

Navisworks COM API Objects

Objectives

At the end of this module, you will be able to:

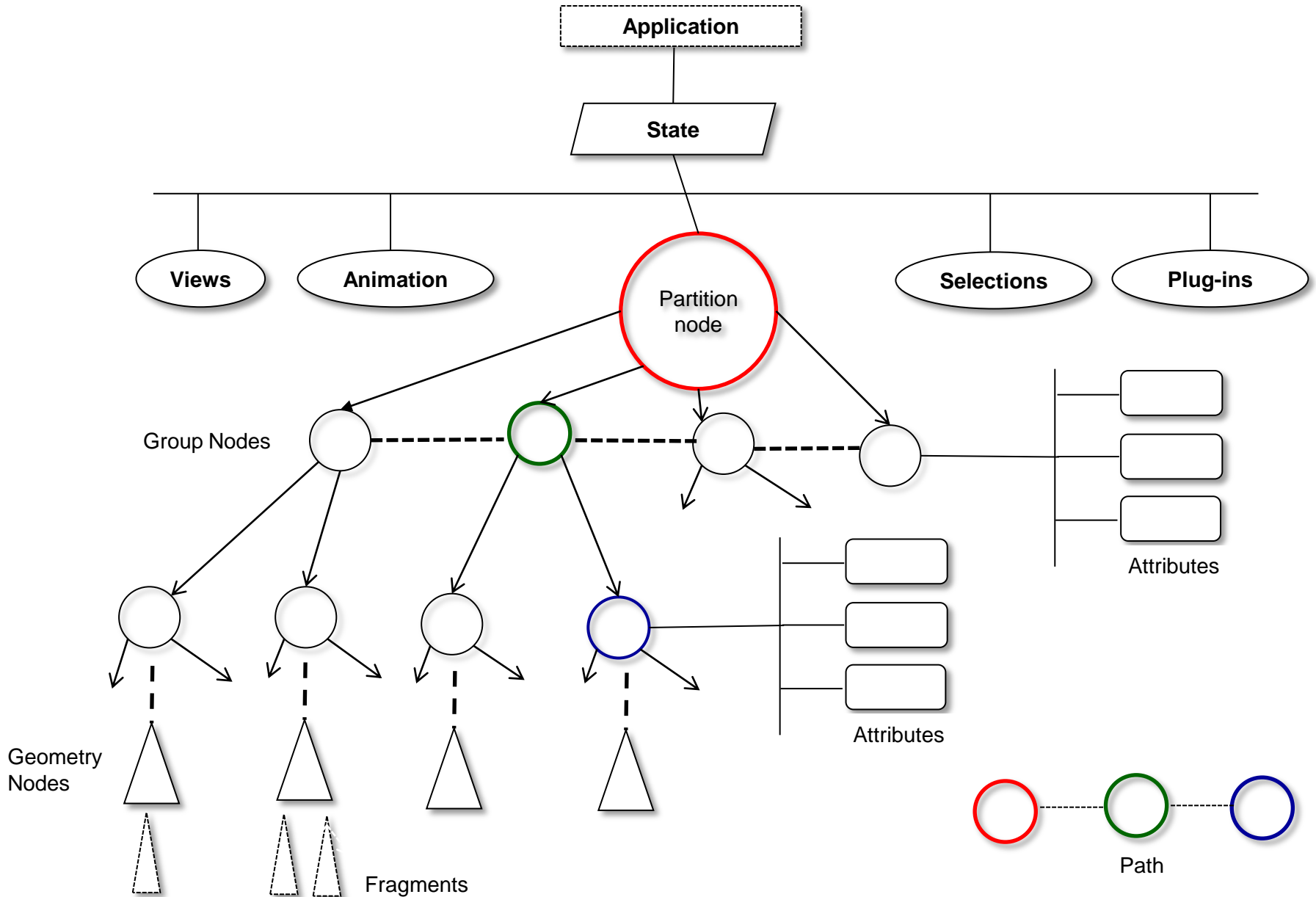
- Describe the core classes in the Navisworks COM API
- Describe the structure of the Navisworks COM API
- Identify the .NET assemblies required for working with the Navisworks COM API

COM Interfaces and Objects

Basics of the Navisworks COM API

- COM API is made up of interfaces and their implementation objects.
- Each object has one interface containing all methods.
- Object and corresponding interface have similar names, for example, `nwOpState` object implements `InwOpState` interface.
- Object properties and methods are accessed via the interfaces.
- Collection objects indexes start from 1.

