



Structure of Dgn File

Agenda

–Files:

- DgnFile

–Models:

- DgnModelRef
- DgnModel
- DgnAttachment

–Elements:

- ElementHandler
- ElementHandle
- EditElementHandle
- ElementRefBase

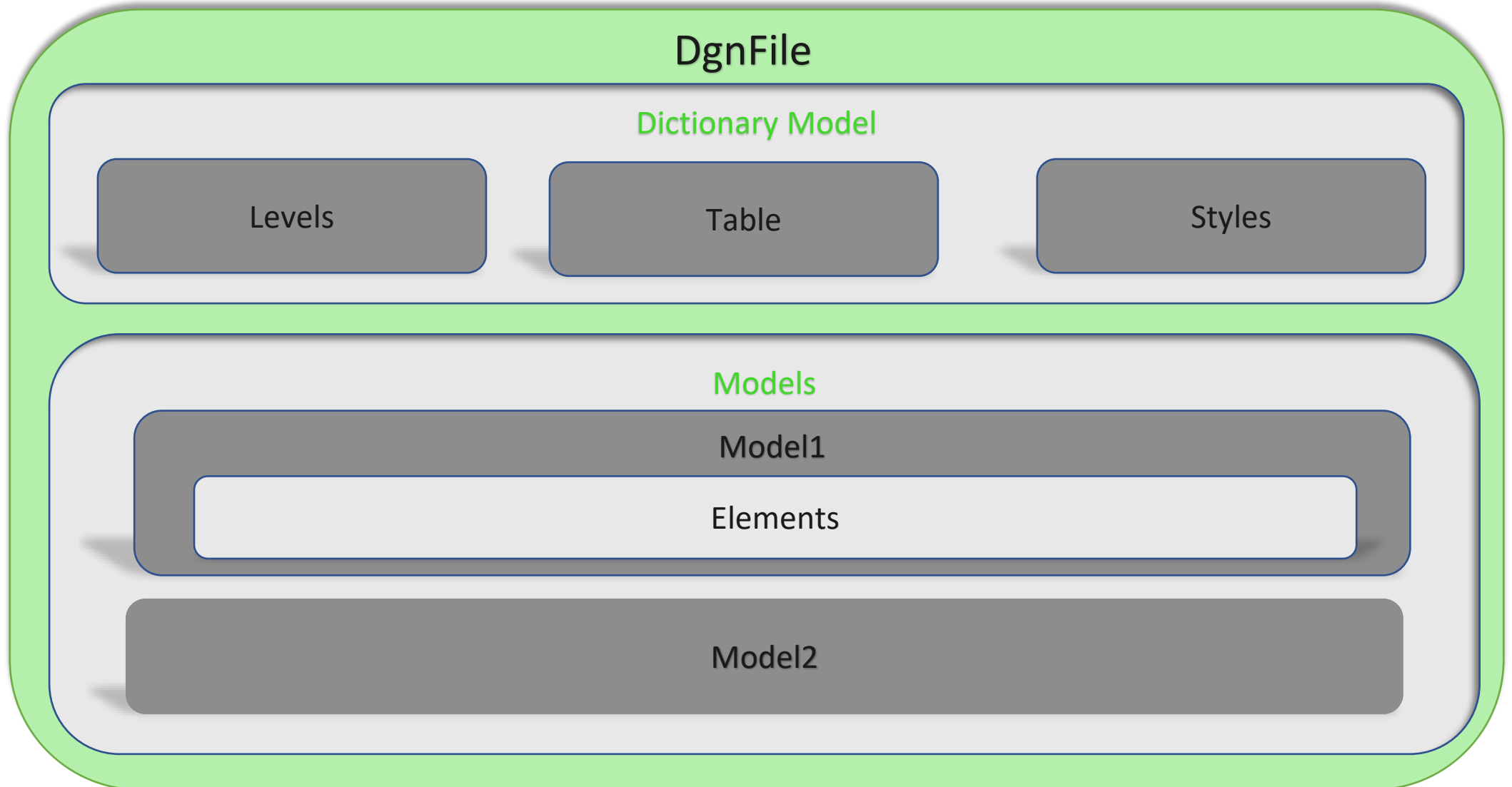
DgnFile

- Primary structure: DgnFile
- A DgnFile is an in-memory representation of a physical file, regardless of its format.
- A [DgnFile](#) is composed of [Design Models](#) which contain [Elements](#).
- A DgnFile can be incrementally saved

