Navisworks API Training

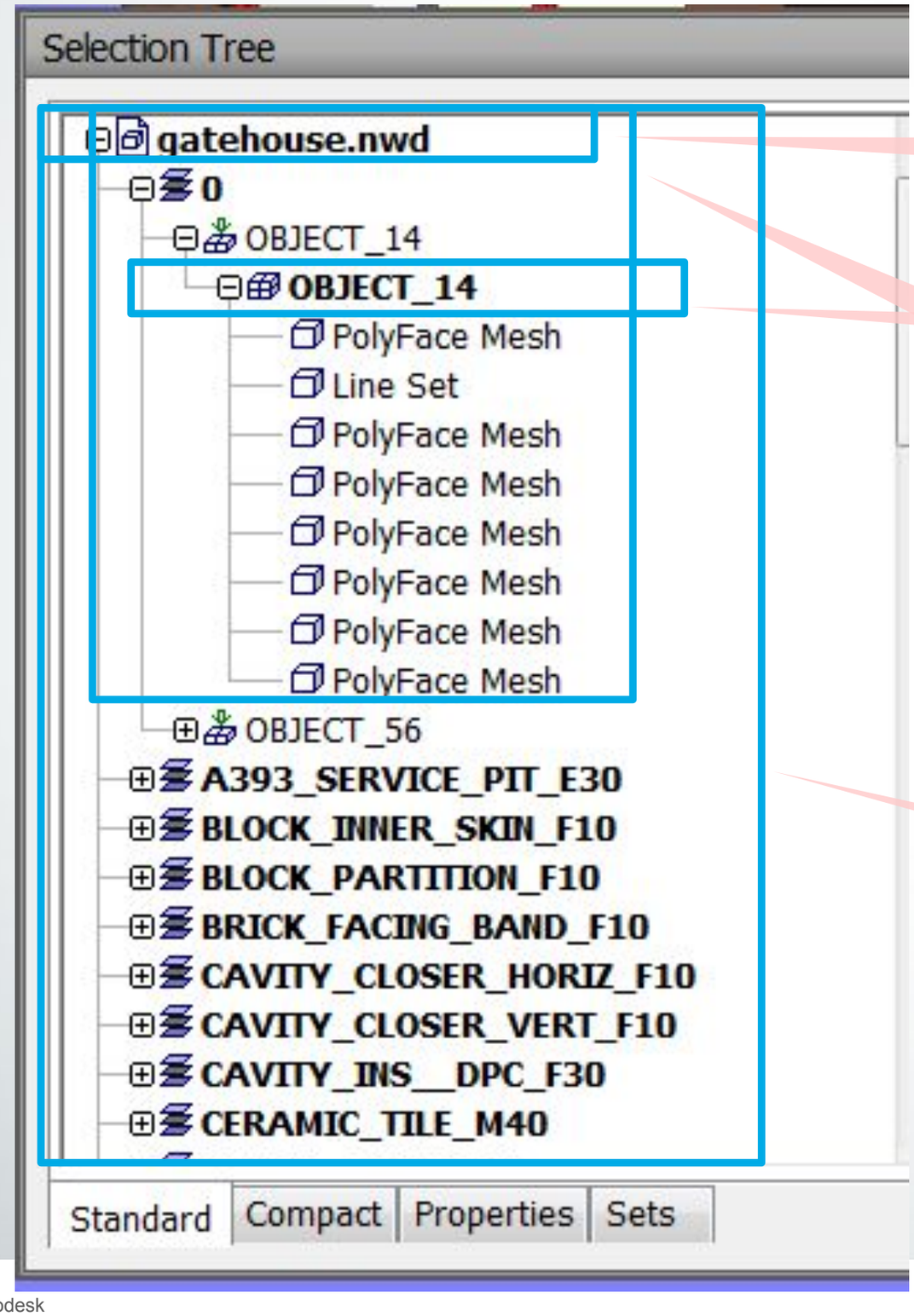
Lab 4 – Model

AUTODESK.

Agenda

- Model and ModelItem
- Attributes of Model Item Geometry

Model and ModelItem



Root Item

Model Item

Model Item Hierarchy:

- **Self**
- **Children**
- **Decedents**
- **Ancestors**

Model

AUTODESK.