COM API Data Model



Application Level Objects

DocumentClass

- `DocumentClass` class implements the `Inavisdock` interface and represents the Navisworks automation application.
- It is the entry point into the automation API.
- It has methods for opening, saving and appending files, defining application behaviour and for accessing the Navisworks internal state.

```
DocumentClass doc = new DocumentClass(); //Start Navisworks
```

The Plugin API does not expose an application object directly but it can be accessed in a plug-in class implementation.

Application Level Objects (contd.)

InwControl

- **InwControl** represents the Navisworks ActiveX control interface.
- Serves as the entry point into the ActiveX control API.
- It has methods for opening, saving and appending files, and defining application behaviour.

State Object

The state object ([InwOpState](#)) represents Navisworks internal state and provides access to the Navisworks model.

It corresponds to a document in Windows terminology, or an opened drawing in AutoCAD.

It is a container for the root partition, views, animation, selection sets, and plug-ins.

It has methods for view control and for creating other API objects.

It has associated events, for example, [OnCurrentViewChanged](#), [OnSelectionChanged](#)

Accessing the State Object

Access `InwOpState`:

- through the `State` method of the `Document` object (Automation)
- as a property of the `InwControl` object (ActiveX control)
- as a call-back function parameter (Plug-in)

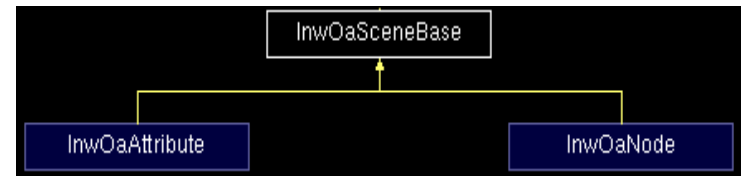
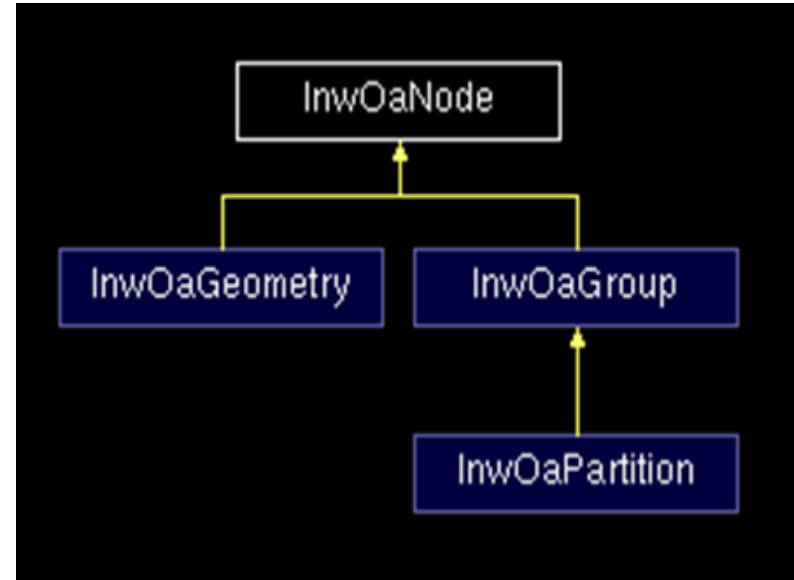
Node Object

The node object ([InwOaNode](#)) is the basic element of the data model. The model is made of nodes.

There are three types of nodes:

- Geometry ([InwOaGeometry](#)) – a leaf object node representing geometry
- Group ([InwOaGroup](#)) – a node containing child nodes
- Partition ([InwOaPartition](#)) – a group node representing root of model, or a source file

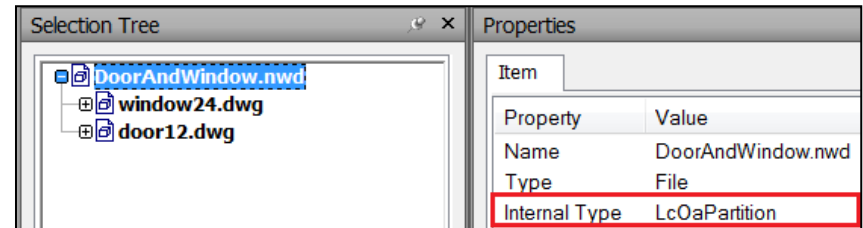
A node is associated with attributes.