DgnFile

- Defined in MSPyDgnPlatform
- A DgnFile
 - Holds one or more DgnModels
 - Every DgnFile holds one Dictionary Model
 - Stores data common to all elements in File
- Elements in Design Models refer to levels, fonts, color definitions, line styles, etc. defined in their containing [DgnFile](#).

DgnFile Structure



DgnFile API's

- File Management API's
- DgnFile Properties API's
- Change Management API's
- Model Management API's
- Utility API's

DgnFile File Management

- **Accessing an Existing File**

- Create a new File Object: `DgnFile ()`
 - Creates a File Object that represents an existing physical file.
 - Sets up a connection to the physical file.
 - Does not load file data from disk.
- Load File Header data from disk: `DgnFile.LoadDgnFile`
 - Loads only File Header data & not any other data.
 - `IsLoaded ()` returns true after call to `LoadDgnFile ()`
 - `IsOpen ()` returns true after call to `LoadDgnFile ()` if opened in write mode.

Create New DgnFile

- You must start with an existing file, which serves as a "seed".
- To create a DGN file:
 - Call **Create** to create a DgnFile object.
 - Call **LoadDgnFile** to attempt to open the seed file.
 - Call **DoSaveAs** to save the file to a new name.

```
file = DgnFile (seedFileDocument, DgnFileOpenMode.eReadOnly)
if file.LoadDgnFile ()[0] != SUCCESS :
    return
if file.DoSaveAs (newFileDocument) != SUCCESS :
    return
```

DgnFile Properties

- GetFileName (), GetVersion ()
- IsOpen (), IsLoaded (), IsTransactable ()
- IsModel ()
- GetLastSaveTime ()
- GetDocument (), GetDocumentProperty ()
- Reference Counted
 - AddRef (), Release ()
- GetSignatureCount ()
- GetLevelCacheR ()

DgnFile Change Management

- Change Related
 - ProcessChanges ()
 - DoSaveAs, ()
 - DoSaveTo ()
 - HasPendingChanges ()
 - SetAbandonChangesFlag ()
 - ClearAbandonChangesFlag ()
 - SetFullSaveFlag ()

DgnFile: Accessing Models

- `GetModelIndex ()`
 - `ModelIndex` - Provides access to basic information about all the models in a `DgnFile`.
 - Accessing this information through the index is much faster than loading the model headers and using the `DgnModel` or `ModelIndex` APIs.
 - `ModelIndex modelIndex = dgnFile.GetModelIndex()`
 - `ModelIndexItem modelItem = modelIndex.GetItemByID (modelId, false);`
- `GetDictionaryModel()`
 - access data that applies to all models in a `DgnFile`.
 - Levels, linestyles, and other styles are defined in the dictionary model.
 - Special elements such as Shared cell definitions are also defined there.

