMeshes are not altered in any way by this composition. The purpose of this composition is to have a single Mesh to be assigned as the Mesh to be used during the simulation of a particular ComponentRepresentation, that is, the one Mesh belonging to a SimulationTask's SimulationMesh that corresponds to the given ComponentRepresentation.

Related glossary terms: *Comet entity, ComponentMesh, ComponentGeometry, PrimitiveUnstructuredMesh, non-manifold geometry, manifold geometry, mesh element topology type, MeshingTask*

- ***Computational Modeling Language (CML)***

The programming language used to drive a Comet Job. CML is a Comet-specific extension to Scheme.

Related glossary terms: *Job, UserDefinedTask, ConstraintFunction, ObjectiveFunction*

- ***Computer Aided Design (CAD)***

The use of a wide range of computer-based tools that assist engineers, architects and other design professionals in their design activities. It is the main geometry authoring tool within the Product Lifecycle Management process and involves both software and sometimes special-purpose hardware. Current packages range from 2D vector-based drafting systems to 3D parametric surface and solid design modelers. "CAD" is sometimes translated as "computer- aided drafting" or "computer- assisted d...".

Related glossary terms: *CAD, CAD model, solid model, CAD annotation, CAD feature, Pro/ENGINEER, SolidWorks, Product Lifecycle Management*

- ***Computer Aided Engineering (CAE)***

The application of computer software in engineering to analyze the robustness and performance of components and assemblies. It encompasses simulation, validation and optimization of products and manufacturing tools.

Related glossary terms: *Computer Aided Design, Simulation Driven Engineering, Virtual Product Engineering, Comet*

- ***ConstructionGeometry***

A type of Comet entity that contains temporary geometry used during the construction of other aspects of the UEM. It uses its own types of elements, such as Axis and Point, rather than BasicTopologyEntities.

Related glossary terms: *Comet entity, PhysicalModel, ComponentRepresentation, CAD model, solid model, ComponentGeometry, AuxiliaryGeometry, BasicTopologyEntity*

- ***Constructive Solid Geometry (CSG)***

A procedural solid modeling technique that creates a complex surface or object by using Boolean operators to combine simple primitive solid objects, such as boxes, cylinders, prisms, pyramids, spheres and cones.

Related glossary terms: *CSG, Boundary Representation solid model, BREP, solid model*

- ***context menu***

The pull-down menu generated by right-clicking a GUI object. It allows users to select an action to be performed on the Comet entity represented by the given GUI object.

Related glossary terms: *GUI, GUI object, Comet entity*

- ***CoordinateSystem***

A type of Comet entity that represents a *coordinate system*, which is a method of representing points in an n -dimensional space as tuples of n numbers, or *coordinates*. All CoordinateSystems in a UEM instance are linked into a tree, called the *CoordinateSystem hierarchy*. The CoordinateSystem attached to each ComponentRepresentation captures the geometric transform from the *local coordinate system* in which that ComponentRepresentation's geometry is defined to its parent's coordinate system. Each CoordinateSystem contains a Vertex, which holds the translational component of the transform; that is, the Vertex defines the location of its CoordinateSystem relative to the *reference coordinate system*. The CoordinateSystem internally stores additional rotational and scaling data that, in conjunction with the Vertex, defines a complete affine transformation from the local coordinate system to its parent's coordinate system.

Related glossary terms: *Comet entity, PhysicalModel, ComponentRepresentation, reference coordinate system, base coordinate system, global coordinate system, local coordinate system, CoordinateSystem hierarchy, Vertex, AuxiliaryGeometry*

- ***CoordinateSystem hierarchy***

The tree containing all CoordinateSystems in a UEM instance; its root is the CoordinateSystem attached directly to the PhysicalModel, which is called the *reference coordinate system*. Thus, any CoordinateSystem defines a local coordinate system, specified through a chain of geometric transforms, relative to the PhysicalModel's reference coordinate system.

Related glossary terms: *CoordinateSystem, UEM instance, reference coordinate system, local coordinate system, PhysicalModel*

- ***CSG***

See *Constructive Solid Geometry*.

- ***current Stage***

During a Comet session, the Stage whose state is currently loaded into memory. It is the sole active Stage, which can be queried and modified, and from which Processes can be run.

Related glossary terms: *Project, Stage*

D

- **DAKOTA[®]**

An interface between analysis codes and iterative systems analysis methods, such as parameter studies and optimization. The DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit is developed at Sandia National Laboratories and is freely available under a GPL license. An earlier version of Comet supported parameter study and optimization capabilities through its *DakotaTask*, and DAKOTA is currently being considered as the means of providing optimization capability in the future. See <http://endo.sandia.gov/DAKOTA>.

Related glossary terms: *DakotaTask*, *design space exploration*, *external tool*

- ***DakotaTask***

A type of *DesignSpaceExplorationTask* that manages runs of the DAKOTA external tool. It is a type of Comet entity found within a *Process*. See <http://endo.sandia.gov/DAKOTA>. (This Task type is currently unavailable.)

Related glossary terms: *DAKOTA*, *design space exploration*, *DesignVariable*, *ObjectiveFunction*, *ConstraintFunction*, *Sample Point Evaluation Process*, *DesignSpaceExplorationTask*, *Process*, *Comet entity*

- ***data model***

(a) A model that describes in an abstract way how data is represented within some domain.

(b) A set of data objects, their relationships, and the semantics of what the data and its relationships mean.

In the case of the Comet data model, called the Universal Engineering Model[™] (UEM[™]), the data being organized is all of the “current” data associated with a Virtual Product Engineering project at some point in its history. This would include the active set of CAD models, systems models, CAE models, environment specifications, process specifications and results data.

Related glossary terms: *Comet*, *Universal Engineering Model*, *UEM*, *Virtual Product Engineering*

- ***dataflow***

A software architecture based on the idea that changing the value of a variable should automatically force recalculation of the values of other variables. Comet’s *TopMan* tool for specification and execution of engineering analysis processes represents the processes as dataflow networks of computational modules. A *Process* does not explicitly specify the order in which its *Tasks* will be updated when the *Process* as a whole is run. Instead, the order emerges dynamically due to dataflow constraints.

Related glossary terms: *TopMan*, *Process*, *run*, *Task*, *update*

- ***DataPipe***

A type of Comet entity that connects an appropriate pair of Ports and provides flow of data, in the form of Values, from the upstream Port to the downstream Port. DataPipes can connect ProcessInputPorts with Task-level InputPorts, Task-level OutputPorts with Task-level InputPorts, and Task-level OutputPorts with ProcessOutputPorts. Data flow occurs through **copying**; that is, when a datum is placed on an upstream Port, a copy of that Value is immediately placed within each of the downstream Ports connected to it by a DataPipe. Note that Port data in the form of CmtBaseEntityPointerValues, which are the carriers for such bulky data as Meshes and Fields, only copy references to the CmtBaseEntity and not the CmtBaseEntity itself.

Related glossary terms: *Comet entity, Process, Task, Value, update, InputPort, OutputPort, ProcessInputPort, ProcessOutputPort, dataflow, ToPMan, CmtBaseEntityPointerValue*

- ***declared Property***

A Property of the given source entity whose content is declared, because it cannot be derived from the source entity's other data, such as its geometry and other Properties.

Related glossary terms: *Property, derived Property*

- ***default Value***

For a given SingleValuedDynamicEntity type, the Value to be assigned to new instances of that type if a Value is not explicitly supplied to the constructor. Some SingleValuedDynamicEntity types are defined to not have a default Value (instead, they have a sample Value); in this case, an initial Value must be explicitly supplied in order to construct a new instance of that type.

Related glossary terms: *SingleValuedDynamicEntity, Value, sample Value*

- ***derived Property***

A Property of the given source entity whose content is derived via computation from the source entity's other data, such as its geometry and other Properties. Examples include Area, Volume, Mass, CenterOfMass and MomentOfInertia for entities with geometry.

Related glossary terms: *Property, declared Property*

- ***DesignSpaceExplorationTask***

A type of Task that manages runs of an external tool that performs design space exploration. The DakotaTask is an example of a DesignSpaceExplorationTask.

Related glossary terms: *Task, design space exploration, external tool, DAKOTA, DakotaTask*

- ***dimensionality***

The integral power to which the subject is raised. For example, in the definition for the "Newton" unit of $kg * m / s^2$, the dimensionality of kg is 1, that of m is also 1, that of s is -2, and that of the other SI base units is 0.

Related glossary terms: *unit of measure, UnitDictionary, base quantity types, SI base unit*

- ***DiscreteFieldRange***

<TBD>

Related glossary terms: ***Field, FieldRange***

- ***dockable window***

An area in the GUI enclosed in a rectangular frame that can be moved, resized, closed and docked in one of four docking locations. The docking locations are alongside the top, bottom, left and right edges, respectively, of the Workspace. When not docked, a dockable window has its own title bar and close button. When docked, it has a tab in the bar of tabs for that docking location. Examples of dockable windows include project views, Comet entity editors, the Terminal window, the CommandLog window, and Query windows.

Related glossary terms: ***GUI, title bar, project view, editor, Terminal window, Command Log window, Query window***

- ***drawable object***

A Comet entity (represented in the GUI by a corresponding GUI object) that can be graphically rendered. A *Scene* is a set of drawable objects, a *Viewport* is a particular view of the drawable objects contained in the given Scene, and a *Scene window* is the window within which the Viewport is rendered. Drawable types of Comet entities include AbstractDomain, Body, BoundaryConditionSet, Component, ComponentRepresentation (including sub-types Part, Assembly, Interaction and UserDefinedComponentRepresentation), CoordinateSystem, Edge, Environment, EnvironmentEntity, Face, InitialConditionSet, LoadSet, Mesh, PhysicalDomain, PhysicalModel, PrescribedConditionSet, SimulationMesh, and Vertex.

Related glossary terms: ***GUI, Comet entity, GUI object, Scene, Viewport, view, Scene window***

- ***DynamicEntity***

A Value-holding Comet entity that is of a dynamically defined, named type. A DynamicEntity type sets the algebraic type (*i.e.*, the number and organization of the constituent Values), a Boolean indicating whether the type has a default Value (if not, it will instead have a sample Value), the default or sample Value, and the Value range for all instances of the given type.

Related glossary terms: ***Comet entity, Value, default Value, sample Value, Range, SingleValuedDynamicEntity***

[Go to start of Glossary](#)

E

- ***Edge***

The type of BasicTopologyEntity that represents 1-dimensional geometric elements.

Related glossary terms: ***BasicTopologyEntity***

- ***editor***

A dockable GUI window that permits users to view and modify information about a Comet entity (represented by a GUI object).

Related glossary terms: *GUI, dockable window, Comet entity, GUI object*

- ***EmbeddedDomain***

The type of GeometricClosure that contains all BasicTopologyEntities that lay within the **interior** of the EntireDomain. These entities have a dimensionality that is one less than that of the highest-dimensionality BasicTopologyEntities found in the EntireDomain, and compose all the entities of that dimensionality in the EntireDomain that are not members of the BoundaryDomain. For example, if the *physicalDomainFilterType* value was “RefVolume”, the EmbeddedDomain would contain the fully interior Faces of the RefVolumes in the EntireDomain. If the *physicalDomainFilterType* value is “Vertex”, “Component” or “Interaction”, there is no EmbeddedDomain. This GeometricClosure type is currently unimplemented.

Related glossary terms: *abstract modeling, AbstractDomain, GeometricClosure, EntireDomain, BoundaryDomain, InterfaceDomain*

- ***EntireDomain***

The type of GeometricClosure that contains **all** entities of the type given by *physicalDomainFilterType* to which the AbstractDomain binds.

Related glossary terms: *abstract modeling, AbstractDomain, GeometricClosure, physicalDomainFilterType, BoundaryDomain, EmbeddedDomain, InterfaceDomain*

- ***entity identification tag***

A string-valued CAD annotation of type “COMET_OBJECT_LABEL”, which is attached to a geometric entity, such as a face or vertex. When it is imported into Comet, it is represented doubly within the Comet model as the corresponding Comet entity’s label and as an ActiveTag named “COMET_OBJECT_LABEL” attached to the same Comet entity. The label and the ActiveTag both have the same value as the CAD annotation. The tags serve to easily identify the entities in later situations, such as setting the PhysicalDomain for a BoundaryCondition or Load.