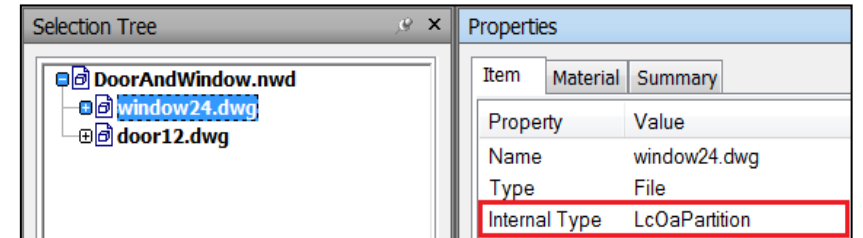
Partition

A partition ([InwOaPartition](#)) is a group node representing:

- the root of a model, or
- the root of each file in a model, or
- the root of an externally referenced file



A partition may contain other partitions.



The model root node is accessed through the [CurrentPartition](#) property of [InwOpState](#).

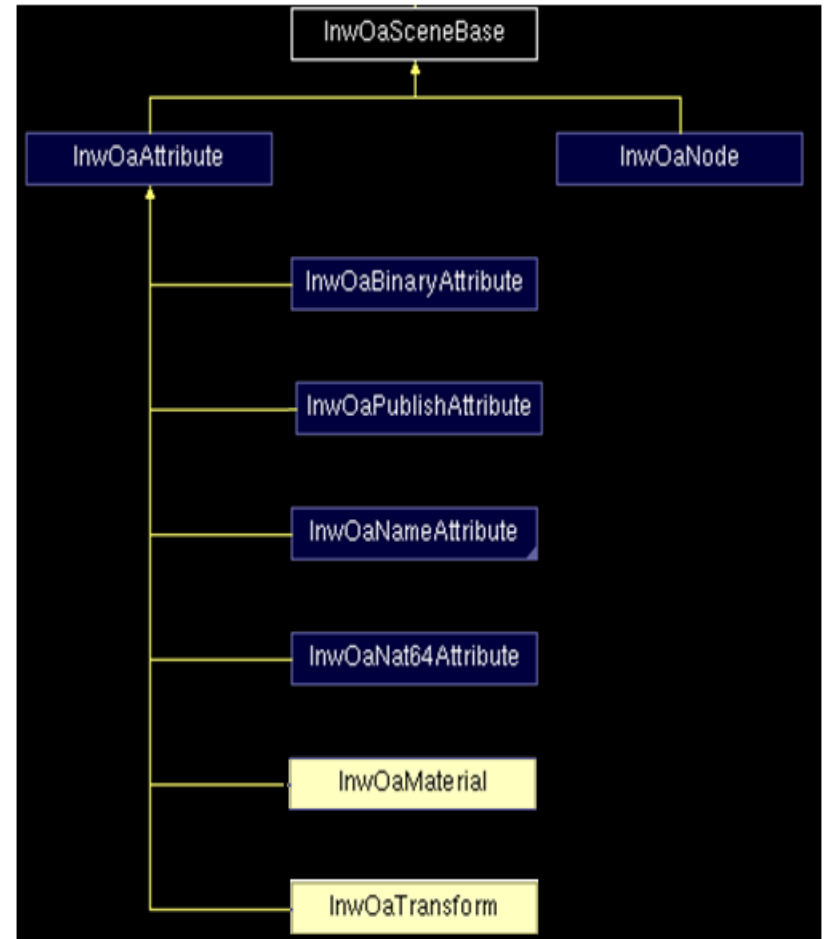
```
Document doc = new Document(); //Start Navisworks as an automation application
InwOpState state = doc.State(); //Get the state object
InwOaPartition rootPartition = state.CurrentPartition; //Get the model root partition
```

Node Attributes

An attribute ([InwOaAttribute](#)) associates information with a node, or controls node's appearance