DgnFile: Model Load

- Detect models in File: GetLoadedModelsCollection ()
 - Get the collection that can be used to iterate all models currently loaded from this file.
- Loading a Model: LoadModelById
 - Given a ModelId, loads Model from disk. More like – establishes a connection to the model data
 - Does not yet fill the Model
- Fill a Model: FillSectionsInModel
 - Fills a specified section of a Model with Elements
- FillDictionaryModel

DgnFile: Model Query

- ModelId related methods:
 - GetDefaultModelId ()
 - FindLoadedModelById ()
 - FindModelIdByName()

DgnFile: Create and Delete Models

- Create a new Model: `CreateNewModel ()`
 - Related method: `GetUniqueModelName ()`
- Delete a Model: `DeleteModel ()`

DgnFile: Model Management

- AddRootModel ()
 - Tell DgnFile to hold a reference to DgnModel. DgnFile keeps the model filled.
- DropRootModel ()
 - Tell DgnFile to drop reference to DgnModel. May cause model to empty itself
- Load a Root Model: LoadRootModelById ()
 - Calls DgnFileBase.LoadModelById () followed by AddRootModel ()
 - Optionally fills the model & loads all its DgnAttachments
- Copy Model: CopyModelContents
 - Call CreateNewRootModel to create destination empty model
 - Copies model data from source model to a newly created empty model

DgnFile Utility Methods/Properties

- Application Data Management
 - AddAppData (), DropAppData (), FindAppData ()
- Access Views & Resources
 - Color Definitions in File: GetColorMapP ()
 - Fonts in File: GetDgnFontMapP ()
 - Named Views in File: GetNamedViews ()
 - ViewGroups in File: GetViewGroups ()
- IsSameFile ()
 - Check if two fully qualified file-names represent the same physical file.

Agenda

–Files:

- DgnFile

–**Models:**

- DgnModelRef**
- DgnModel**
- DgnAttachment**

–Elements:

- ElementHandler
- ElementHandle
- EditElementHandle
- ElementRefBase
- [DgnPlatformAPI.h](#)