ModelItem

- Represent a node in selection tree of UI
- same information like UI are available
  - IsHidden, IsCollection, IsComposite, DisplayName
- Store the properties
  - PropertyCategories (see Lab [Properties])
- Hierarchy
  - Ancestors: All ancestors of this item (excluding item itself) within the model hierarchy
  - AncestorsAndSelf : All ancestors of this item (including item itself) within the model hierarchy
  - Children: Children of this item within the model hierarchy (first level of child)
  - Descendants: All descendants of this item (excluding item itself) within the model hierarchy
  - DescendantsAndSelf :All descendants of this item (including item itself) within the model hierarchy

Querying Model

- Get specific model in the document

```
Model model = doc.Models[0];
```

- Get root Item of a model

```
ModelItem root = model.RootItem;
```

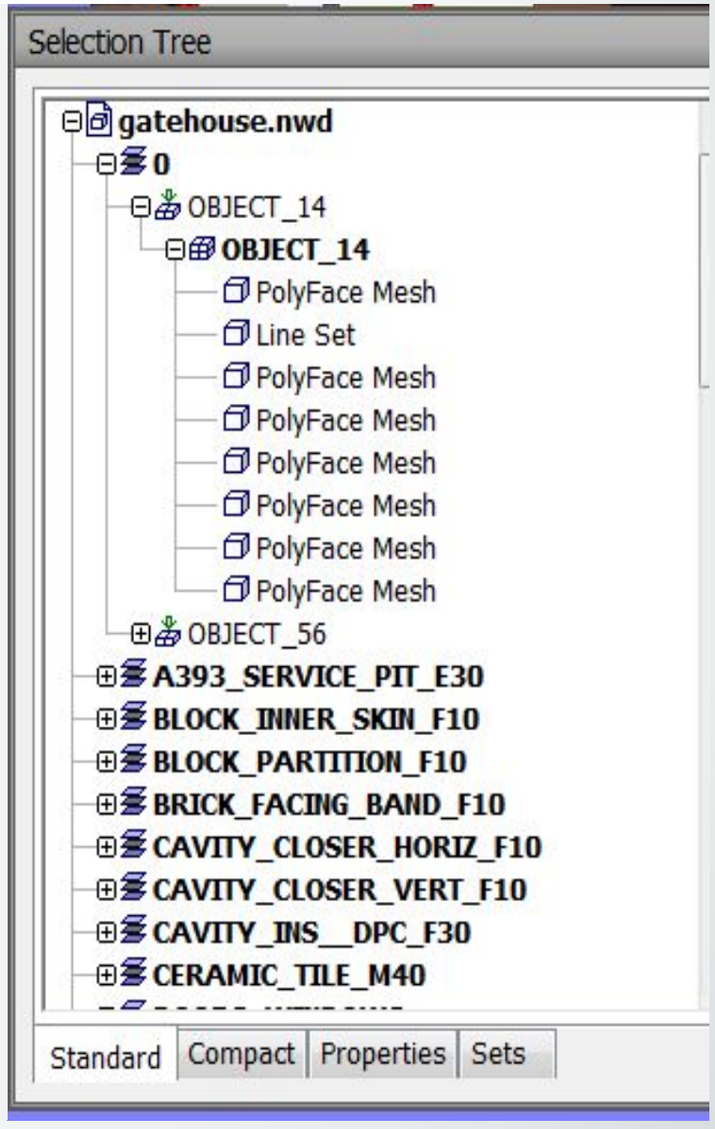
- Query property of model item

```
bool is_hidden = root.IsHidden;
```

Demo: Hierarchy of Model Tree

```
private void recursTree()
{
   Document oDoc = Autodesk.Navisworks.Api.Application.ActiveDocument;
   ModelItem rootItem = oDoc.Models[0].RootItem;
   //rootItem.DisplayName_
   //recurs from root item
   recursModel(rootItem);
}_

// recurs function
private void recursModel(ModelItem oParentModItem)
{
   foreach (ModelItem oSubModItem in oParentModItem.Children)
   {
      //Dump information of this item such as :
      //oSubModItem.DisplayName
      //recurs the children of this item
      recursModel(oSubModItem);
   }
}
```