There are two groups of attributes:

- Information attributes – [InwOaPropertyAttribute](#), [InwOaBinaryAttribute](#), [InwOaTextAttribute](#)
- Appearance attributes – [InwOaMaterial](#), [InwOaTransform](#)



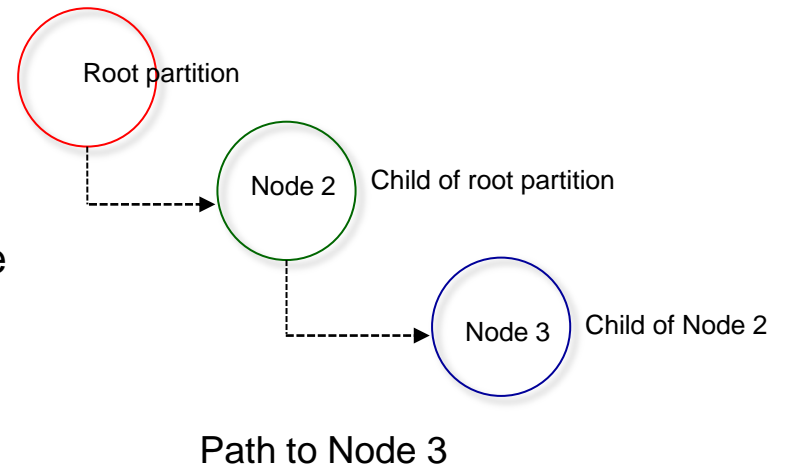
Path Object and Selection

A path ([InwOaPath](#)) is a parent-child sequence of nodes from the root partition to a particular node.

Path object is used in selection.

Select a node by adding its path to the paths within the current selection.

A selected node is the last node in a path.



Fragment Object

A fragment ([InwOaFragment](#)) is the representation of a geometry node or part of a geometry node.

Fragment provides the means of accessing geometry information in a model.

The fragments in a node, path or partition object is obtained using its [Fragments](#) method

Get the geometry associated with a fragment using [Fragment.GenerateSimplePrimitives\(\)](#).

Geometric Utility Objects

The API has several utility objects used in geometric operations, such as:

- [InwLPos3f](#) - 3D position representation
- [InwLVec3f](#) - 3D vector representation
- [InwLUnitVec3f](#) - 3D unit vector representation
- [InwLBox3f](#) - 3D box representation, consisting of 2 [InwLPos3f](#) corner objects
- [InwLPlane3f](#) - 3D plane representation
- [InwLRotation3f](#) - 3D rotation
- [InwLTransform3f3](#) - 3D transform

Creating COM Objects

New objects are created using the `ObjectFactory` method of `InwOpState` specifying the object type as a parameter

The object type is defined by `nwEObjectType`

```
Document doc = new Document(); //Start Navisworks as an automation application
InwOpState state = doc.State(); //Get the state object

//Create a camera object
InwNvCamera camera = state.ObjectFactory(eObjectType_nwNvCamera);

//Create a named view
InwOpView opView = state.ObjectFactory(eObjectType_nwOpView);
```

Navisworks COM Interop Assemblies

Use the interop assemblies, located in the Navisworks installation directory, for accessing Navisworks COM API functionality from .NET applications:

- [Autodesk.Navisworks.Api.Interop.ComApi.dll](#) - contains all the core interfaces
- [Autodesk.Navisworks.Interop.ComApiAutomation.dll](#) - provides access to the COM automation application

Summary

The Navisworks COM API provides a broad coverage of Navisworks functionality, and can be programmed in .NET

The API data model is a scene graph consisting of nodes.

Nodes have attributes that provide information about them or define their appearance.

A group node contains other nodes.

A partition node is a group node that represents the root of the model or of a file in the model.

A geometry node is a leaf node that represents geometry.

Navisworks COM API is used for creating automation, plug-in and ActiveX control applications.

.NET interop assemblies provide access to COM API functionality from .NET applications.

Further Information

Reference documentation is installed in <Navisworks>\api\com\documentation:

- Autodesk Navisworks COM Interface User Guide (COM Interface.pdf)
- Navisworks COM API Reference (NavisworksCOM.chm)

Code samples are installed in <Navisworks>\api\com\examples