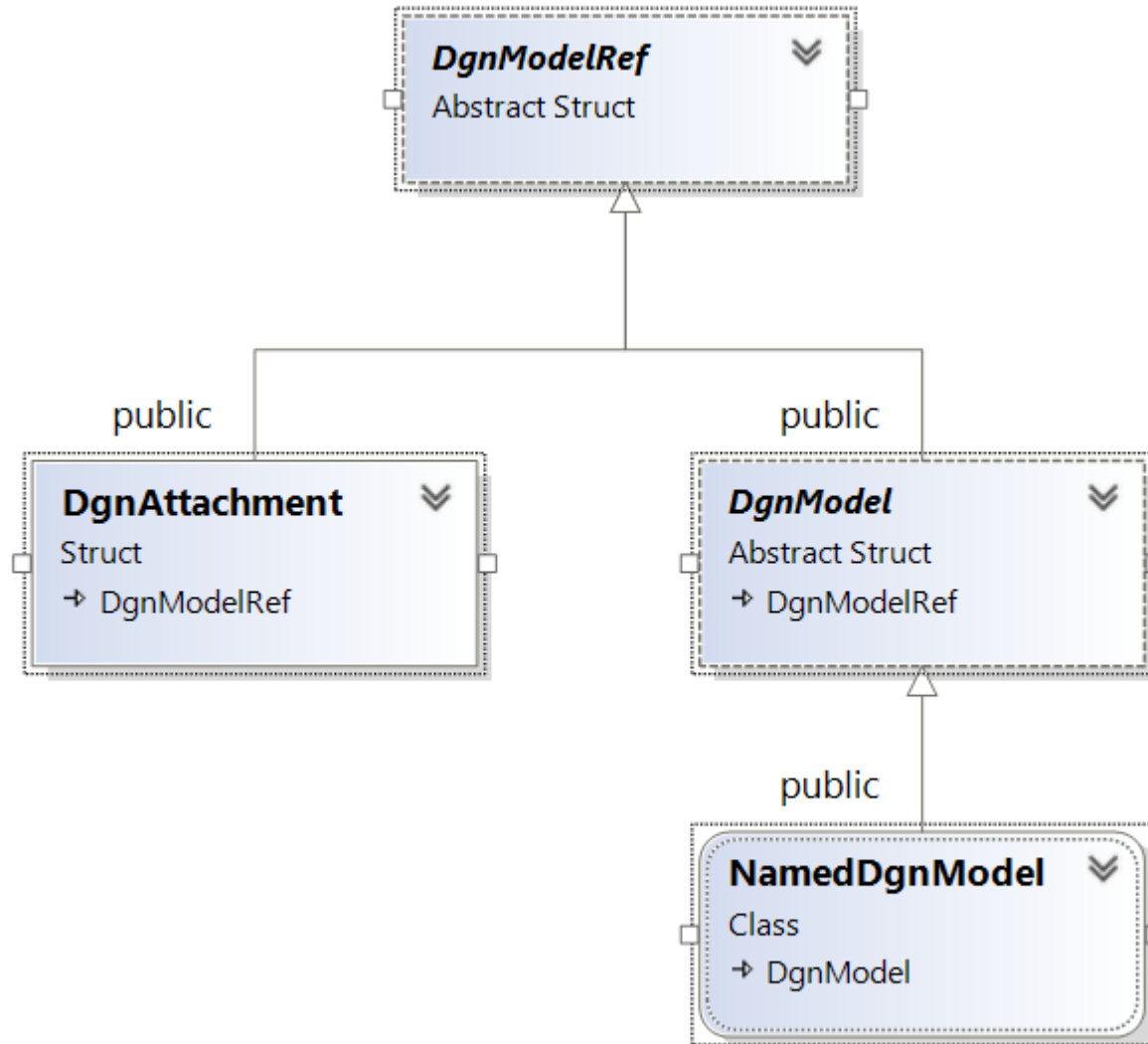
Models

- DgnFile Contains one or more Models
- One model may reference another. A DgnAttachment captures a reference between models.

Structures that represent Models

- **DgnModelRef:** provides access to a model that may be a direct reference to a root model or an indirect reference to an attachment model
 - Is base class of DgnModel & DgnAttachment.
- **DgnModel:** represents a (root) model in memory
- **DgnAttachment:** represents an attachment from one model to another

Class Diagram



DgnModelRef

- Defined in <[DgnModelRef.h](#)>
- A [DgnModelRef](#) provides access to a model in a [DgnFile](#).
- A [DgnModelRef](#) may be a
 - direct reference to a model (a [DgnModel](#))
 - an indirect reference to a model ([DgnAttachment](#)).
- This base class defines functions that are common to both models and attachments.

DgnModelRef

- Key Methods
 - GetDgnModel (), GetModelInfo ()
 - GetModelRefType ()
 - Root Model, Primary Attachment, Non-Model (Dictionary), Transient
 - Is3D ()
 - GetLevelCache (), GetFileLevelCache ()

DgnModel

- Defined in MSPyDgnPlatform
 - A [DgnModel](#) represents a [DgnModel](#) in memory.
 - A [DgnModel](#) is part of a [DgnFile](#).
 - A [DgnModel](#) belongs to one and only one [DgnFile](#).
-
- Sub-class of DgnModelRef
 - A DgnModel consists of:
 - One or more elements
 - One or more attachments

DgnModel: Elements in a Model

- A DgnModel consists of two ElementLists
 - Control ElementList – GetControlElementsP ()
 - Graphic ElementList – GetGraphicElementsP ()
 - GetElementsCollection () – Gets both Control & Graphic Elements
- Read Elements from physical File into memory: FillSections ()
- Empty Elements from this Model: Empty ()
- Check if a Model is Filled: IsFilled ()
- Get Element Count: GetElementCount ()
- Get an Element given its ElementId: FindElementById ()

ModelInfo

- Defined in MSPyDgnPlatform
- Stores the properties of a Model
 - Name
 - Type – DgnModelType {Normal, Sheet, Drawing}
 - Description
 - Properties & Settings
 - Properties: ModelPropertyFlags
 - modells3D, Hidden, IsMarkup
 - Settings: ModelSettingFlags
 - Units
- GetSolidExtent ()
- Sheet Definition
 - GetSheetDef ()
- GetDefaultRefLogical ()