Related glossary terms: *CAD annotation, label, ActiveTag*

- ***Environment***

A type of Comet entity that represents an instance of external conditions to which the engineering system being designed could be subjected. A simulation run analyzes how the engineering system would behave in some particular physical situation. That physical situation is specified by the Environment contained within the SimulationEnvironment of each of a SimulationTask’s Procedures, in conjunction with that SimulationTask’s set of InitialConditions. An Environment contains a BoundaryConditionSet, a LoadSet and a PrescribedConditionSet.

Related glossary terms: *Comet entity, EnvironmentSet, BoundaryCondition, Load, InitialCondition, PrescribedCondition, SimulationTask, Procedure, SimulationEnvironment, solution variable, PhysicalDomain*

- ***EnvironmentEntity***

A type of Comet entity that represents an element within an Environment, where that entity plays the role of a boundary condition, a load, a prescribed condition or an initial condition. The role that the EnvironmentEntity is playing in some given context is indicated by its parent being, variously, a BoundaryConditionSet, a LoadSet, a PrescribedConditionSet, or an InitialConditionSet. Note that the same EnvironmentEntity can play different roles within different Environments; for example, an EnvironmentEntity specifying a Temperature distribution across a particular surface could play the role of boundary condition in a thermal analysis and prescribed condition in a structural analysis. An Environment contains a BoundaryConditionSet, a LoadSet and a PrescribedConditionSet, while a SimulationTask contains an InitialConditionSet. An EnvironmentEntity's PhysicalDomain represents the subset of the PhysicalModel over which the boundary condition, load or prescribed condition is defined.

Related glossary terms: *Comet entity, Environment, solution variable, boundary condition, load, prescribed condition, initial condition, BoundaryConditionSet, LoadSet, InitialConditionSet, PrescribedConditionSet, InitialConditionSet, SimulationTask, Procedure, SimulationEnvironment, PhysicalDomain*

- ***EnvironmentSet***

A top-level Comet entity, which contains all of the Environments defined within a UEM instance. Every Project has exactly one EnvironmentSet.

Related glossary terms: *Comet entity, UEM instance, Environment, top-level entity*

- ***Excel[®]***

A commercial spreadsheet tool, manufactured by Microsoft Corporation, that Comet supports through its ExcelTask. See <http://office.microsoft.com/excel>.

Related glossary terms: *ExcelTask, spreadsheet, external tool*

- ***ExcelTask***

A type of SpreadsheetTask that manages runs of the Excel[®] external tool. It is a type of Comet entity found within a Process. "ExcelTask" is an abbreviation of "ExcelTask-2003". See <http://office.microsoft.com/excel>.

Related glossary terms: *Excel, SpreadsheetTask, Process, Comet entity*

- ***execution queue***

The data structure within ToPMan that holds an ordered list of the Tasks that are currently eligible to be updated. A Task is added to the end of the queue when all of its InputPorts, if any, have received their inputs. (Tasks that do not have any InputPorts are queued for execution once at the start of a Process run.) The Task at the front of the queue is updated as soon as a processor is available, whereupon the Task is removed from the queue. Updating a Task empties its InputPorts. Tasks that have InputPorts can be repeatedly scheduled for update, provided that its InputPorts receive fresh batches of inputs.

Related glossary terms: *Process, Task, update, run, dataflow, ToPMan*

- ***Expression***

A type of Comet entity that contains a combination of values, operators and functions, defined by a string, which is interpreted and evaluated according to Comet's particular rules of precedence and association. Comet supports two broad categories of Expressions. The first category consists of Expressions that can be fully evaluated in Comet; they all evaluate to Values. The second category consists of Expressions that are meant to be exported to external tools for evaluation there and can never be fully evaluated by Comet alone.

Related glossary terms: *Comet entity, Value, Comet entity*

- ***external tool***

A software tool that Comet uses but that is not part of Comet. The software for the external tool resides in a separate executable from Comet, is never modified, and is run as a separate process from the Comet GUI and Job processes. Examples of currently supported external tools are ABAQUS, ANSYS, CODE V, DAKOTA, Excel, FLUENT, MSC.Adams, MSC.Nastran, Pro/ENGINEER, SolidWorks, the visualizer ParaView, and *PuTTY*. Tasks that launch external tool processes have a *LauncherConfiguration*, which sets the path of the external tool executable to be launched and, through the *LauncherType* Property, whether the external tool is to be run remotely or locally.

Related glossary terms: *LauncherConfiguration, LauncherType, MeshingTask, SimulationTask, SpreadsheetTask, DesignSpaceExplorationTask, ABAQUS, ANSYS, CODE V, DAKOTA, Excel, FLUENT, ParaView, MSC.Adams, MSC.Nastran, Pro/ENGINEER, SolidWorks, ThermalDesktop*

- ***evaluatable***

A characteristic of an Expression, indicating whether evaluation of the Expression currently succeeds. The typical reason for an Expression with valid syntax not being evaluatable is that one of its references cannot be resolved.

Related glossary terms: *Expression, evaluate*

- ***evaluate***

To calculate the value of an Expression.

Related glossary terms: *Expression, Value*

[Go to start of Glossary](#)

F

- ***Face***

The type of *BasicTopologyEntity* that represents 2-dimensional geometric elements.

Related glossary terms: *BasicTopologyEntity*

- ***Field***

A type of Comet entity that represents a mapping, either explicitly or implicitly, from each point in a domain to a Value. A Field contains a *FieldDomain* and a *FieldRange*.

Related glossary terms: *Comet entity, FieldDomain, FieldRange, dimension, Plot, PlotType, 2DPlot, 3DPlot, 4DPlot, TimeSeriesField, XMLBasedPlot, UserDefinedFieldSet, InterpolationType, ExtrapolationType, MinimumMagnitude, MaximumMagnitude, MeanMagnitude, FieldVisualizationTask*

- ***FieldDomain***

A type of Comet entity that represents the collection of points for which a Field can generate a mapped Value. The currently supported types of FieldDomain are MeshBasedDomain, GridNodeBasedDomain and KeyBasedDomain.

Related glossary terms: *Comet entity, Field, MeshBasedDomain, GridNodeBasedDomain, KeyBasedDomain, FieldRange*

- ***FieldRange***

A type of Comet entity that represents mapping, either explicitly or implicitly, from each point in the affiliated FieldDomain to a Value. The currently supported types of FieldRange are NumericalFieldRange, DiscreteFieldRange and PropertyBasedFieldRange.

Related glossary terms: *Comet entity, Field, NumericalFieldRange, DiscreteFieldRange, PropertyBasedFieldRange, FieldDomain*

- ***FirstOrderTensor***

A type of Value that represents a vector of any length, each of whose elements contains a numerical quantity and a unit, where the numerical component can be either a real number or an integer.

Related glossary terms: *Value*

- ***flexible-body dynamics analysis***

Comet supports a three-step process for the modeling of flexible-body dynamics: (1) rigid-body dynamics analysis by MSC.Adams defines the large overall motion of each body, output as a set of constraint nodes for each one; (2) FEA structural analysis by ANSYS of each body generates Modal Neutral Files that characterize each body's flexibility; (3) flexible-body dynamics analysis by MSC.Adams superimposes linear deformations onto each body's large overall motion. The analysis technique employed by MSC.Adams in the third step is *component mode synthesis*.

Related glossary terms: *MSC.Adams, AdamsTask, ANSYS, AnsysTask, MSC.Nastran, MSCNastranTask, component mode synthesis, ModalNeutralConstraintNode, ModalNeutralFile, MNF file, SpiderMeshControl, AuxiliaryVertexMeshControl*

- ***Fluent[®]***

A commercial simulation tool, manufactured by ANSYS, Inc. (following its acquisition of Fluent, Inc. in May 2006), that Comet supports through its FluentTask. Fluent performs computational fluid dynamics (CFD) analysis. See <http://www.fluent.us> and <http://www.ansys.com>.

Related glossary terms: *FluentTask, CFD, computational fluid dynamics, external tool*

- ***FluentTask***

A type of SimulationTask that manages runs of the Fluent[®] external tool. It is a type of Comet entity found within a Process. “FluentTask” is an abbreviation of “FluentTask-6.2”. See <http://www.fluent.us>. (This Task type is currently unavailable.)

Related glossary terms: *Fluent, SimulationTask, Process, Comet entity*

[Go to start of Glossary](#)

G

- ***GeometricClosure***

The type of Comet entity that contains a topologically defined subset of the BasicTopologyEntities contained within the Comet entities to which the AbstractDomain binds. The only type of GeometricClosure that is currently enabled is the *EntireDomain*. Other types are the *BoundaryDomain*, which is implemented but disabled, and the *EmbeddedDomain* and *InterfaceDomain*, which are not implemented.

Related glossary terms: *Comet entity, abstract modeling, BasicTopologyEntity, AbstractDomain, EntireDomain, BoundaryDomain, EmbeddedDomain, InterfaceDomain*

- ***global coordinate system***

See *reference coordinate system*.

- ***GlobalMeshControlSet***

A type of Comet entity that belongs to a MeshProcedure and contains the globally applicable controls for a meshing computation. These include “meshElementTopologyType”, “maximumMeshElementSize”, “meshElementSizePropagationFactor”, “maximumMeshCurvatureTolerance”, “maximumCurvatureBasedMeshElementSize”, “nodalCoordinateOutputPrecision”, and “maximumSmallFeatureSize”.

Related glossary terms: *Comet entity, MeshingTask, MeshProcedure*

- ***GlobalSimulationControlSet***

A type of Comet entity that belongs to a SimulationProcedure and contains the globally applicable controls for the phase of the simulation computation controlled by the given Procedure.

Related glossary terms: *Comet entity, SimulationTask, SimulationProcedure*

- ***Graphical User Interface (GUI)***

- (a) A method of interacting with a computer through a metaphor of direct manipulation of graphical images and widgets in addition to text.
- (b) The Comet process that provides the front-end interface with which Comet users interact. It composes the top tier of Comet’s 3-tiered architecture, where Job processes and external tool processes compose the other two. The GUI provides a layout of graphic and text controls with which users can interact. While Comet’s

GUI uses screen pointers, icons, menus, windows, and other features typically associated with Microsoft Windows® applications, it was designed with cross-platform tools to maximize its versatility.

Related glossary terms: *Comet, Job, external tool, border, Canvas, Command Log window, context menu, dockable window, drawable object, editor, GUI object, hotkey, menu bar, menu pad, pane, Process Schematic, project view, Query window, Scene, Scene window, Session, Session window, splitter, status bar, system menu, Terminal window, title bar, toolbar, view, view manipulation toolbar, Viewport, Windows control buttons, Workspace*

- ***graphics Property***

A Property controlling some aspect of how the drawable Comet entities for which the given graphics Property is in scope are visualized. Examples include color, transparency and line thickness.

Related glossary terms: *Property, Comet entity, drawable*

- ***GridNodeBasedDomain***

The type of FieldDomain whose members, called *nodes*, form a grid. For each dimension, the FieldDomain contains a set of coordinate values along that dimension's axis, where the node set is defined implicitly as being every possible combination of a coordinate value for each axis.

Related glossary terms: *Field, FieldDomain, Plot*

- ***GUI***

See *Graphical User Interface*.

- ***GUI object***

A GUI representation of a particular Comet entity. Right-clicking on it opens a context menu that allows users to select an action to be performed on the given Comet entity.

Related glossary terms: *GUI, context menu, Comet entity*

[Go to start of Glossary](#)

H

- ***HOOPS 3D Application Framework (HOOPS/3dAF)***

A commercial 2-D and 3-D graphics framework, manufactured by Tech Soft America, which is linked into Comet. See <http://www.spatial.com/components/hoops>.

Related glossary terms: *graphics, component software*

- ***hotkey***

A key combination linked to commands or menu items. Other names for *hotkey* are *keyboard equivalent* (an alternative to using the mouse buttons) and *keyboard shortcut*. Hotkeys in any software usually include a booster key (CTRL, ALT, or

SHIFT) and a commonly used key on the keyboard or a function key (e.g., F1, F2). Comet provides the typical menu hotkeys (e.g., ALT+F opens the File menu) and some atypical hotkeys (CTRL+A creates a new AbstractDomain object in the AbstractDomain project view.).

Related glossary terms: *GUI, menu bar, menu pad*

[Go to start of Glossary](#)

I

- ***inherited Property***

A Property that is associated with the given source entity by inheritance from the source entity's nearest ancestor to have a Property of the given name.