Current Selection

- Document.CurrentSelection
- Store collection of model items that are selected
  - ModelItemCollection

```
//Get current selection
ModelItemCollection oModelColl= doc.CurrentSelection.SelectedItems;
```

```
//create a collection add the 9th item of the tree to the collection
 ModelItemCollection oNewItems = new ModelItemCollection();
 oItems.Add(doc.Models.First.RootItem.Children.ElementAt<ModelItem>(9));
```

```
//iterate over the selected Items
foreach (ModelItem item in Autodesk.Navisworks.Api.Application.ActiveDocument.
        CurrentSelection.SelectedItems)
 {
   //Add the children of the selected item to a new collection
    newCollection.AddRange(item.Children);
 }
```

```
//iterate over the selected Items
foreach (ModelItem item in Autodesk.Navisworks.Api.Application.ActiveDocument.
        CurrentSelection.SelectedItems)
 {
   //Add the Descendants of the selected item to a new collection
    newCollection.AddRange(item.Descendants);
 }
```

ModelGeometry

- Represents the geometry node in the Hierarchy
- No primitives information
  - Need use COM API (see Lab [COM Interop])

- **ModelItem.Geometry**
  - Geometry for this item, null if it does not have geometry
  - ModelItem.HasGeometry

Bounding Box

- **BoundingBox3D**
- Extents which aligns 3D Axis
- Identifies a cuboid-shaped bounded area in 3D space
- **ModelGeometry.BoundingBox**
  - extents of a model item
- **ModelItemCollection.BoundingBox**
  - bounding box of all items contained in the collection

```
// check if the bounding box of two items are intersected
ModelItemCollection oSelItems = doc.CurrentSelection.SelectedItems;
ModelItem oItem1 = oSelItems.ElementAt<ModelItem>(0);
ModelItem oItem2 = oSelItems.ElementAt<ModelItem>(1);
bool isIntersect = oItem1.Geometry.BoundingBox.Intersects(
            oItem2.Geometry.BoundingBox);
```

Where Clause

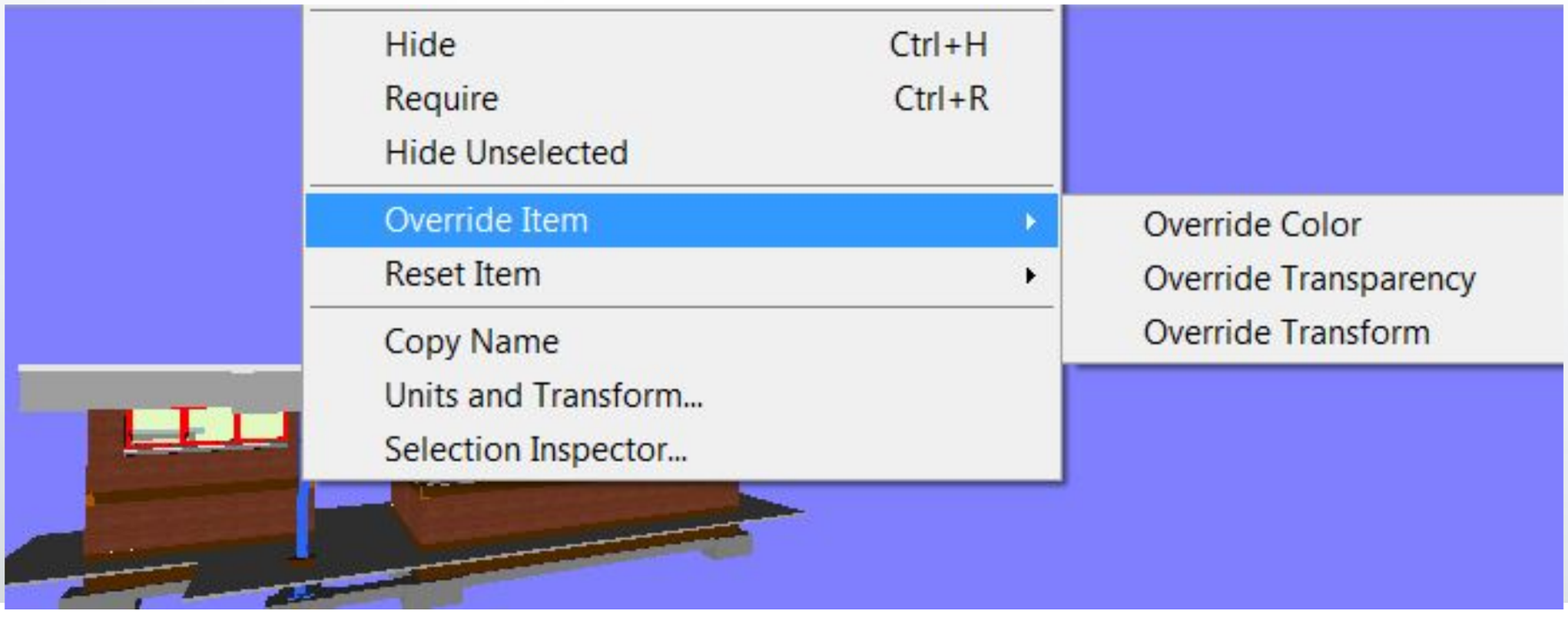
- Deeper search
  - Common search

```
//from root item, find those items that has geometry and is required.
IEnumerable<ModelItem> items =
        doc.Models.First.RootItem.Descendants.
         Where(x => x.HasGeometry && x.IsRequired);
```