DgnModel: Properties

- GetModelName ()
- ModelInfo
 - GetModelInfo ()
- GetDgnFileP ()
- IsReadOnly (), GetRange ()
- IsDictionaryModel ()
- Application Data Management
 - AddAppData (), DropAppData (), FindAppData ()

DgModels & DgnAttachments

- A DgnModel can have one or more attachments.
- A DgnModel can be used by one or more Attachments
 - DgnModel.GetReferencedBy ()
 - List of DgnAttachments that reference this model
- DgnModelRef structure provides access into DgnAttachments of a DgnModel
 - DgnModelRef.FindDgnAttachmentByElementID ()
 - DgnModelRef.GetDgnAttachments (), CreateDgnAttachment (), DeleteDgnAttachment ()
 - DgnModelRef.GetParentModelRef (), IsDgnAttachment (), IsNestedDgnAttachment ()

Reachable Attachments

- `DgnModelRef.GetReachableModelRefs ()`
 - Get a collection of all `DgnModelRefs` that includes “this” `DgnModelRef`, its attachments and nested attachments

```
DgnModelRef model = ...
```

```
ReachableModelRefCollection c = model.GetReachableModelRefs()
```

```
for modelRef in c:
```

```
    DgnFileP file = modelRef.GetDgnFileP()
```

```
    DgnModelP model = modelRef.GetDgnModelP()
```

```
    ... .
```

Reachable Elements

- `DgnModelRef.GetReachableElements ()`
 - Get a collection of all Elements reachable from “this” `DgnModelRef`
 - The reachable set includes all elements in this model as well as all elements in models attached to this model, including nested attachments.

```
DgnModelRefR model = ... ;  
ReachableElementCollection c = model.GetReachableElements ()  
    for elem in c:  
        elem...
```

DgnAttachment

- The action to attach a Model to another (parent) Model
 - represented by DgnAttachment
- A Model may be attached to itself – self attachment
- The same Model may be attached multiple times

DgnAttachment

- Defined in MSPyDgnPlatform
- Sub-class of DgnModelRef
- A [DgnAttachment](#) represents a reference from one model to another, with transforming, clipping, and filtering applied.
- An attachment is represented in memory by a [DgnAttachment](#) object. The data needed to construct the [DgnAttachment](#) object is persistent in the referencing model.

DgnAttachment: Name/Description

- Every Attachment can have a Logical Name
- If two attachments are sourced from the same model, then each attachment should have a unique Logical Name
 - One of the Logical Names can be NULL
 - Methods: GetLogicalName (), SetLogicalName (), SetUniqueLogical ()
- Each Attachment can have an optional Description
 - GetAttachDescription (), SetAttachDescription ()

DgnAttachment: Source File/Model

- Defines which Model is attached
 - GetAttachFileName (), GetAttachModelName ()
- Get absolute File Path when originally attached
 - GetAttachFullFileSpec ()
- Checks for when source file or model is deleted
 - IsMissingFile (), IsMissingModel ()