Related glossary terms: *Property, transferred Property, owned Property*

- ***initial condition***

For a given simulation method that involves solving a set of physics equations, the values of the solution variables over the entire PhysicalModel at just the starting time of the analysis (i.e., time = 0). The UEM represents an initial condition as an EnvironmentEntity that is a member of some SimulationTask's InitialConditionSet.

Related glossary terms: *solution variable, PhysicalModel, UEM, EnvironmentEntity, SimulationTask, InitialConditionSet*

- ***InitialConditionSet***

A type of Comet entity whose parent is a SimulationTask and that contains the set of EnvironmentEntities playing the role of initial condition within that Task. These initial conditions are applied at the start of the entire simulation run. The solutions to the first analysis phase (controlled by the first SimulationProcedure) then supply the initial conditions for the next analysis phase (controlled by the second SimulationProcedure, if one exists), and so forth for the remainder of the chain of analysis phases specified by the SimulationTask's Procedures.

Related glossary terms: *Comet entity, SimulationTask, EnvironmentEntity, initial condition, SimulationProcedure*

- ***InputPort***

A type of Comet entity that belongs to a Task and receives data in the form of a Value across the DataPipe to which the InputPort is connected. At the start of a Task update, the input datum within each of the Task's InputPorts is inserted into the Comet model at a location specified relative to the Task itself, following which the InputPorts are cleared

Related glossary terms: *Comet entity, Value, Task, DataPipe, InputPortSet*

- ***InputPortSet***

A type of Comet entity that is attached directly to a Task and contains all the InputPorts belonging to that Task.

Related glossary terms: *Comet entity, Task, InputPort*

- ***interferogram file (.int file)***

A format of input file supported by the *CODE V* optics analysis code, which uses it to attach data to an optical surface, the entrance pupil, or the exit pupil. *SigFit* exports a Zernike polynomial fitting of an optical surface deformation as an interferogram file. The rigid-body motion component of the fitting is exported as a tiny *sequence file*.

Related glossary terms: *CODE V, optical surface, SigFit, sequence file, optics*

- ***integrated optomechanical analysis***

Multi-physics analysis of the optical performance of telescopes, lens barrels and other optical systems, taking into account structural and thermal effects upon the system. It is usually achieved through loose coupling of structural/thermal analysis and optics analysis. Comet supports the conversion of optical surface deformations from the discrete, mesh-based form generated by FEA structural/thermal analysis codes (as nodal displacement fields) to the smooth analytical form required by optical analysis codes through its adaptor to SigFit.

Related glossary terms: *CODE V, CodeVTask, SigFit, SigfitTask, MSC.Nastran, MSCNastranTask, ANSYS, AnsysTask, ABAQUS, AbaqusTask, optical system, optics, OpticalEntityDisturbance*

- ***Interaction***

A type of ComponentRepresentation that defines the behavioral relationship between two portions of the PhysicalModel. That is, it captures the physics in the space between two Components without defining any relevant geometry. *Interaction* is a peer to *Part* in that both appear as leaf nodes in Assembly hierarchies; an Interaction may reference other ComponentRepresentations whose behavioral relationship it is defining, but it does not contain them.

Related glossary terms: *Comet entity, PhysicalModel, ComponentRepresentation, Part, Assembly hierarchy, LowFidelityInteraction, PhysicalDomain, Interaction modeling specification, CAD model, CAD annotation, ActiveTag, SimulationTask, MSC.Adams, AdamsTask, CODE V, CodeVTask*

- ***Interaction modeling specification***

A set of Comet-specific CAD annotations and features that, when the CAD model is imported into Comet, specify a LowFidelityInteraction.

Related glossary terms: *Interaction, LowFidelityInteraction, CAD model, CAD annotation, CAD feature, ActiveTag, MSC.Adams, AdamsTask, CODE V, CodeVTask*

- ***InterfaceDomain***

The type of GeometricClosure that contains all BasicTopologyEntities within the EntireDomain that **touch** objects not belonging to the EntireDomain. These entities have a dimensionality that is one less than that of the highest-dimensionality BasicTopologyEntities found in the EntireDomain. If the *physicalDomainFilterType* value is “Vertex”, “Component” or “Interaction”, there is no InterfaceDomain. This GeometricClosure type is currently unimplemented.

Related glossary terms: *abstract modeling*, *AbstractDomain*, *GeometricClosure*, *EntireDomain*, *BoundaryDomain*, *EmbeddedDomain*

[Go to start of Glossary](#)

J

- ***Job***

A Comet kernel process, which is driven by a stream of CML commands, maintains the UEM instance for the currently loaded Project, and does Comet’s real work. Job processes compose the middle tier of Comet’s 3-tiered architecture, where the GUI process and external tool processes compose the other two.

When Comet is launched, two executables are simultaneously launched—comet-job.exe and comet-gui.exe—which become the GUI process and the sole initial Job process. The GUI process opens a Session window, which provides the means for the user to interact with that initial Job; the Session window’s *title bar* displays “Initializing Job-1” before any data in the Project is saved.

Related glossary terms: *Comet*, *UEM instance*, *CML*, *GUI*, *external tool*, *title bar*, *Project*, *Session*

[Go to start of Glossary](#)

K

- ***KeyBasedDomain***

The type of FieldDomain whose members are a set of string-valued keys.

Related glossary terms: *Field*, *FieldDomain*

[Go to start of Glossary](#)

L

- ***label***

For a given Comet entity, an owned Property named “label”, represented as a SingleValuedDynamicEntity of type GenericString whose Value is a string identifier for the entity. All Comet entities are guaranteed to own a “label” Property, even if the string value is empty.

Related glossary terms: *Comet entity*, *owned Property*

- ***LauncherConfiguration***

A type of Comet entity that belongs to a Task that needs to launch an external tool and specifies how to do so. The LauncherConfiguration contains an “ExecutablePath”

Property that sets the path of the executable to be launched and a “LauncherType” Property that sets whether the external tool will be run locally or remotely. If the LauncherType is “SSH”, indicating that the external tool will be run remotely, then the LauncherConfiguration also specifies the “remoteWorkDirectory”, the “host”, and the “login” name. If the LauncherType is “Local”, indicating that the external tool will be run locally, then the LauncherConfiguration also specifies whether the external tool process will be run in the background, meaning that Comet will not monitor the process’ status once it is launched; an example of an external tool that is usually run in the background is the visualizer, ParaView.

Related glossary terms: *Comet entity, Task, external tool, LauncherType, SimulationTask, SpreadsheetTask, DesignSpaceExplorationTask*

- **LauncherType**

A Property of a LauncherConfiguration. If the LauncherType is set to “Local”, the external tool will be run locally; if it is set to “SSH”, the external tool will be run remotely.

Related glossary terms: *external tool, LauncherConfiguration, PuTTY, SimulationTask, SpreadsheetTask, DesignSpaceExplorationTask*

- **leaf Stage**

A leaf in the tree of Stages comprising a Project. A leaf Stage has no child Stages. Non-leaf (or *trunk*) Stages are locked, whereas leaf Stages can be modified unless they are already in use by another user or Session.

Related glossary terms: *Comet, Stage, Project, locked Stage, trunk Stage*

- **LeafOpticalEntity**

A type of ComponentRepresentation that models the optics-specific aspects of a component of an optical system, or *optical entity*. Comet divides optical entities into *mirrors, lenses* and *non-physical optical surfaces*, which it represents by the *Mirror, Lens* and *NonPhysicalOpticalSurface* subtypes of LeafOpticalEntity, respectively. A *RayTransmissionInteraction* models the relationship between two adjacent optical entities.

Related glossary terms: *CODE V, optical entity, optical system, mirror, Mirror, lens, Lens, non-physical optical surface, NonPhysicalOpticalSurface, RayTransmissionInteraction*

- **lens (note the lower-case letter “l”)**

(1) A ground or molded piece of glass, plastic, or other transparent material with opposite surfaces, either or both of which are curved, by means of which light rays are refracted so that they converge or diverge to form an image. A lens is a type of *optical entity*.

(2) The term by which CODE V refers to an *optical system*.

Related glossary terms: *optical entity, CODE V, optical system, optics*

- **Lens (note the upper-case letter “L”)**

A type of LeafOpticalEntity that models a *lens*. It consists of an adjacent pair of refractive, transmissive *optical surfaces* for which the material following the first

surface is **not** “Air” or “Vacuum” and the material following the second surface is “Air” or “Vacuum”.

Related glossary terms: *lens*, *LeafOpticalEntity*, *optical surface*

- **load**

For a given simulation method that involves solving a set of physics equations, the forcing functions of the equations being solved at a user-selected subset of the PhysicalModel over the duration of the analysis. The UEM represents a load as an EnvironmentEntity that is a member of some Environment’s LoadSet.

Related glossary terms: *solution variable*, *PhysicalModel*, *UEM*, *EnvironmentEntity*, *Environment*, *LoadSet*

- **LoadSet**

A type of Comet entity whose parent is an Environment and that contains the set of EnvironmentEntities playing the role of load within that Environment.

Related glossary terms: *Comet entity*, *Environment*, *EnvironmentEntity*, *load*

- **local coordinate system**

For a given computational representation of geometry, the coordinate system relative to which the geometry coordinates are specified. The UEM represents the local coordinate system of a ComponentRepresentation by a CoordinateSystem attached to it.

Related glossary terms: *PhysicalModel*, *ComponentRepresentation*, *CoordinateSystem*, *reference coordinate system*

- **LocalMeshControlSet**

A type of Comet entity that belongs to a MeshProcedure and contains the meshing controls that are specific to local regions of the Component geometry. The BasicTopologyEntities defining the domain for a particular local mesh control are contained within the control’s PhysicalDomain.

Related glossary terms: *Comet entity*, *MeshingTask*, *MeshProcedure*, *PhysicalDomain*

- **LocalSimulationControlSet**

A type of Comet entity that belongs to a SimulationProcedure and contains the simulation controls that are specific to local regions of the Component geometry. The BasicTopologyEntities defining the domain for a particular local simulation control are contained within the control’s PhysicalDomain.

Related glossary terms: *Comet entity*, *SimulationTask*, *SimulationProcedure*, *BasicTopologyEntity*, *PhysicalDomain*

- **locked Stage**

A Stage that cannot be modified, that is, cannot have Comet entities added to it, deleted from it, or edited within it. The locked Stages within a Project (from the perspective of a given Comet session) include its non-leaf (or *trunk*) Stages and the Stages that a different Comet session, including one owned by a different user, is already in.

Related glossary terms: *Project, Stage, leaf Stage, trunk Stage*

- ***LowFidelityInteraction***

The subset of Interactions that contain two ComponentRepresentations named “firstComponentRepresentation” and “secondComponentRepresentation”, two Vertices named “part1Vertex” and “part2Vertex”, and two CoordinateSystems named “firstCoordinateSystem” and “secondCoordinateSystem”. All of the Interactions relevant to MSC.Adams and CODE V are LowFidelityInteractions, and can be specified within the CAD model through an appropriate set of CAD annotations and features, called an *Interaction modeling specification*.

Related glossary terms: *Interaction, PhysicalDomain, Interaction modeling specification, CAD model, CAD annotation, ActiveTag, SimulationTask, MSC.Adams, AdamsTask, CODE V, CodeVTask*

[Go to start of Glossary](#)

M

- ***manifold geometry***

“Manifold solid models are models whose volume is well defined. They have the following properties: (1) every edge belongs to two faces; (2) every vertex is surrounded by one sequence of edges and faces; (3) faces only intersect each other in common edges on vertexes; (4) there is a material on only one side of a face. Another way of putting it is that a manifold object is a shape that you can create in real life. Non-manifold objects, by contrast, have qualities that cause badly defined volumes in your shape, making it impossible to recreate the model exactly that way in real life.” Geometry associated with a single Part is always manifold geometry.

Related glossary terms: *PrimitiveUnstructuredMesh, CompositeUnstructuredMesh, non-manifold geometry, Part*

- ***material property***

A characteristic of the material composition of an engineering system, represented in the UEM by a Property of a Section.

Related glossary terms: *Property, Section, physical property, behavioral property*

- ***MATLAB[®]***

A commercial controls and simulation tool for which Comet is likely to have an adapter in the near future. See <http://www.mathworks.com>.

- ***MayaVi***

A scientific data visualizer, developed and freely distributed by Prabhu Ramachandran, which Comet previously used as an external tool for 3-D Field visualization. MayaVi uses the freeware Visualization Toolkit (VTK) for its underlying graphics functionality. See <http://mayavi.sourceforge.net>.