- Search API : (see Lab [Search])

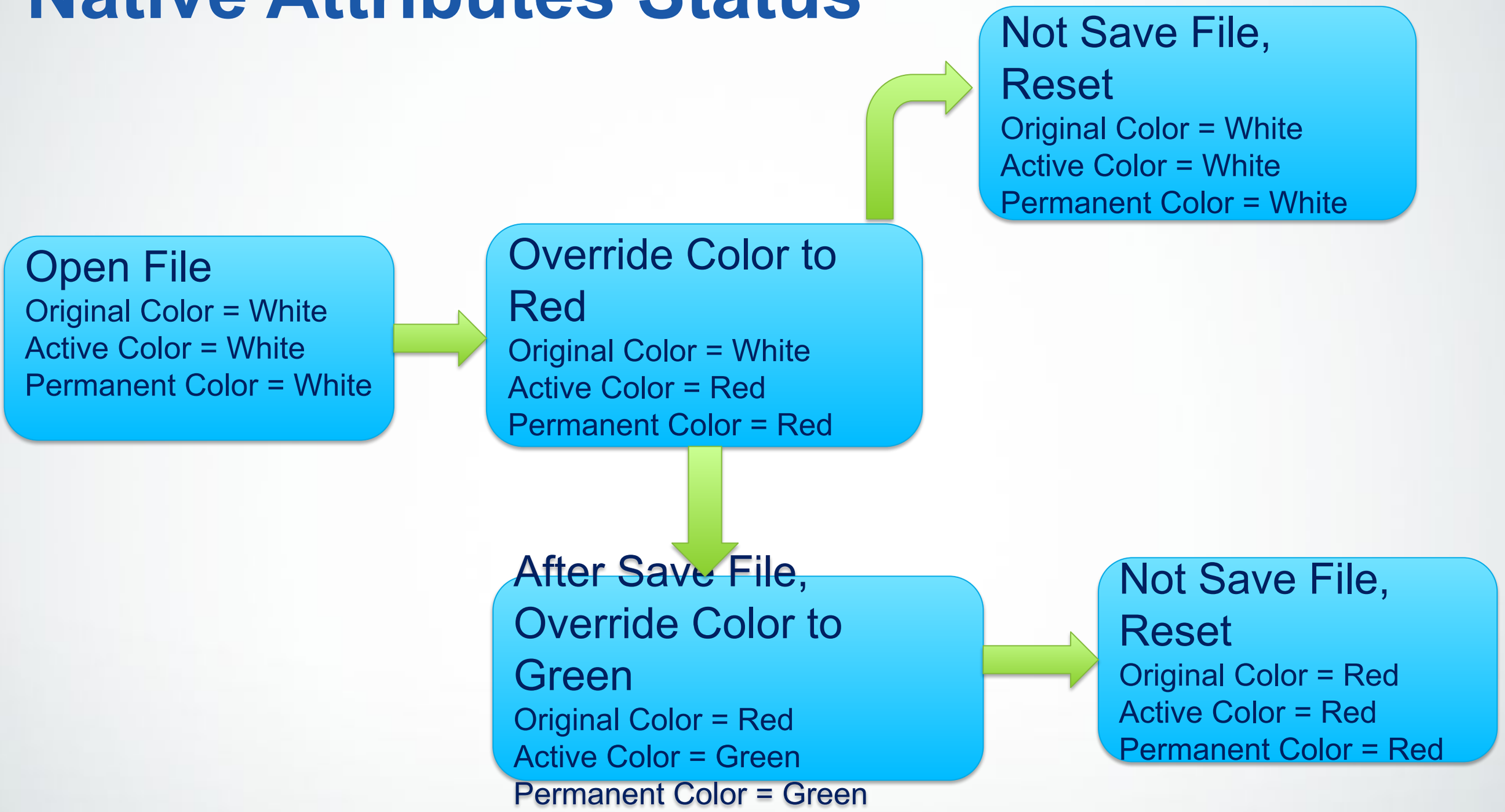
Attributes of Model Item Geometry

- Native attributes from original CAD file
  - Apply to model geometry node
  - Transforms: translation, rotation, and scale
  - Appearance: color and transparency
- Attributes in Navisworks scene view
  - Hidden, Required
- .NET API
  - Can access/modify appearance and transforms
  - Can access/modify hidden, required



AUTODESK.

Native Attributes Status



- When we say "override attributes", it is to override data of "permanent“.

Color

- **ModelGeometry.ActiveColor**
  - Current (visible) color for this geometry
- **ModelGeometry.OriginalColor**
  - Original color for this geometry (as specified by design file)
- **ModelGeometry.PermanentColor**
  - Permanent color for geometry. Either original color or color explicitly overridden by user
- **DocumentModels.OverridePermanentColor**
  - Override the permanent color of all ModelGeometry descendants of items.
- **Color** class
  - Represents a color as three floating-point components (r,g,b) with a normal range 0.0 to 1.0

```
//change the color of selected items to red
doc.Models.OverridePermanentColor(doc.CurrentSelection.SelectedItems,
Color.Red);
```

Transparency

- **ModelGeometry.ActiveTransparency**
  - Current (visible) transparency for this geometry
- **ModelGeometry.OriginalTransparency**
  - Original transparency for this geometry (as specified by design file)
- **ModelGeometry.PermanentTransparency**
