OMF API: Power to Customize AutoCAD Architecture and MEP

Amod Kulkarni

Principal Engineer

amod.kulkarni@autodesk.com

Vinit Shukla

Sr. Software Engineer

vinit.shukla@autodesk.com

Class summary

What this class is about:

- Discuss fundamentals of ObjectARX and OMF API
- Explain differences between ObjectARX and OMF API
- Demonstrate live examples of OMF capabilities

What this class is not about:

- Discuss C++/C#/VB/.NET programming languages
- Discuss all public API available in ObjectARX or OMF
- Discuss or resolve specific issues related to any existing API

Prerequisite:

- Knowledge of ACA or MEP product
- Basic knowledge of programming and/or object oriented concepts

Class Agenda

- Fundamentals of ObjectARX API
  - Understand concepts of ARX/DBX/CRX
  - Introduction to concepts (DWG data structure, AcDbObject, AecDbEntity, reactors, draw functions etc.)
- OMF API as extension of ObjectARX
  - What is difference between ObjectARX and OMF
  - Introduction to OMF concepts
  - Create our first OMF application using ObjectARX wizard
  - Demo of more OMF samples

Key learning objectives

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

- Get familiar with overall concept of product customization using API
- Understand basics of OMF API
- Understand different customization possibilities in ACA and MEP
- Begin to think innovatively on how OMF power can be used to improve productivity.

Introduction to ObjectARX World

What is API and ObjectARX?

API

- Application Programming Interface

AutoCAD Runtime eXtension

- A DLL plug-in model
- A set of Object Oriented C++ libraries
- A framework

A framework?

- More than just a toolkit
- Specialized objects can be plugged back into the system
  - Custom entities
  - Reactors
  - ...

What can I do with ObjectARX?

Modify and extend the drawing database

- Create/modify/erase objects
- Create new types of objects

Modify AutoCAD’s user interface

- Commands
- Toolbars/dialogs
- Properties Window
- Design Center
- Display system