Related glossary terms: *visualization, Field, external tool*

- ***menu bar***

The horizontal strip directly beneath a Session window's title bar. It contains menu pads that allow users to specify an action for Comet to perform within given categories. The currently supported menus are File, Insert, Tools, View, Window, and Help.

Related glossary terms: *GUI, Session window, title bar, menu pad*

- ***menu pad***

A GUI region that, when pushed, opens a pull-down menu providing access to a related group of Comet actions. The currently supported menus are File, Insert, Tools, View, Window, and Help.

Related glossary terms: *GUI, menu bar, hotkey*

- ***mesh element topology type***

A classification of mesh elements by their topology, that is, by the number of edges or faces in a mesh element and which of these are connected to which others. Comet supports the following types of 3-dimensional element topology types: Tet4, Tet10, Hex8 and Hex20. It supports the following types of 2-dimensional element topology types: Tri3, Tri6, Quad4 and Quad8. It supports the following types of 1-dimensional element topology types: Line2 and Line3.

Related glossary terms: *PrimitiveUnstructuredMesh, CompositeUnstructuredMesh, MeshingTask, GlobalMeshControlSet, LocalMeshControlSet, ComponentMeshControlSet*

- ***MeshBasedDomain***

The type of FieldDomain whose members form a Mesh.

Related glossary terms: *Field, FieldDomain, MeshingTask*

- ***MeshDomain***

A type of Comet entity that belongs to a MeshingTask and specifies the ComponentRepresentation to be meshed. The ComponentRepresentation is contained within the MeshDomain's PhysicalDomain.

Related glossary terms: *Comet entity, MeshingTask, PhysicalDomain, ComponentRepresentation*

- ***MeshingTask***

A type of Task that, when updated, constructs a Mesh that fits the geometry specified in its MeshDomain. The type of algorithm and the controls to be used during the meshing computation are set within the Task's MeshProcedure. The only type of MeshingTask that Comet currently supports is a CometMeshingTask-1.0.

Related glossary terms: *Task, update, MeshDomain, MeshProcedure, ComponentMeshControlSet, GlobalMeshControlSet, LocalMeshControlSet, InputPortSet, OutputPortSet, OutputRequestSet, Mesh*

- ***MeshProcedure***

A type of Comet entity that belongs to a MeshingTask and specifies the type of algorithm and the controls to be used during the meshing computation. A MeshProcedure contains a ComponentMeshControlSet, a GlobalMeshControlSet, a LocalMeshControlSet and an OutputRequestSet.

Related glossary terms: *MeshingTask*, *ComponentMeshControlSet*, *GlobalMeshControlSet*, *LocalMeshControlSet*, *OutputRequestSet*

- ***mirror*** (note the lower-case letter “m”)

A surface capable of reflecting sufficient undiffused light to form an image of an object placed in front of it. A mirror is a type of *optical entity*.

Related glossary terms: *optical entity*, *CODE V*, *optics*

- ***Mirror*** (note the upper-case letter “M”)

A type of LeafOpticalEntity that models a *mirror*. It consists of a single perfectly reflective *optical surface*.

Related glossary terms: *mirror*, *LeafOpticalEntity*, *optical surface*

- ***MNF file***

See *ModalNeutralFile*.

- ***ModalNeutralConstraintNode***

A mesh node that will be fixed during the ModalNeutral structural analysis of a particular body, which is performed by either ANSYS or MSC.Nastran as the second step in the 3-step Adams→ANSYS/MSC.Nastran→Adams process for modeling a system of flexible bodies.

Related glossary terms: *ANSYS*, *AnsysTask*, *MSC.Nastran*, *MSCNastranTask*, *MSC.Adams*, *AdamsTask*, *flexible-body dynamics analysis*, *component mode synthesis*, *ModalNeutralFile*, *MNF file*, *SpiderMeshControl*, *AuxiliaryVertexMeshControl*

- ***ModalNeutralFile*** (*MNF file*)

A type of file generated for each of multiple bodies when ANSYS or MSC.Nastran performs a ModalNeutral structural analysis. It contains data describing the body’s dynamic response for each of the requested modes of vibration. The ModalNeutral structural analysis is the second step in the 3-step Adams→ANSYS/MSC.Nastran→Adams process for modeling a system of flexible bodies.

Related glossary terms: *ANSYS*, *AnsysTask*, *MSC.Nastran*, *MSCNastranTask*, *MSC.Adams*, *AdamsTask*, *flexible-body dynamics analysis*, *component mode synthesis*, *ModalNeutralConstraintNode*, *MNF file*, *SpiderMeshControl*, *AuxiliaryVertexMeshControl*

- ***MSC.Adams***[®]

A commercial simulation tool, manufactured by MSC Software Corporation, that Comet supports through its AdamsTask. MSC.Adams[®] performs multi-body dynamics analysis. See <http://www.mscsoftware.com/products/adams.cfm>.

Related glossary terms: *AdamsTask*, *external tool*

- ***MSC.Nastran***[®]

A commercial simulation tool, manufactured by MSC Software Corporation, that Comet supports through its MSCNastranTask. MSC.Nastran[®] performs finite element analysis, particularly structural and thermal analysis. See <http://www.mscsoftware.com/products/adams.cfm>.

Related glossary terms: *MSCNastranTask*, *external tool*

- ***MSCNastranTask***

A type of SimulationTask that manages runs of the MSC.Nastran external tool. It is a type of Comet entity found within a Process. “MSCNastranTask” is an abbreviation of “MSCNastranTask-005”. See <http://www.mscsoftware.com/products/nastran.cfm>.

Related glossary terms: *MSC.Nastran*, *SimulationTask*, *Process*, *Comet entity*

[Go to start of Glossary](#)

N

- ***Nastran***

See *MSC.Nastran*[®].

- ***non-manifold geometry***

Geometry that is not manifold geometry. Geometry formed by taking the union of the geometries associated with all of the Parts in an Assembly is sometimes non-manifold geometry. Therefore, if you set the MeshingTask to “createSingleMesh” on an Assembly, the resulting PrimitiveUnstructuredMesh may be on non-manifold geometry.

Related glossary terms: *manifold geometry*, *Part*, *Assembly*, *MeshingTask*, *PrimitiveUnstructuredMesh*

- ***non-physical optical surface***

An optical “virtual surface”, such as an object surface, image surface or aperture stop surface that does not necessarily correspond to a face of a physical element. Even if there is a physical component present at the virtual surface, it does not bend light passing through it. For Comet purposes, a non-physical optical surface is a type of *optical entity*.

Related glossary terms: *optical entity*, *CODE V*, *optics*

- ***NonPhysicalOpticalSurface***

A type of LeafOpticalEntity that models a *non-physical optical surface*. It consists of a single refractive, transmissive optical surface for which the material following that surface is either “Air” or “Vacuum”. That is, a non-physical optical surface is an optical “virtual surface”, such as an object surface, image surface or aperture stop surface that does not necessarily correspond to a face of a physical element. Even if

there is a physical component present at the virtual surface, it does not bend light passing through it.

Related glossary terms: *non-physical optical surface*, *LeafOpticalEntity*, *optical surface*

- ***NumericalFieldRange***

The type of *FieldRange* whose members are real numbers.

Related glossary terms: *Field*, *FieldRange*, *Plot*

[Go to start of Glossary](#)

O

- ***object***

See *GUI object*.

- ***on-demand loading***

The practice of maintaining in memory only a stub for bulky aspects of a UEM instance, such as Mesh and Field data, until some request is made that requires access to the bulk data. At the point of that demand, the bulk data is loaded into memory.

Related glossary terms: *UEM instance*, *Mesh*, *Field*

- ***optical entity***

A component of an *optical system*, modeled within Comet's UEM as a Component and associated *LeafOpticalEntity*. Comet divides optical entities into *mirrors*, *lenses* and *non-physical optical surfaces*, which it represents by the *Mirror*, *Lens* and *NonPhysicalOpticalSurface* subtypes of *LeafOpticalEntity*, respectively. A *RayTransmissionInteraction* models the relationship between two optical entities that are "adjacent" with respect to the propagation of light rays through the system.

Related glossary terms: *CODE V*, *optical system*, *mirror*, *Mirror*, *lens*, *Lens*, *non-physical optical surface*, *NonPhysicalOpticalSurface*, *RayTransmissionInteraction*, *optics*

- ***optical surface***

A surface of interest in an *optical system*, typically marking the boundary between two volumes composed of homogeneous materials with different indexes of refraction. CODE V models an optical system as a sequence of optical surfaces. It traces rays from one optical surface to the next in the given sequence. CODE V associates various optically relevant data with each optical surface, such as the surface's shape type, shape parameters (*e.g.*, curvatures), position (displacement and tilt from the optical axis), aperture (outer extent in space), obscuration (inner extent in space), the distance from the given surface to the next one, the material filling the space from the given surface to the next one, tolerances, and mirror substrates, cements and coatings. Comet, on the other hand, models an optical system as a set of *optical entities*, each represented by a *LeafOpticalEntity*, and a set of *RayTransmissionInteractions*, each of the latter specifying the relationship between

two optical entities that are “adjacent” with respect to the propagation of light rays through the system. Three optical surfaces play special roles. These are the *object surface* (where rays originate), the *image surface* (where rays terminate), and the *aperture stop surface*.

Related glossary terms: *CODE V, optical system, OpticalSystem, optical entity, LeafOpticalEntity, RayTransmissionInteraction, optics*

- ***optical surface identification tag***

A string-valued CAD annotation of type “COMET_OPTICAL_SURFACE_NAME” or “COMET_OUTGOING_OPTICAL_SURFACE_NAME”, which is attached to a face, marking that face as being a particular optical surface within an optical system. When the CAD model is imported into CoMeT, the annotation is represented within the UEM as an ActiveTag that is attached to the corresponding Face, has a name matching the annotation type, and is of the same value as the annotation. An ActiveTag of type “COMET_OPTICAL_SURFACE_NAME” marks the Face as representing either the reflective surface of a Mirror or the incoming surface of a Lens, while an ActiveTag of type “COMET_OUTGOING_OPTICAL_SURFACE_NAME” marks the Face as representing the outgoing surface of a Lens. The CAD annotations of type “COMET_OPTICAL_SURFACE_NAME” play a second role: they establish the correspondence between a Part and a LeafOpticalEntity of type Mirror or Lens as being alternative ComponentRepresentations for the same Component. This occurs when, as a sequence file is being imported into Comet relative to an existing Assembly, the label assigned (using the “SLB” keyword) to a mirror surface or incoming lens surface by the sequence file matches the value of an ActiveTag named COMET_OPTICAL_SURFACE_NAME and attached to a Face belonging to a Part in that Assembly. The new LeafOpticalEntity is then made a child of the same Component as is that Part.

Related glossary terms: *CAD annotation, label, ActiveTag, optical surface, Face, LeafOpticalEntity, Mirror, Lens, optics*

- ***optical system***

A physical system that directs the propagation of light through it. While CODE V models an optical system as a sequence of *optical surfaces*, Comet models the same system as a set of *optical entities*, each represented by a *LeafOpticalEntity*, and a set of *RayTransmissionInteractions*, each specifying the relationship between two adjacent optical entities. The optical system as a whole is represented by a NonLeafComponentRepresentation of type *OpticalSystem*. Note that, since an optical system represents a particular sequence of optical surfaces that light might traverse within the actual physical system and alternative sequences are possible, a single solid model of an assembly of optical components can have any number of different optical systems associated with it. CODE V uses the term “lens” in place of “optical system”, while Comet uses the term “lens” to mean a particular type of optical entity.

Related glossary terms: *CODE V, optical surface, optical entity, LeafOpticalEntity, RayTransmissionInteraction, OpticalSystem, lens, optics*

- ***OpticalEntityDisturbance***

A type of Comet entity that models disturbances upon the optical surfaces of associated LeafOpticalEntities. OpticalEntityDisturbance follows the Composite design pattern. The two types of PrimitiveOpticalEntityDisturbance—*MirrorDisturbance* and *LensDisturbance*—represent disturbances to the optical surfaces of Mirrors and Lenses, respectively. A *CompositeOpticalEntityDisturbance* represents disturbances to the optical surfaces of the Mirrors and Lenses contained in the associated OpticalSystem. Currently, the only supported type of disturbance is a deformation to the optical surface’s shape, which is represented as a *Zernike polynomial* offset from the original surface by a 6-DOF “rigid-body motion”. Running a *SigfitTask* with a SIGFIT:DisturbedOpticalSystemProcedure will cause the construction of a tree of OpticalEntityDisturbances that mirrors the structure of the Task’s OpticalSystem, and the attachment of the CompositeOpticalEntityDisturbance topping that tree into the SigfitTask’s ResultSet. When OpticalEntityDisturbances are linked into the *SimulationComponentDisturbanceSet* of a *CodeVTask*, CODE V will take the surface deformations into account in its analysis.