DgnAttachment: Parent Properties

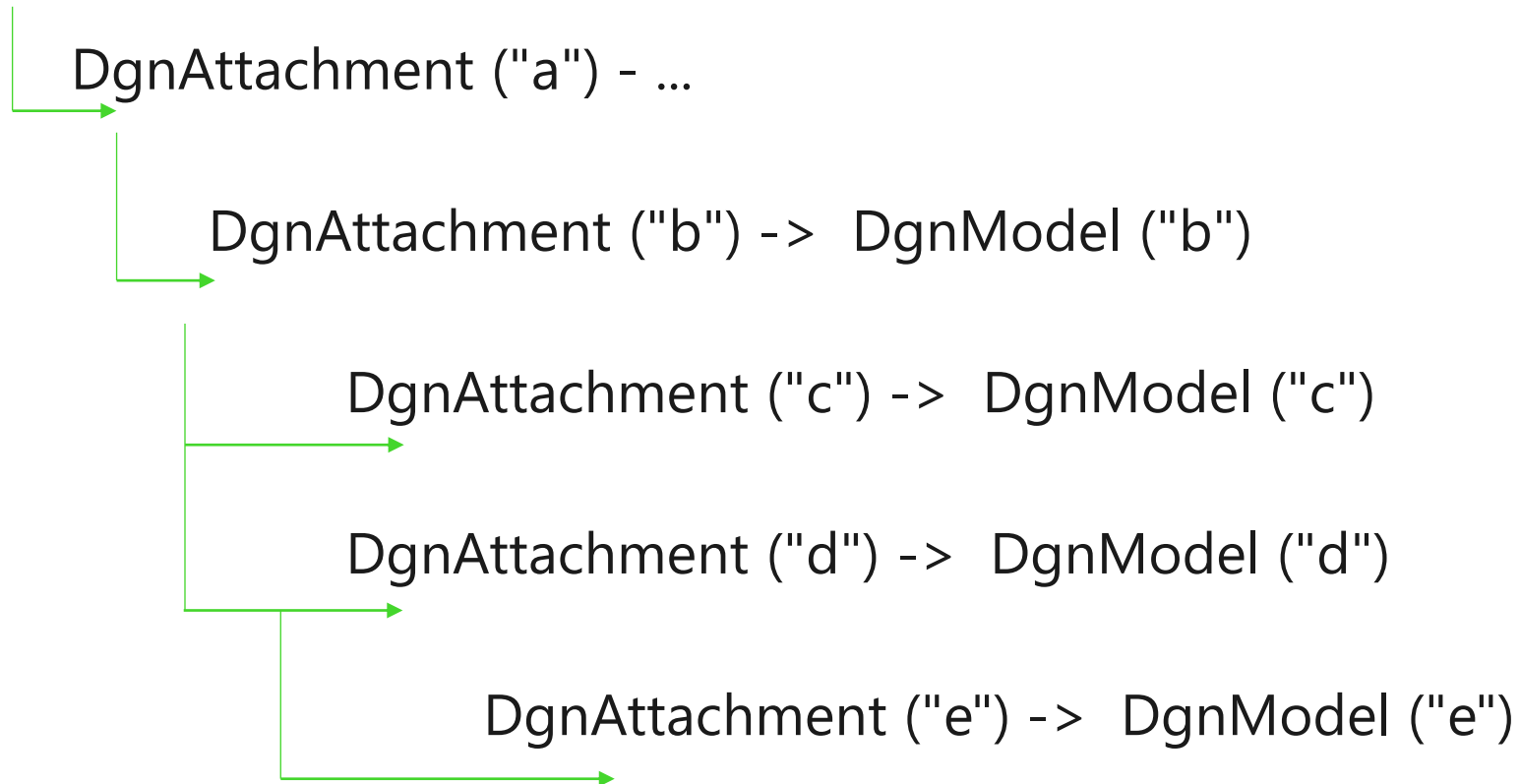
- For a Primary Attachment: `GetParent ()`, `GetParent ()`
- For a Nested Attachment:
 - Get Direct Parent: `GetParentDgnAttachment ()`, `GetParentDgnAttachment ()`
 - Get Base Primary Attachment: `GetBaseDgnAttachment ()`, `GetBaseDgnAttachment ()`

Nested Attachments

- A DgnModel attaches one or more Models
- Each attachment model can attach one or more Models
- Primary Attachment: Parent is a root DgnModelRef
- Nested Attachment: Parent is an attachment DgnModelRef
- Nesting can be to any level
 - A parent DgnModelRef may specify a cut-off limit to its nesting: `GetNestDepth ()`, `SetNestDepth ()`
 - An attachment may specify that it should be displayed as nested through its parent:
`GetDoNotDisplayAsNested ()`, `SetDoNotDisplayAsNested ()`
- The collection of all attachments to a root DgnModel & its Nested Attachments form a Directed Acyclic Graph
 - Circular attachments are cut-off