  - Permanent transparency for geometry. Either original color or color explicitly overridden by user
- **DocumentModels.OverridePermanentTransparency**
  - Override the permanent transparency of all ModelGeometry descendants of items.
  - Range: 0.0-1.0

```
//change the transparency of the current selection to 0.5
doc.Models.OverridePermanentTransparency( doc.CurrentSelection.SelectedItems,0.5);
```

Transform

- **ModelGeometry.ActiveTransform**
  - Currently active transform of the geometry
- **ModelGeometry.OriginalTransform**
  - Original transform of the geometry when it was loaded
- **ModelGeometry.PermanentOverrideTransform**
  - Transform applied to the original transform of the model geometry
- **DocumentModels.OverridePermanentTransform**
  - Apply an incremental transformation of all ModelGeometry descendants
  - Transform3D: generic transform in 3D space. Provide static methods of translation, roation

```
Document doc = Autodesk.Navisworks.Api.Application.MainDocument;
// current selection
ModelItemCollection coll = doc.CurrentSelection.SelectedItems;
//build a vector for moving, along
Vector3D oNewVector3d = new Vector3D(1, 1, 0);
// orthogonal transforms + translation
//build an identity matrix which represents orthogonal transforms
Matrix3 oNewIndentityM = new Matrix3();

//create a transform from a matrix with a vector.
Transform3D oNewOverrideTrans = new Transform3D(oNewIndentityM, oNewVector3d);
//override the transformation of the selection
doc.Models.OverridePermanentTransform(coll, oNewOverrideTrans, true);
```