Related glossary terms: *Comet entity, optical surface, LeafOpticalEntity, Mirror, Lens, OpticalSystem, Zernike polynomial, SigFit, SigfitTask, SimulationComponentDisturbanceSet, CODE V, CodeVTask, optics*

- ***OpticalSystem***

A type of NonLeafComponentRepresentation that models an optical system. Its children consist of *LeafOpticalEntity* objects, which are divided into *Mirror, Lens* and *NonPhysicalOpticalSurface* subclasses, and the same number of *RayTransmissionInteractions*, each of which specifies the relationship between two adjacent LeafOpticalEntities. The RayTransmissionInteractions thus order the LeafOpticalEntities into a sequence, which gives the order of optical surfaces through which light would traverse the system. A RayTransmissionInteraction also specifies the distance between its source and destination LeafOpticalEntities, and the material filling the space between them. An OpticalSystem typically is constructed by importing a CODE V *sequence file* relative to either the PhysicalModel or an existing Assembly. In the latter case, corresponding ComponentRepresentations within the Assembly and OpticalSystem are made children of the same Component, reflecting that they are alternative computational models for the same physical element. The SimulationComponent for *CodeVTasks* and *SigfitTasks* is required to be an OpticalSystem.

Related glossary terms: *NonLeafComponentRepresentation, optical system, LeafOpticalEntity, Mirror, Lens, NonPhysicalOpticalSurface, RayTransmissionInteraction, optical surface, sequence file, PhysicalModel, Assembly, SimulationComponent, CODE V, CodeVTask, SigFit, SigfitTask, optics*

- ***optics***

A branch of physics that describes the behavior and properties of light and the interaction of light with matter. The dual nature of light—it is both wave and particle—leads optics to be broken into two branches—classical optics and modern optics. CODE V models classical optics, the study of light as electromagnetic radiation described by Maxwell’s equations. CODE V primarily describes light propagation in terms of rays, which bend at the interface between two dissimilar media. To a lesser extent, it models the propagation of complex wavefronts through

optical systems, including both the amplitude and phase of the wave, which allows diffraction, interference and polarization to be taken into account.

Related glossary terms: *CODE V*, *optical system*, *optical surface*, *optical entity*, *mirror*, *lens*, *non-physical optical surface*, *integrated optomechanical analysis*

- ***optomechanical analysis***

See *integrated optomechanical analysis*.

- ***OutputPort***

A type of Comet entity that belongs to a Task and outputs data in the form of a Value across the DataPipe to which the OutputPort is connected. At the end of the Task's update, for each of the Task's OutputPorts, a datum is extracted from the Comet model at a location specified relative to the Task and copies of the datum are pushed across any DataPipes connected to the corresponding OutputPort and placed on the DataPipes' receiving Ports.

Related glossary terms: *Comet entity*, *Value*, *Task*, *DataPipe*, *OutputPortSet*

- ***OutputPortSet***

A type of Comet entity that is attached directly to a Task and contains all the OutputPorts belonging to that Task.

Related glossary terms: *Comet entity*, *Task*, *OutputPort*

- ***OutputRequest***

A type of Comet entity that belongs to a Task and contains requests for the Task to either generate particular results beyond the default set or control the production or formatting of some results in a particular way.

Related glossary terms: *Comet entity*, *MeshingTask*, *SimulationTask*, *OutputRequestSet*

- ***OutputRequestSet***

A type of Comet entity that belongs to a Task (or to a specific Procedure within a Task for those types of Task that have multiple Procedures) and contains all the OutputRequests belonging to that Task (or Procedure).

Related glossary terms: *Comet entity*, *MeshingTask*, *MeshProcedure*, *SimulationTask*, *SimulationProcedure*, *ExcelTask*, *OutputRequest*

- ***owned Property***

A Property that is directly linked to the given source entity.

Related glossary terms: *Property*, *transferred Property*, *inherited Property*

[Go to start of Glossary](#)

P

- ***pane***

A division of a GUI window that has its own function, usually to display a category of information. For example, editor windows are divided into multiple panes to permit the user to quickly see (and modify) by category the Properties of a Comet entity. Comet separates adjacent panes within a single window by splitters that can be dragged to interactively reappportion the relative areas of the two panes.

Related glossary terms: *GUI, editor, splitter*

- ***Parameter***

A type of Comet entity that associates a Value with a name. Parameters are typically referenced by name and with respect to a source entity. The name would then be dereferenced by performing a search that uses the given source entity as a starting point, travels the UEM instance in the direction of the source entity's ancestors, and finds the first Parameter of the given name attached to a Comet entity on the search path.

Related glossary terms: *Comet entity, Value, Comet entity, UEM instance*

- ***Parasolid[®]***

A commercial 3-D modeling and computational geometry library, manufactured by UGS Corporation, which is linked into Comet. Comet supports the import and export of solid models in the Parasolid format. See <http://www.ugs.com/products/open/parasolid>.

Related glossary terms: *solid model, component software*

- ***ParaView***

An open-source, multi-platform visualization application, which Comet uses as an external tool for 3-D Field visualization. ParaView was created by Kitware in conjunction with Jim Ahrens of Los Alamos National Laboratory (LANL). Contributors currently include Kitware, LANL, Sandia National Laboratories, and the Army Research Laboratory. ParaView uses the freeware Visualization Toolkit (VTK) for its underlying graphics functionality. See <http://www.paraview.org>.

Related glossary terms: *visualization, Field, external tool, Visualization Toolkit*

- ***Part***

A type of ComponentRepresentation that models a physical part in an engineering system. A Part is always attached to a leaf Component in a Component hierarchy, and stores the numerical data that represents that Component to some category of simulation code. Parts, along with Interactions, comprise the leaf nodes of an Assembly hierarchy (one providing a consistent computational model for the aforementioned Component hierarchy), while Assemblies comprise the Assembly hierarchy's internal nodes. Each Part, Interaction and Assembly in an Assembly hierarchy has a CoordinateSystem; the CoordinateSystems are themselves linked into a hierarchy that exactly parallels the Assembly hierarchy, where the CoordinateSystems indicate the spatial location of each Part, Interaction or Assembly relative to its parent and the root ComponentRepresentation's CoordinateSystem is linked to the PhysicalModel's CoordinateSystem—the reference coordinate system for the entire PhysicalModel. A Part can belong to only one Assembly hierarchy; that is, a Part is either the root of an Assembly hierarchy or directly contained (*i.e.*, parented) by exactly one Assembly.

A Part can have geometry, contained within a ComponentGeometry, although that is not required; a Part without geometry usually models its Component as lumped parameters, that is, numerical values that capture the behavior of the Component as a whole. Parts that have geometry contain a Section and a ComponentMesh. The Section contains the Part's geometry and material property distribution in the topological dimensions that are **not** in the Part's ComponentGeometry, that is, that are **not** discretized for analysis; in this case, the material and behavioral properties associated with the Part are stored as Properties of the Section. The ComponentMesh contains any number of Meshes generated to fit the ComponentGeometry.

Related glossary terms: *PhysicalModel*, *Component hierarchy*, *ComponentRepresentation*, *Part*, *Interaction*, *Assembly*, *CoordinateSystem*, *CAD model*

- *physical property*

A characteristic of the physical embodiment of an engineering system design. Material properties and behavioral properties are subsets of the physical properties of a system.

Related glossary terms: *Property*, *material property*, *behavioral property*

- *PhysicalDomain*

A type of Comet entity that represents the geometric domain over which some action or condition is to be applied. For example, the MeshDomain of a MeshingTask contains a PhysicalDomain that specifies the geometry to which the mesh constructed by the MeshingTask is to be fit. The SimulationDomain of a SimulationTask contains a PhysicalDomain that specifies the geometry extent to be simulated. EnvironmentEntities, representing boundary conditions, initial conditions, prescribed conditions and loads, each contain a PhysicalDomain that specifies the domain over which the condition is defined. A PhysicalDomain defines a set of BasicTopologyEntities. This set can be specified either as an explicit list of particular BasicTopologyEntities or as an AbstractDomain that dynamically binds to BasicTopologyEntities of an appropriate type.

Related glossary terms: *Comet entity*, *PhysicalModel*, *ComponentRepresentation*, *Interaction*, *EnvironmentEntity*, *boundary condition*, *load*, *initial condition*, *prescribed condition*, *MeshDomain*, *SimulationDomain*, *BasicTopologyEntity*, *abstract modeling*, *AbstractDomain*

- *physicalDomainFilterType*

A Property of an AbstractDomain that specifies the types of Comet entity to which the AbstractDomain can bind. The valid values for the *physicalDomainFilterType* Property are "Component", "ComponentRepresentation", "Interaction", "RefVolume", "Face", "Edge" and "Vertex".

Related glossary terms: *abstract modeling*, *AbstractDomain*, *Property*, *BoundEntitySet*, *EntireDomain*, *Component*, *ComponentRepresentation*, *Interaction*, *RefVolume*, *Face*, *Edge*, *Vertex*

- *PhysicalModel*

A top-level Comet entity, which is the root of that portion of a UEM instance that captures those aspects of an engineering system design required to simulate it. This

includes all the information in a CAD model, such as geometry and shape parameters, the system's material properties, meshes, auxiliary geometry, construction geometry, and interactions between the system's components. Every Project has exactly one PhysicalModel.

Related glossary terms: *Comet entity, UEM instance, top-level entity, Project, CAD model, Component, ComponentRepresentation, CoordinateSystem, Component hierarchy, ComponentRepresentation hierarchy, CoordinateSystem hierarchy, global coordinate system*

- ***PLM***

See *Product Lifecycle Management*.

- ***Plot***

The type of Field that has a GridNodeBasedDomain and a NumericalFieldRange. Currently, the supported *PlotTypes* are “2DPlot”, “3DPlot” and “4DPlot”.

Related glossary terms: *Field, GridNodeBasedDomain, NumericalFieldRange, PlotType, 2DPlot, 3DPlot, 4DPlot*

- ***PlotType***

A Plot Property that indicates the dimensionality of the Plot's FieldDomain. Currently, the supported *PlotTypes* are “2DPlot”, “3DPlot” and “4DPlot”.

Related glossary terms: *Field, Plot, 2DPlot, 3DPlot, 4DPlot*

- ***prescribed condition***

For a given simulation method that involves solving a set of physics equations, the values of the non-solution variables over the entire PhysicalModel and over the duration of the analysis, where the solution variables somehow depend upon the non-solution variables. For example, material properties relevant to structural analysis, such as Young's Modulus and Poisson's Ratio, can be dependent upon temperature. A prescribed condition for a structural analysis might therefore specify the ambient temperature over the simulation's spatial and temporal domains. The UEM represents a prescribed condition as an EnvironmentEntity that is a member of some Environment's PrescribedConditionSet.

Related glossary terms: *solution variable, PhysicalModel, UEM, EnvironmentEntity, Environment, PrescribedConditionSet*

- ***PrescribedConditionSet***

A type of Comet entity whose parent is an Environment and that contains the set of EnvironmentEntities playing the role of prescribed condition within that Environment.

Related glossary terms: *Comet entity, Environment, EnvironmentEntity, prescribed condition*

- ***PrimitiveUnstructuredMesh***

A type of Comet entity that represents a single unstructured mesh. A PrimitiveUnstructuredMesh can be fit to a manifold geometry, such as that of a single part in a CAD assembly. Alternatively, a PrimitiveUnstructuredMesh can be fit to a

non-manifold geometry, such as that internally produced by the MeshingTask when it merges the Parts in an Assembly as a preliminary step to meshing that geometry (this only occurs when the “createSingleMesh” control is set to true).