DgnAttachment Hierarchy

DgnModel ("a")



DgnAttachment: Properties

- Locate: IsLocateLocked (), SetLocateLock (), GetLocateStatus ()
- Snap: IsSnapLocked (), SetSnapLocked ()
- Display: IsDisplayed (), SetIsDisplayed ()
- Level: GetDgnAttachmentLevel (), SetDgnAttachmentLevel (), DoesLevelControlDisplay (), SetLevelControlsDisplay ()
- Is in selection set: IsInSS ()
- Is Highlighted: IsHighlighted ()
- Application Data Management
 - AddAppData (), DropAppData (), FindAppData ()

DgnAttachment: Transformation

- Origin Displacement
 - Origin point in DgnAttachment: GetRefOrigin (), SetRefOrigin ()
 - Origin point in parent DgnModel that origin of DgnAttachment should coincide with: GetMasterOrigin (), SetMasterOrigin ()
- Rotation
 - Rotation relative to parent: GetRotMatrix (), SetRotMatrix ()
- Scaling
 - Scale mode: GetScaleMode (), SetScaleMode ()
 - TrueScale (default), StorageUnits, Direct
- Transformation
 - Transformation relative to Parent: GetTransformToParent ()
 - Combination of origin displacement, rotation & scaling

DgnAttachment Scaling

- Display Scale
 - attachment display scale in the referenced model.
- Stored Scale
 - scale factor to be applied when displaying elements from the referenced model.
- If the scale mode is `SCALE_MODE_Direct`
 - the displayed scale is the same as the display scale
- otherwise
 - the displayed scale is calculated at load time based on the stored scale, the ScaleMode, and the units of the DgnAttachment model and the parent model.

DgnAttachment: Clipping

- If an Attachment is clipped then
 - Geometry of elements inside the clip boundary displayed.
 - Geometry of element outside the clip boundary not displayed.
- Check if Attachment is clipped: `IsClipped ()`
 - A nested attachment is clipped if any of its parents/ancestors are also clipped
- Clipping can applied by:
 - An element of parent: `GetClipElementId ()`, `SetClipElementId ()`
 - A set of points: `GetClipPoints ()`, `SetClipPoints ()`
- Clipping can be rotated:
 - `GetClipRotMatrix ()`, `SetClipRotMatrix ()`, `IsRotateClipping ()`
- In 3D, can set front & back clip planes:
 - `HasFrontClip ()`, `HasBackClip ()`
 - `GetFrontClipDepth ()`, `SetFrontClipDepth ()`
 - `GetBackClipDepth ()`, `SetBackClipDepth ()`

Agenda

–Files:

- DgnFile

–Models:

- DgnModelRef
- DgnModel
- DgnAttachment

–**Elements:**

- ElementHandler**
- ElementHandle**
- EditElementHandle**
- ElementRefBase**
- [DgnPlatformAPI.h](#)

Elements

- Key Concepts
 - ElementHandler
 - ElementHandle
 - EditElementHandle
 - ChildElement/ChildEditElement

ElementHandler

- Defines the behavior of elements that the handler controls
- There is one ElementHandler for each Element Type
- A Handler typically has no state information of its own
- A Handler method is expected to apply its logic to the element that it is asked to process
- All ElementHandler methods typically take an ElementHandle or an EditElementHandle reference as argument

Handler

- Defined in MSPyDgnPlatform
- Defines the standard queries and operations available on all elements
 - Graphical or Non-graphical
- Every element in MicroStation has a Handler.