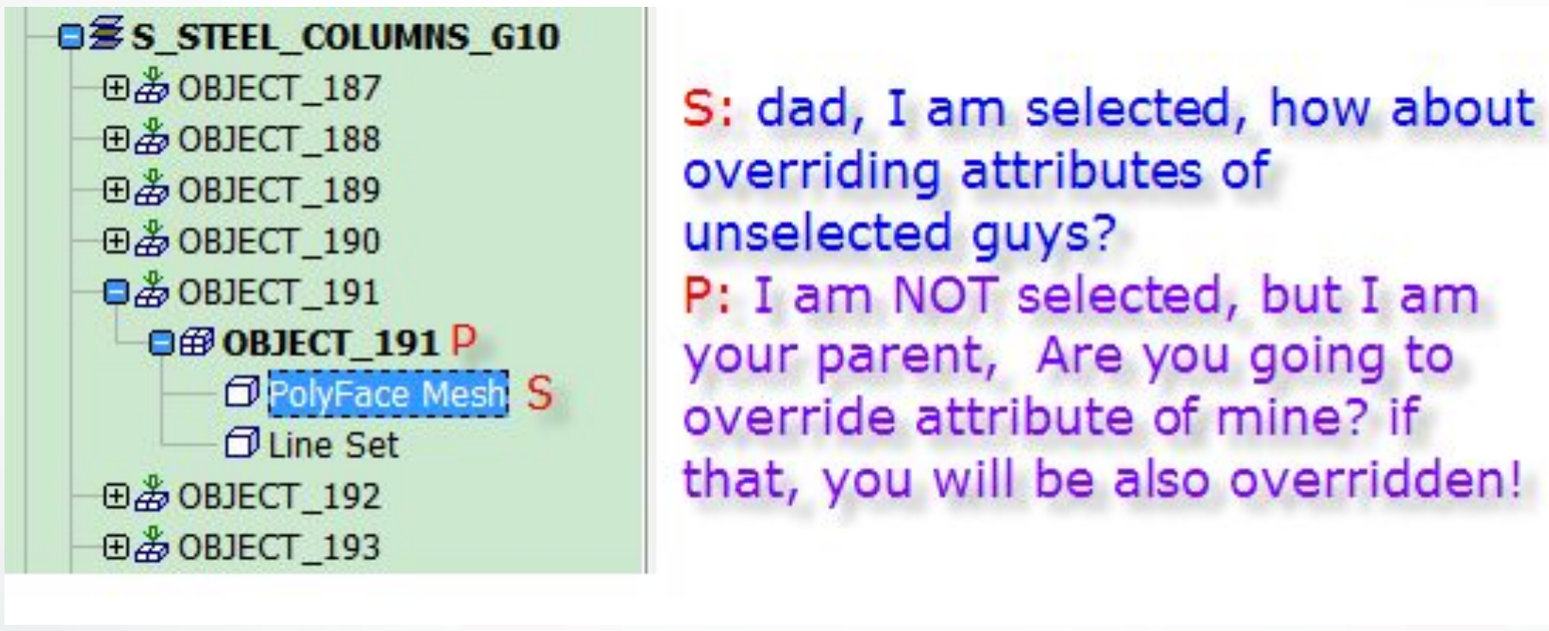
Hide Object

- **DocumentModels. SetHidden**
  - Set visible or invisible

```
//build a collection of model items
ModelItemCollection hidden = new ModelItemCollection();
//add current selected items and all of their descendants to the collection
hidden.AddRange(oDoc.CurrentSelection.SelectedItems);
//hide all of them
oDoc.Models.SetHidden(hidden, true);
```

Override Native Attributes of Unselected Items

- Challenge: parent and ancestors of selected item are not selected.
  - Cannot override attributes of parent and ancestors, otherwise, the item itself will be overridden.
  - Must filter out all those items



- Way1: Traverse and filter out item one by one. See this blog
- Way2: ModelItemCollection.Invert

Reset Attributes

- DocumentModels.ResetAllPermanentMaterials
  - reset the status all items since last saving (color and transparency)
- DocumentModels.ResetPermanentMaterials
  - reset the status specific items since last saving (color and transparency)
- DocumentModels. ResetAllPermanentTransforms
  - reset the status all items since last saving
- DocumentModelsResetPermanentTransform
  - reset the status specific items since last saving
- DocumentModels.ResetAllHidden
  - reset the status all items status of hidden

```
//reset appearance of selected items
doc.Models.ResetPermanentMaterials(doc.CurrentSelection.SelectedItems);
//reset transform of selected items
doc.Models.ResetPermanentTransform(doc.CurrentSelection.SelectedItems);
//reset hidden status of all items
doc.Models.ResetAllHidden();
```

Exercise

- Create a plugin
- Ask the user to select some items.
- Find other items whose boundingbox intersect with that of the selected items.
- Highlight other items as well

AUTODESK®