Related glossary terms: *Comet entity, ComponentMesh, ComponentGeometry, CompositeUnstructuredMesh, manifold geometry, non-manifold geometry, MeshingTask*

- ***Process***

A type of Comet entity that represents an engineering analysis process as a dataflow network whose nodes are Tasks, where each Task represents a single step within the process. A Process contains a ProcessInputPortSet and a ProcessOutputPortSet, which contain the Process’ ProcessInputPorts and ProcessOutputPorts, respectively. DataPipes connect ProcessInputPorts with Task-level InputPorts, Task-level OutputPorts with Task-level InputPorts, and Task-level OutputPorts with ProcessOutputPorts. When a Process is *run*, TopMan causes the Tasks to be *updated* and control signals and data to flow downstream along the DataPipes in an order determined dynamically by dataflow constraints.

Related glossary terms: *Comet entity, dataflow, ProcessSet, run, ProcessInputPort, ProcessOutputPort, ProcessInputPortSet, ProcessOutputPortSet, Task, update, InputPort, OutputPort, DataPipe, ToPMan*

- ***ProcessInputPort***

A type of Comet entity that belongs to a Process and allows the user to supply run-specific data, in the form of Values, to the Process. ProcessInputPorts have their data values set by the user before the Process is run. ProcessInputPorts can be connected to Task-level InputPorts through DataPipes, in which case, copies of the ProcessInputPorts’ data are placed within the connected Task-level InputPorts when the Process is run.

Related glossary terms: *Comet entity, Process, DataPipe, ProcessInputPortSet*

- ***ProcessInputPortSet***

A type of Comet entity that is attached directly to a Process and contains all the ProcessInputPorts belonging to that Process.

Related glossary terms: *Comet entity, Process, ProcessInputPort*

- ***ProcessOutputPort***

A type of Comet entity that belongs to a Process and allows data generated by a Process run, in the form of Values, to persist and be easily extracted. ProcessOutputPorts can be connected to Task-level OutputPorts through DataPipes. After a Task has updated, for each of the Task’s OutputPorts, a datum is extracted from the Comet model at a location specified relative to the Task and copies of the datum are pushed across any DataPipes connected to the corresponding OutputPort and placed on the DataPipes’ receiving Ports. In the event that a receiving Port was a ProcessOutputPort, the received data will remain there to be read after the Process run.

Related glossary terms: *Comet entity, Process, DataPipe, ProcessOutputPortSet*

- ***ProcessOutputPortSet***

A type of Comet entity that is attached directly to a Process and contains all the ProcessOutputPorts belonging to that Process.

Related glossary terms: *Comet entity, Process, ProcessOutputPort*

- **Process Schematic**

A GUI window that visually represents the elements of a Process. Process Schematics are drawn onto the Canvas.

Related glossary terms: *GUI, Process, Canvas*

- **ProcessSet**

A top-level Comet entity, which contains all of the Processes defined within a UEM instance. Every Project has exactly one ProcessSet.

Related glossary terms: *Comet entity, UEM instance, top-level entity, Process*

- **Product Lifecycle Management (PLM)**

A strategic business approach to solving the problem of managing the complete set of product definition information—creating that information, managing it, and disseminating and using it throughout the lifecycle of the product. Implementing a PLM solution requires both software tools, including CAD and CAE, and business processes, including Simulation-Driven Engineering.

Related glossary terms: *Computer Aided Engineering, Computer Aided Design, Simulation Driven Engineering*

- **Pro/ENGINEER®**

A commercial CAD design tool, manufactured by Parametric Technology Corporation (PTC). Comet supports the import and export of Pro/ENGINEER parametric solid models for CAD. Comet sessions maintain a live, bi-directional link to a Pro/ENGINEER process whenever the current Project has an imported Pro/E model. See <http://www.ptc.com>.

Related glossary terms: *CAD, external tool*

- **Project**

The central repository for all of the models, environment specifications, process specifications, and results data associated with the analysis of an engineering system over the course of its design. The Project captures the iterative, incremental history of changes to the models and specifications as a tree of Stages, where each Stage contains an instance of Comet's Universal Engineering Model.

Properties of a Project include handles to the root Stage, the current Stage, the root directory, the Project's title, and the top-level Comet entities—currently, the PhysicalModel, the AbstractDomainSet, the EnvironmentSet, the ProcessSet and the UserDefinedFieldSet.

A Comet session opens a Project (just one per session), and allows it to be interactively examined, modified, and operated on.

Related glossary terms: *Universal Engineering Model, UEM instance, Stage, root Stage, current Stage, root directory, top-level entity, Property, Project management system, session*

- ***Project management system***

The subsystem of the Comet software that manages operations upon a Project. These operations include constructing a new Project, saving a Project, opening a saved Project, adding a new Stage, moving to a Stage, removing a branch of the Project, and un-doing and re-doing the state of the current Stage.

Related glossary terms: *Comet, Project, Stage*

- ***project view***

A dockable GUI window, usually represented with tabs, that contains information of some particular category about a Project. The currently supported types of project views are *Project, PhysicalModel, AbstractDomain, Environment, Process, and Misc. Data.*

Related glossary terms: *GUI, dockable window, Project, PhysicalModel, AbstractDomainSet, EnvironmentSet, ProcessSet, UserDefinedFieldSet*

- ***Property***

For a given Comet entity (called the *source entity*), a *SingleValuedDynamicEntity, CompositeDynamicEntity, Expression* or *Field* that is returned when the source entity is queried and that is identifiable by key. The query can request a single *Property* by key or ask for all of them, in which case, a map from keys to *Properties* is returned. Each of a source entity's *Properties* can be categorized according to ownership as being an owned *Property*, a transferred *Property* or an inherited *Property*, and can be categorized according to origin as being either a declared *Property* or a derived *Property*.

Related glossary terms: *Comet entity, SingleValuedDynamicEntity, Expression, Field, owned Property, transferred Property, inherited Property, declared Property, derived Property*

- ***PropertyBasedFieldRange***

<TBD>

Related glossary terms: *Field, FieldRange*

- ***PuTTY***

PuTTY is a free implementation of Telnet and SSH for Win32 and Unix platforms, along with an `xterm` terminal emulator. It is written and maintained primarily by [Simon Tatham](http://www.putty.nl). Comet uses PuTTY's SSH capabilities to implement remote launching of external tools. See <http://www.putty.nl>.

Related glossary terms: *LauncherType, remote launch, external tool*

[Go to start of Glossary](#)

Q

- ***Query window***

A type of dockable window that permits the user to search the current Stage's UEM instance with respect to the given source entity, a selected search direction, and a selected Comet entity type. For example, you may query the descendants of a Component object for entities of type SolidSection, find a SolidSection in the Results box of the Query window, and click SolidSection in that box to highlight the desired SolidSection object in the PhysicalModel project view.

Related glossary terms: *GUI, dockable window, UEM instance, Comet entity*

[Go to start of Glossary](#)

R

- **Range**

For a given SingleValuedDynamicEntity type, the set of all values to which the Values inside instances of that type can be set. A Range is the union of a set of Intervals. For StringValues, each Interval is a single string literal. For the other types of Values, each Interval is the set of real or integral numbers between two bounding numbers, where the Interval can be defined to either include or exclude each of the bounding numbers.

Related glossary terms: *SingleValuedDynamicEntity, Value*

- **RayTransmissionInteraction**

A type of Interaction that models the relationship between two optical entities, each represented by a LeafOpticalEntity. A RayTransmissionInteraction specifies the identities of its source and destination LeafOpticalEntities, the distance between them, and the material filling the space between them. Collectively, the set of RayTransmissionInteractions associated with an OpticalSystem order the LeafOpticalEntities associated with that same OpticalSystem into a sequence.

Related glossary terms: *Interaction, optical entity, LeafOpticalEntity, OpticalSystem, optics*

- **reference coordinate system**

The CoordinateSystem attached directly to the PhysicalModel. The placement of all other CoordinateSystems is defined relative to the reference coordinate system. Other names for it are *base coordinate system* and *global coordinate system*.

Related glossary terms: *PhysicalModel, CoordinateSystem, base coordinate system, global coordinate system, local coordinate system*

- **RefVolume**

The type of BasicTopologyEntity that represents 3-dimensional geometric elements.

Related glossary terms: *BasicTopologyEntity*

- **root directory**

<TBD>

Related glossary terms: *Project*

- ***root Stage***

The root of the tree of Stages comprising a Project. When a new Project is opened, it initially contains only the root Stage, which is empty aside from the handful of top-level entities.

Related glossary terms: ***Project, Stage, top-level entity***

- ***run***

Given a Process, the action of having ToPMan “execute” the Process according to dataflow constraints. That is, ToPMan initializes its *execution queue* by scheduling those Tasks that either have no InputPorts or only InputPorts connected to ProcessInputPorts to be updated. It then moves data across DataPipes, schedules Tasks for updating as they become eligible, and updates Tasks in the order upon which they appear on the execution queue until the execution queue is empty.

Related glossary terms: ***Process, Task, update, execution queue, dataflow, ToPMan, DataPipe***

[Go to start of Glossary](#)

S

- ***sample Value***

For a given SingleValuedDynamicEntity type that has been defined to not have a default Value, a sample Value that demonstrates the unit and algebraic type with which new instances of the type must be compatible.

Related glossary terms: ***SingleValuedDynamicEntity, Value, default Value***

- ***Scalar***

A type of Value that combines a numerical quantity and a unit, where the numerical component can be either a real number or an integer.

Related glossary terms: ***Value***

- ***Scene***

A set of drawable objects.

Related glossary terms: ***GUI, drawable object, Scene window, Viewport***

- ***Scene window***

A window in which a Viewport is rendered, *i.e.*, visualized. Scene windows are drawn onto the Canvas.

Related glossary terms: ***GUI, drawable object, Scene, Viewport***

- ***SDE***

See ***Simulation Driven Engineering***.

- ***Section***

Given an actual, 3-D physical part that is being modeled, a type of Comet entity that represents the part's geometry and material property distribution in the topological dimensions that are **not** in the Part's ComponentGeometry, that is, that are **not** discretized for analysis. A Part contains up to one Section, where it is a sibling of the Part's ComponentGeometry; the material and behavioral properties associated with the Part are stored as Properties of the Section. The types of Section are 2-dimensional BeamSection, 1-dimensional ShellSection, and 0-dimensional SolidSection. For example, in the case of beam analysis, the ComponentGeometry is 1-dimensional and represented by an Edge running along the length of the beam. The BeamSection's geometry is therefore a 2-dimensional Face; part of the Section specification is the material distribution across that Face. Sweeping the BeamSection (the Face) along the ComponentGeometry's Edge defines the full 3-D geometry and material distribution of the beam.

Related glossary terms: *Comet entity, PhysicalModel, Part, ComponentGeometry, BeamSection, ShellSection, SolidSection, material property*

- **Section geometry**

The geometry defined by a Section. Its topological dimensionality is three minus that of the ComponentGeometry. The geometry of a BeamSection is a Face of pre-defined parameterized 2-dimensional shape determined by the type of PrimitiveBeamSection that it is. The geometry of a ShellSection is an ordered list of Layer thicknesses (currently limited to one Layer). The geometry of a SolidSection, being a single Vertex, does not require explicit definition.

Related glossary terms: *Comet entity, Section, ComponentGeometry, BeamSection, PrimitiveBeamSection, ShellSection, Layer, Layup, SolidSection, Vertex*

- **sequence file (.seq file)**

A script file of CODE V commands. Comet uses sequence files in three ways: (1) **Comet users typically specify optical system models within Comet by importing an existing sequence file, which is the result of an optical designer interactively developing the optical system design in CODE V and exporting it from there as a sequence file.** Importing the sequence file causes additions to the current UEM instance that cover all model data specified within the sequence file—a new Component/ComponentRepresentation hierarchy models the optical system, a new CodeVTask sets up a mechanism for optics analysis of the optical system, and a new Environment defines the set of test rays that are applied to the optical system in order to assess its behavior. Currently, requests for particular types of analysis must be specified from within CoMeT (as OutputRequests) and cannot be imported from a sequence file. (2) **Comet drives CODE V using a Comet-generated sequence file.** When a CodeVTask is updated, it generates a new sequence file derived from the CoMeT session's current UEM instance and launches CODE V in non-interactive mode with instructions to load the newly generated sequence file. This sequence file defines the optical system model (this part of the Comet-exported sequence file essentially reconstructs the user-provided sequence file), runs analyses, and exports their results into output files. (3) **SigFit exports its surface-disturbance-fitting results as tiny sequence files (plus interferogram files referenced by the sequence file) that augment the optical surface shape specifications.** Comet imports these files as *OpticalEntityDisturbances*.