Handler Types

- There are different Handler Types represented by
 - GetDisplayHandler () – Handler of a Displayable Element
 - GetTransactionHandler () - Handler of an Element that wants to participate in Transactions
 - GetDependencyHandler () – Handler of an Element that wants to know if other Elements referenced by its XAttributes have changed

Handler: Key Methods

- GetDescription (), GetTypeName ()
- QueryProperties (), EditProperties ()
- ApplyTransform ()
- ExposeChildren ()
- Handler Extension
 - Used to add additional interfaces to a Handler at runtime

ElementHandle

- Defined in MSPyDgnPlatform
- An indirect reference to an element's current state
- Provides "ReadOnly" access to underlying element
- Lightweight Object & usually stack based
- ElementHandle Constructors: Can constructed from
 - ElementRefBase/DgnModelRef pair
 - MSElementDescr. DgnModelRef can be optionally provided.
 - ElementId/DgnModelRef pair
 - MSElement/DgnModelRef pair
 - Another ElementHandle – copy Constructor

ElementHandle: Properties

- Element Properties
 - GetElementID ()
 - IsValid ()
 - IsPersistent ()
 - GetElementType ()

ElementHandle: Queries

- Element Queries
 - GetElementRef (), PeekElementDescr (), GetElementDescr (),
 - GetElementHeader ()
- Container Queries
 - GetDgnModel (), GetDgnFile ()
- Handler Queries
 - GetHandler (), GetDisplayHandler (), GetDependencyHandler (), GetTransactionHandler ()
- Text Queries
 - GetITextQuery (), GetITextEdit ()
- Linkage Queries:
 - BeginElementLinkages (), EndElementLinkages ()

EditElementHandle

- Defined in MSPyDgnPlatform
- Sub-class of ElementHandle
- Provides “ReadWrite” access to underlying element
- Typical Workflow:
 - Create an EditElementHandle using one of its constructors
 - Extract & modify underlying element using one or more modification methods
 - Commit change to Model
 - AddToModel (), DeleteFromModel (), ReplaceInModel ()

EditElementHandle: Extraction Methods

- Can be constructed in the same way as a ElementHandle
- Element Extraction Methods
 - Element Related
 - GetElement ()
 - Element Descr Related
 - GetElementDescr (), ExtractElementDescr ()

EditElementHandle: Modification Methods

- ModelRef/ElementRef Related
 - SetElementRef (), SetModelRef (), ReplaceElement ()
- XAttribute Related- Add, Delete XAttributes
 - ScheduleWriteXAttribute (), CancelWriteXAttribute (), ScheduleDeleteXAttribute (), CancelDeleteXAttribute ()
- Linkage Related Add, Remove Linkages
 - ReplaceElementLinkage (), RemoveElementLinkage (), InsertElementLinkage (), AppendElementLinkage ()

Writing XAttribute on Elements

```
EditElementHandle eh
eh.SetElementDescr(...)
XAttributeHandlerId myXaHandlerId (...)
eh.ScheduleWriteXAttribute(myXaHandlerId, xaIndex, m_data)
eh.AddToModel() //adds the element and XAttribute to the active model cache
```

ChildElemIter

- Defined in MSPyDgnPlatform
- An Iterator for children of an ElementHandle
- Sub-class of ElementHandle
- It takes ElementHandle and ExposeChildrenReason as argument
- Default Reason is Query
- Typical Loop

```
childIter = ChildElemIter(m_eeh)
while childIter.IsValid():
    elm = childIter.GetElement()
    childIter = childIter.ToNext ()
```

ChildEditElemIter

- Defined in [<ElementHandle.h>](#)
- An Iterator for children of an EditElementHandle
- Sub-class of EditElementHandle
- It takes EditElement and ExposeChildrenReason as argument
- Default Reason is Edit
- Typical Loop

```
childIter = ChildEditElemIter(m_eeh)
while childIter.IsValid():
    elm = childIter.GetElement()
    childIter = childIter.ToNext ()
```

Connect with Bentley

Connect with us

Bentley®
www.bentley.com

developer.bentley.com

communities.bentley.com/products/programming

Social Media



[Medium.com](https://medium.com)

MicroStation