Related glossary terms: *CODE V, optical system, UEM instance, Component hierarchy, OpticalSystem, CodeVTask, Environment, SigFit, SigfitTask, interferogram file, optical surface, OpticalEntityDisturbance, optics*

- ***Session***

The combination of a Job process and the GUI window (called a *Session window*) through which the user interacts with that Job. Note that, when Comet is launched, a single GUI process is started. It immediately creates the initial Session. Additional Sessions can be started, which are controlled by the original and sole GUI process. Each Session started by the same GUI process has a different Job number, *e.g.*, Job-1, Job-2, Job-3.

Related glossary terms: *GUI, Job, Session window*

- ***Session window***

The main GUI window associated with a particular Session. When Comet starts up, a Session window for the initial Session pops up. If additional Sessions are started, a new Session window is opened for each one.

Related glossary terms: *GUI, Session*

- ***shape parameter***

A parameter in a parametric feature-based CAD or solid model that affects the shape of the modeled geometry.

Related glossary terms: *CAD model, solid model*

- ***ShellSection***

The type of Section used for “shell analysis”, in which the ComponentGeometry is topologically 2-dimensional and the Section geometry is topologically 1-dimensional. In future Comet releases, ShellSection will contain a Layup that defines an ordered list of Layers, each of which specifies the thickness and material composition of one homogeneous layer of material. Currently, a ShellSection directly contains just one Layer, where the Layer’s material properties and thickness are represented as Properties.

Related glossary terms: *PhysicalModel, Part, ComponentGeometry, Section, BeamSection, SolidSection, Layer, Layup, Property, material property*

- ***SI base unit***

One of the set of seven dimensionally independent units forming the basis functions of the SI unit system. For the seven base quantity types, the SI base units are as follows. Length: meter or *m*. Mass: kilogram or *kg*. Time: second or *s*.

Thermodynamic temperature: degrees Kelvin or *K*. Luminous intensity: candela or *cd*. Quantity of matter: mole or *mol*. Electrical current: ampere or *A*.

Related glossary terms: *unit of measure, base quantity types, SI unit system*

- ***SI unit system***

The “International System of Units”, a modern form of the metric system and the world’s most widely used system of units of measure. It defines seven SI base units from which all other physical units are derived. It is Comet’s default unit system.

Related glossary terms: *unit of measure, base quantity types, SI base unit*

- **SigFit[®]**

A commercial data translation and simulation tool, manufactured by Sigmadyne, Inc., which Comet supports through its SigfitTask. SigFit transforms results from mechanical finite element analyses into result import formats for optical analysis tools. It is also a low-end optical analysis tool in its own right, capable of generating some subset of the optical performance results that CODE V can generate. It is a critical component of Comet's support for integrated optomechanical analysis. See <http://www.sigmadyne.com>.

Related glossary terms: *SigfitTask, CODE V, CodeVTask, optics, external tool, integrated optomechanical analysis*

- **SigfitTask**

A type of SimulationTask that manages runs of the SigFit external tool. It is a type of CoMeT entity found within a Process. "SigfitTask" is an abbreviation of "SigfitTask-2007". See <http://www.sigmadyne.com>.

Related glossary terms: *SigFit, SimulationTask, Process, CoMeT entity, integrated optomechanical analysis, optical system, OpticalSystem, optical surface, optical surface identification tag, optical entity, LeafOpticalEntity, mirror, Mirror, lens, Lens, non-physical optical surface, NonPhysicalOpticalSurface, RayTransmissionInteraction, CODE V, CodeVTask, OpticalEntityDisturbance, MirrorDisturbance, LensDisturbance, SimulationComponentDisturbanceSet, Zernike polynomial, optics*

- **Simulation Driven Engineering (SDE)**

An approach to product engineering in which up-front simulation results and simulation studies throughout the engineering process drive design decisions. Important capabilities for driving SDE include the modeling of product behavior at the abstract level, engineering process integration, and bi-directional links between CAD and CAE.

Related glossary terms: *Computer Aided Engineering, Computer Aided Design, Virtual Product Engineering, Comet*

- **SimulationComponent**

A type of Comet entity that belongs to a SimulationTask's SimulationDomain and specifies the ComponentRepresentation to be simulated. The ComponentRepresentation is contained within the SimulationComponent's PhysicalDomain.

Related glossary terms: *Comet entity, SimulationTask, SimulationDomain, PhysicalDomain, ComponentRepresentation*

- **SimulationComponentDisturbanceSet**

A type of Comet entity that optionally belongs to a SimulationTask's SimulationDomain and specifies any disturbances upon the ComponentRepresentation to be simulated. Currently, the only supported type of disturbance is an *OpticalEntityDisturbance*, which represents deformations upon the shapes of optical surfaces contained in an OpticalSystem, and the only type of Task

taking the `SimulationComponentDisturbanceSet` into account is a `CodeVTask`. A `SigfitTask` can generate an `OpticalEntityDisturbance` as a result; it fits a smooth analytical surface (a Zernike polynomial) to the discrete mesh-based “nodalDisplacements” field generated by an FEA structural analysis code.

Related glossary terms: *Comet entity, SimulationTask, SimulationDomain, ComponentRepresentation, OpticalEntityDisturbance, optical surface, OpticalSystem, CodeVTask, SigfitTask, Zernike polynomial, optics*

- ***SimulationDomain***

A type of Comet entity that belongs to a `SimulationTask`. It contains the `SimulationComponent` whose `PhysicalDomain` specifies the `ComponentRepresentation` to be simulated, and the `SimulationMesh` whose `PhysicalDomain` contains the Mesh to be used in the simulation.

Related glossary terms: *Comet entity, SimulationTask, SimulationComponent, SimulationMesh, PhysicalDomain, ComponentRepresentation*

- ***SimulationMesh***

A type of Comet entity that belongs to a `SimulationTask`'s `SimulationDomain` and specifies the Mesh to be used in the simulation.

Related glossary terms: *Comet entity, SimulationTask, SimulationDomain, Mesh, PrimitiveUnstructuredMesh, CompositeUnstructuredMesh, ComponentRepresentation, SimulationComponent, PhysicalDomain*

- ***SimulationProcedure***

A type of Comet entity that belongs to a `SimulationTask` and specifies the type of algorithm and the controls to be used during one phase of the simulation computation. A `SimulationTask` potentially contains any number of `SimulationProcedures`, specifying a series of computational phases. Each `SimulationProcedure` contains a `SimulationEnvironment`, which contains the Environment to be applied to the `PhysicalModel` during that phase of the simulation, a `GlobalSimulationControlSet`, a `LocalSimulationControlSet`, and an `OutputRequestSet`.

Related glossary terms: *Comet entity, SimulationTask, SimulationEnvironment, GlobalSimulationControlSet, LocalSimulationControlSet, OutputRequestSet, InitialConditionSet*

- ***SimulationTask***

A type of Task that, when updated, launches an external simulation tool to perform a simulation. The `SimulationTask` contains a `SimulationDomain`. It contains the `SimulationComponent` whose `PhysicalDomain` specifies the `ComponentRepresentation` to be simulated, and the `SimulationMesh` whose `PhysicalDomain` contains the Mesh to be used in the simulation. The `SimulationTask` also contains some number of `SimulationProcedures`, which specify the type of algorithm and the controls to be used during each phase of the simulation process. Each `SimulationProcedure` contains a `SimulationEnvironment`, which contains the Environment to be applied to the `PhysicalModel` during that phase of the simulation, a `GlobalSimulationControlSet`, a `LocalSimulationControlSet`, and an `OutputRequestSet`. The `SimulationTask` also contains an `InitialConditionSet`, which specifies the set of `InitialConditions` to be applied at the start of the entire simulation

run. The solutions to that first analysis phase then supply the initial conditions for the next one, and so forth for the remainder of the chain of analysis phases specified by the Procedures. The currently supported types of SimulationTask are AbaqusTask, AdamsTask, AnsysTask, CodeVTask, FluentTask and MSCNastranTask.

Related glossary terms: *Task, update, external tool, SimulationDomain, SimulationComponent, PhysicalDomain, ComponentRepresentation, SimulationMesh, Mesh, SimulationProcedure, SimulationEnvironment, Environment, GlobalSimulationControlSet, LocalSimulationControlSet, OutputRequestSet, InitialConditionSet, InputPortSet, OutputPortSet, AbaqusTask, AdamsTask, AnsysTask, CodeVTask, FluentTask, MSCNastranTask*

- **SingleValuedDynamicEntity**

The most common class of DynamicEntity, members of which contain a single Value. A SingleValuedDynamicEntity type sets whether the type has a default Value (if not, it will instead have a sample Value), the default or sample Value, and the Value range for all instances of the given type.

Related glossary terms: *DynamicEntity, Value, default Value, sample Value, Range*

- **solid model**

The unambiguous representation of the solid parts of an object, that is, models of solid objects suitable for computer processing.

Related glossary terms: *Boundary Representation solid model, BREP, Constructive Solid Geometry, CSG, CAD model, ComponentGeometry, AuxiliaryGeometry, ConstructionGeometry, BasicTopologyEntity, Pro/ENGINEER, SolidWorks, ACIS, Parasolid*

- **SolidSection**

The type of Section used for “solid analysis”, in which the ComponentGeometry is topologically 3-dimensional and the Section geometry is topologically 0-dimensional. Comet currently limits its support for SolidSections to homogeneous material composition, where the Section’s material properties are represented as Properties. The geometry of a SolidSection, being a single Vertex, is defined implicitly.

Related glossary terms: *PhysicalModel, Part, ComponentGeometry, Section, BeamSection, ShellSection, Property, material property*

- **SolidWorks®**

A commercial CAD design tool, manufactured by SolidWorks Corporation. Comet supports the import and export of SolidWorks parametric solid models for CAD. Comet sessions maintain a live, bi-directional link to a SolidWorks process whenever the current Project has an imported SolidWorks model. See <http://www.solidworks.com>.

Related glossary terms: *CAD, external tool*

- **solution variable**

For a given simulation method that involves solving a set of physics equations, an output of the solution, *i.e.*, one of the variables being solved for.

Related glossary terms: *Environment, BoundaryCondition, Load, InitialCondition, PrescribedCondition*

- ***SpiderMeshControl***

A type of Comet entity that belongs to a MeshProcedure and contains the meshing controls that specify the generation of a “spider Mesh”. This type of Mesh is used primarily during the ModalNeutral structural analysis of a body, which is performed by ANSYS or MSC.Nastran as the second step in the 3-step

Adams→ANSYS/MSC.Nastran →Adams process for modeling a system of flexible bodies.

Related glossary terms: *Comet entity, MeshingTask, MeshProcedure, ANSYS, AnsysTask, MSC.Nastran, MSCNastranTask, MSC.Adams, AdamsTask, flexible-body dynamics analysis, component mode synthesis, ModalNeutralFile, MNF file*

- ***splitter***

A movable bar that separates a pair of adjacent windows. Dragging a splitter changes the relative apportionment of area between the windows separated by the splitter.

Related glossary terms: *GUI*

- ***SpreadsheetTask***

A type of Task that manages runs of a spreadsheet external tool. It is a type of Comet entity found within a Process.

Related glossary terms: *Excel, ExcelTask, Task, Process, Comet entity*

- ***Stage***

The representation within a Project of a named, reconstructable state in the history of a Virtual Product Engineering project. It is an instance of the UEM.

Related glossary terms: *Project, UEM instance, Virtual Product Engineering*

- ***status bar***

The horizontal strip at the bottom of a Session window in which status information is displayed.

Related glossary terms: *GUI, Session window*

- ***StringValue***

A type of Value that contains a single string.

Related glossary terms: *Value*

- ***system menu***

A menu that provides you with controls (via mouse or hotkey) for a particular window. It is usually accessed by right-clicking (or clicking at the far left of) the title bar of the window. A Session window’s system menu can also be accessed by ALT+SPACEBAR.

Related glossary terms: *GUI, title bar, hotkey, Session window, windows control buttons*

T

- **Task**

A type of Comet entity that represents a single step within an engineering analysis process. A Task contains an InputPortSet and an OutputPortSet, which contain its InputPorts and OutputPorts, respectively. DataPipes connect ProcessInputPorts with Task-level InputPorts, Task-level OutputPorts with Task-level InputPorts, and Task-level OutputPorts with ProcessOutputPorts. When a Task is *updated*, it performs a computation. Some types of Task call upon external tools, such as a mesher or a simulation package, to perform the actual work, where the Task can run the external tool either locally or remotely. Other types of Task use strictly internal Comet functionality to carry out their work. Currently supported types of Task include MeshingTasks, SimulationTasks, CmtBaseEntityVisualizationTasks, FieldVisualizationTasks, UserDefinedTasks, SpreadsheetTasks, and DesignSpaceExplorationTasks.

Related glossary terms: *Comet entity, Process, update, external tool, LauncherConfiguration, LauncherType, InputPort, OutputPort, InputPortSet, OutputPortSet, ProcessInputPort, ProcessOutputPort, DataPipe, execution queue, MeshingTask, SimulationTask, CmtBaseEntityVisualizationTask, FieldVisualizationTask, UserDefinedTask, SpreadsheetTask, ExcelTask, DesignSpaceExplorationTask, DakotaTask, AbaqusTask, AdamsTask, AnsysTask, CodeVTask, FluentTask, MSCNastranTask*

- **Terminal window**

A dockable GUI window, usually represented with tabs, that allows users to interact with Comet textually. It displays messages from Comet, and allows users to enter CML commands. These are executed when the user enters a carriage return. Until that point, the command-line text can be navigated and edited using all the usual keyboard functions.

Related glossary terms: *GUI, dockable window, CML*

- **Thermal Desktop[®]**

A commercial design and simulation tool, manufactured by Cullimore & Ring Technologies, Inc., for which Comet support is planned for the near future. Thermal Desktop performs thermal modeling of electronics and vehicles. See <http://www.thermaldesktop.com>.

Related glossary terms: *thermal modeling, external tool*

- **TimeSeriesField**

The type of Field that contains a 1-dimensional GridNodeBasedDomain whose DomainType is “Time” and a DiscreteFieldRange whose elements are themselves Fields sharing the same FieldDomain. A TimeSeriesField represents a time-dependent Field and its changes over time.

Related glossary terms: *Field, GridNodeBasedDomain, DiscreteFieldRange*

- ***title bar***

The horizontal strip at the top of a Session window that is vividly colored when Comet is the active application. The title bar contains the Comet icon (a command button that provides access to the system menu), the path to the file storing the Project that is open in this Session (unless the Project has not yet been saved, in which case, the title bar displays “Initializing Job – 1”), the current Stage, and Windows control buttons.

Related glossary terms: *GUI, Session window, system menu, Project, current Stage, Windows control buttons*

- ***toolbar***

A strip-shaped dockable window that contains icons providing quick access to Comet features. While many Comet features are accessed through the menu bar and context menus, some features may be more conveniently accessed by clicking a toolbar button. The initial location of the toolbar is as a horizontal strip immediately below the menu bar.

If a non-toolbar dockable window is already in a docking location, toolbars can also be docked in that location; the toolbars will share a strip along the outer border of the Workspace.

Related glossary terms: *GUI, dockable window, Scene window, Viewport*

- ***top-level entity***

A Comet entity that is added to the root Stage of a Project during the Project’s construction and is a Property of the Project. Currently, the top-level entities are the PhysicalModel, the AbstractDomainSet, the EnvironmentSet, the ProcessSet and the UserDefinedFieldSet. In contrast, all other Comet entities are added to the Project after its initial construction, are descendants of the top-level entities, and are found by searching downward from a top-level entity.

Related glossary terms: *Comet entity, root Stage, Project, Property, PhysicalModel, AbstractDomainSet, EnvironmentSet, ProcessSet, UserDefinedFieldSet*

- ***ToPMan***

A tool that drives the execution of *Processes*, Comet’s representation of engineering analysis processes as dataflow networks whose nodes are Tasks. ToPMan is an acronym for *Total Engineering Process Manager*.

Related glossary terms: *Total Engineering Process Manager, Process, Task, update, InputPort, OutputPort, ProcessInputPort, ProcessOutputPort, DataPipe, execution queue, dataflow*

- ***transferred Property***

A Property that is attached to an AbstractDomain and dynamically transferred to the source entity for whatever period that it is bound to the AbstractDomain.

Related glossary terms: *Property, owned Property, inherited Property*

- ***TransferredEntitySet***

The type of Comet entity that, given a particular GeometricClosure within a particular AbstractDomain, contains the set of Comet entities that will be dynamically transferred to the members of the GeometricClosure for the duration of their membership. The TransferredEntitySet is a child of the GeometricClosure.

Related glossary terms: *Comet entity, abstract modeling, AbstractDomain, GeometricClosure, physicalDomainFilterType, BoundEntitySet, PhysicalDomain*

[Go to start of Glossary](#)

U

- ***UEMTM***

See *Universal Engineering ModelTM*.

- ***UEMTM instance***

The data associated with a Stage. It is an instance of the Universal Engineering Model.

Related glossary terms: *Universal Engineering Model, Stage, Project*

- ***unit conversion factor***

A dimensionless, real-valued multiplicative conversion factor, which is part of the definition of a unit within the UnitDictionary. It is the factor by which the product of powers of the SI base units with the same dimensionalities as the unit being defined needs to be multiplied to match the unit being defined. For example, the conversion factor for *mile* is 1609.344 because a mile equals $1609.344 * m^1$, the conversion factor for *hour* is 3600.0 because an hour equals $3600.0 * s^1$, and the conversion factor for *mile/hour²* is $1609.344 * 3600.0^{-2}$ or 0.000124 because a *mile/hour²* equals $0.000124 * m^1 * s^{-2}$.

Related glossary terms: *unit of measure, UnitDictionary, base quantity types, SI base unit, dimensionality, UnitSystem*

- ***unit full name***

A string giving the full name of a unit of measure defined in the UnitDictionary. It is not the string by which a unit is referenced, which is instead the unit symbol.

Related glossary terms: *unit of measure, UnitDictionary, base quantity types, unit symbol, UnitSystem*

- ***unit of measure***

A fundamental quantity of measurement. Physical quantities are measured by estimating the ratio of their magnitude to the relevant unit of measure. The measurement is expressed as the product of the ratio and the unit.

Related glossary terms: *SI unit system, data model, Property*

- ***unit symbol***

The case-sensitive string by which a unit is referenced, *e.g.*, “m” for meter and “N” for Newton.

Related glossary terms: *unit of measure, UnitDictionary, base quantity types, unit full name, UnitSystem*

- ***UnitDictionary***

A Comet session's registry of the currently defined units. Each entry in the UnitDictionary contains the case-sensitive string by which the unit is referenced ("N" for the Newton unit), which is called the unit symbol, a potentially longer string giving the unit's full name ("Newton"), the exponents for the seven base quantity types, and the conversion factor.

Related glossary terms: *base quantity types, unit symbol, unit full name, dimensionality, unit conversion factor, UnitSystem*

- ***UnitSystem***

A set of seven units, one for each of the seven base quantity types.

Related glossary terms: *unit of measure, base quantity types*

- ***Universal Engineering Model™ (UEM™)***

Comet's general CAE data model, which captures all of the data associated with a CAE design project—systems and subsystems, engineering analysis processes and tasks, geometry, meshes, properties, environments, and analysis results—and does so in a largely CAD/CAE code-independent manner.

The Universal Engineering Model specifies the types of Comet data entities, the relationship rules for connecting Comet data entities into graphs, and the semantics for interpreting such a graph as the state of a CAE design project.

Related glossary terms: *Comet, CAE, data model, Stage, Project*

- ***update***

Given a Task, the action of that Task performing a computation in which it made use of data received on its InputPorts and it placed output data on its OutputPorts. ToPMan's scheduler places a Task on the *execution queue* as soon as all of the Task's InputPorts, if any, have received their inputs. When the Task reaches the front of the execution queue, it is updated. At the start of a Task update, the input datum within each of the Task's InputPorts is inserted into the Comet model at a location specified relative to the Task itself, following which the InputPorts are cleared. The Task then performs the computation specific to that Task's type and configuration. At the end of the Task's update, for each of the Task's OutputPorts, a datum is extracted from the Comet model at a location specified relative to the Task and copies of the datum are pushed across any DataPipes connected to the corresponding OutputPort and placed on the DataPipes' receiving Ports.

Related glossary terms: *Task, ToPMan, execution queue*

- ***UserDefinedFieldSet***

A top-level Comet entity, which contains the Fields that have been imported into a UEM instance and were defined by users, as opposed to the ones generated by external tool runs in response to OutputRequests. Every Project has exactly one UserDefinedFieldSet.

Related glossary terms: *Comet entity, UEM instance, top-level entity, Field*

- ***UserDefinedTask***

A type of Task that, when updated, evaluates a user-defined CML procedure. It is a type of Comet entity found within a Process.

Related glossary terms: ***CML, Task, Process, Comet entity***

[Go to start of Glossary](#)

V

- ***Value***

A type of Comet entity used to represent a single value. The available types of Value are StringValue, Scalar, FirstOrderTensor, CmtBaseEntityPointerValue, Expression, ColorValue, and CmtPosition. A Parameter associates a name with a Value. DynamicEntities group Values, where a SingleValuedDynamicEntity contains one Value.

Related glossary terms: ***Comet entity, StringValue, Scalar, FirstOrderTensor, CmtBaseEntityPointerValue, Expression, ColorValue, CmtPosition, Parameter, SingleValuedDynamicEntity***

- ***Vertex***

The type of BasicTopologyEntity that represents 0-dimensional geometric elements. Every CoordinateSystem contains a Vertex that represents the translational component of the CoordinateSystem's transform.

Related glossary terms: ***BasicTopologyEntity, CoordinateSystem***

- ***view***

A characterization of the virtual camera that will be “snapping a picture” of the set of 3-dimensional graphical objects representing the drawable objects in a given Scene. This characterization includes the camera's location, the direction in which it is aiming, and its angle of view. (In computer graphics lingo, the virtual camera is the *viewing frustum* and it is converted to an equivalent 4x4 *view transformation matrix*.)

Related glossary terms: ***GUI, drawable object, Scene, Scene window, Viewport***

- ***View manipulation toolbar***

The toolbar, typically located at the left of a Scene window, whose buttons enable users to alter the view of the drawable objects in the Scene/Viewport.

Related glossary terms: ***GUI, Scene, Scene window, view, Viewport***

- ***Viewport***

A particular view of the drawable objects contained in the given Scene. A Scene window is the window in which a Viewport is rendered.

Related glossary terms: ***GUI, drawable object, Scene, Scene window, view***

- ***Virtual Product Engineering (VPE)***

A strategy to achieve better product design and greater innovation, and to do so faster at lower cost, through the use of CAE tools and methods to simulate your product performance early, often and systematically in your product development efforts. An effective approach to VPE utilizes Simulation-Driven Engineering capabilities.

Related glossary terms: *Computer Aided Engineering, Computer Aided Design, Simulation Driven Engineering, Comet*

- **Visualization Toolkit** (also *The Visualization Toolkit* and *VTK*)

An open source, freely available software system for 3D computer graphics, image processing, and visualization. Comet currently employs VTK for 3-D Field visualization through its inclusion within ParaView; in the future, we plan to use it directly within visualization tools internal to Comet. VTK consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. Professional support and products for VTK are provided by Kitware, Inc. VTK supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as implicit modeling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. In addition, dozens of imaging algorithms have been directly integrated to allow the user to mix 2D imaging / 3D graphics algorithms and data. See <http://www.vtk.org> and <http://www.kitware.com>.

Related glossary terms: *visualization, Field, component software, ParaView*

- **VPE**

See *Virtual Product Engineering*.

- **VTK**

See *Visualization Toolkit*.

[Go to start of Glossary](#)

W

- **Windows control buttons**

The buttons located at the right of the title bar in a Windows window (*e.g.*, a Session or Scene window) that permit you to size and close the window.

Related glossary terms: *GUI, system menu, title bar, Session window, Scene window*

- **Workspace**

The configurable portion of a GUI Session window. This excludes the regions of the Session window containing fixed elements, such as the border, the title bar, the menu bar and the status bar. It includes the remaining area within which movable elements, such as project views, Process Schematics and toolbars, can be docked and dragged.

Related glossary terms: *GUI, Session window*

[Go to start of Glossary](#)

X

[Go to start of Glossary](#)

Y

[Go to start of Glossary](#)

Z

[Go to start of Glossary](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [X](#) [Y](#) [Z](#)