Monitor/Modify AutoCAD’s standard behavior

- Event notifications
- Input point acquisition

Development Environment

ObjectARX SDK 2017

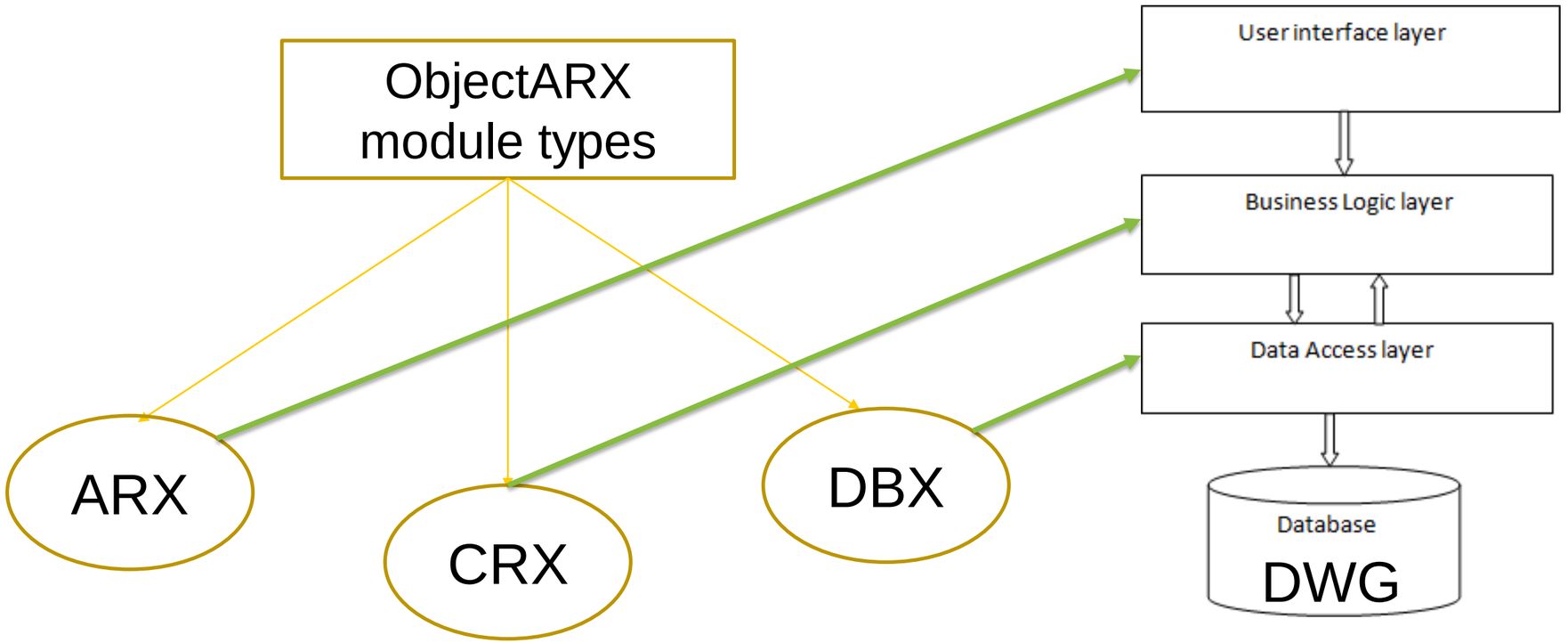
Visual Studio 2015 RTM or Update 1

Microsoft Windows Win7 or better

Main Libraries

- AcRx Object and Class Management (rxapi.lib)
- AcDb AutoCAD Database (acdb21.lib)
- AcGi AutoCAD Graphics Interface (acgiapi.lib)
- AcGe AutoCAD Geometry Library (acge21.lib)
- AutoCAD Core Library (accore.lib and ac1st21.lib)
- AcUi/AdUi MFC Extension Library (adui21.lib acui21.lib)

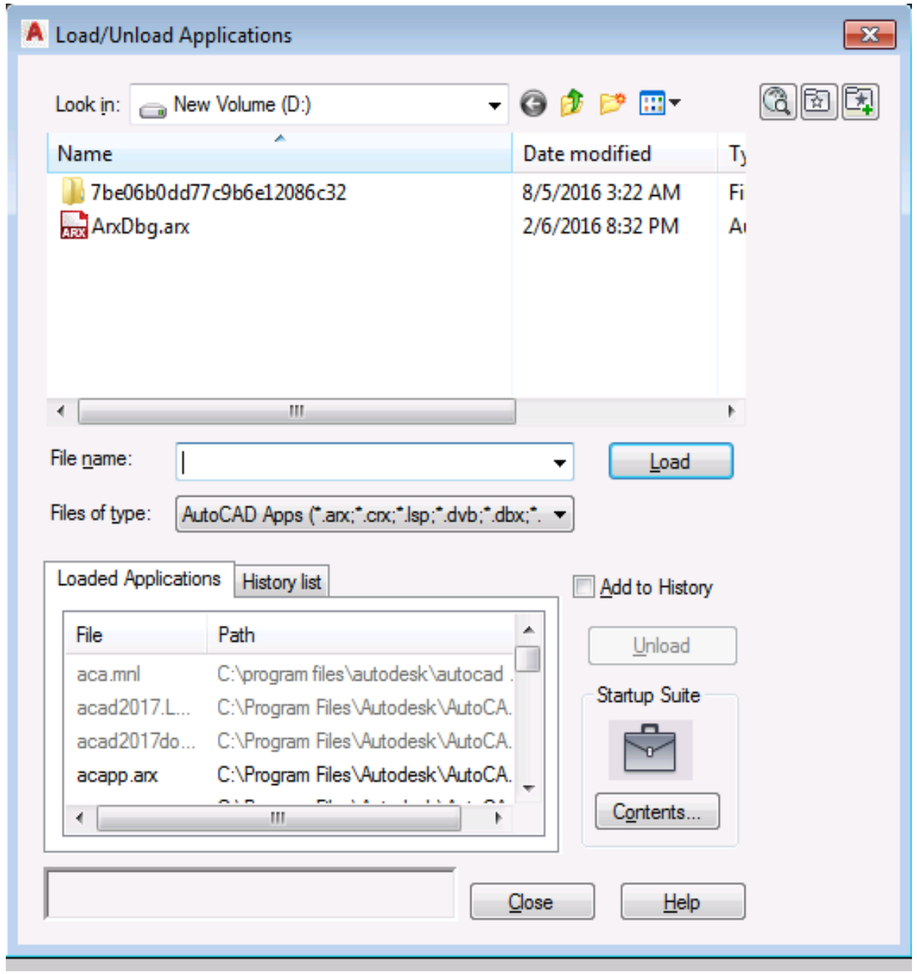
Types of modules



Loading ObjectARX Apps

- Loaded by the user
  - ARX & APPLOAD commands
  - Drag & Drop

- Demand loading



What is ObjectDBX and RealDWG

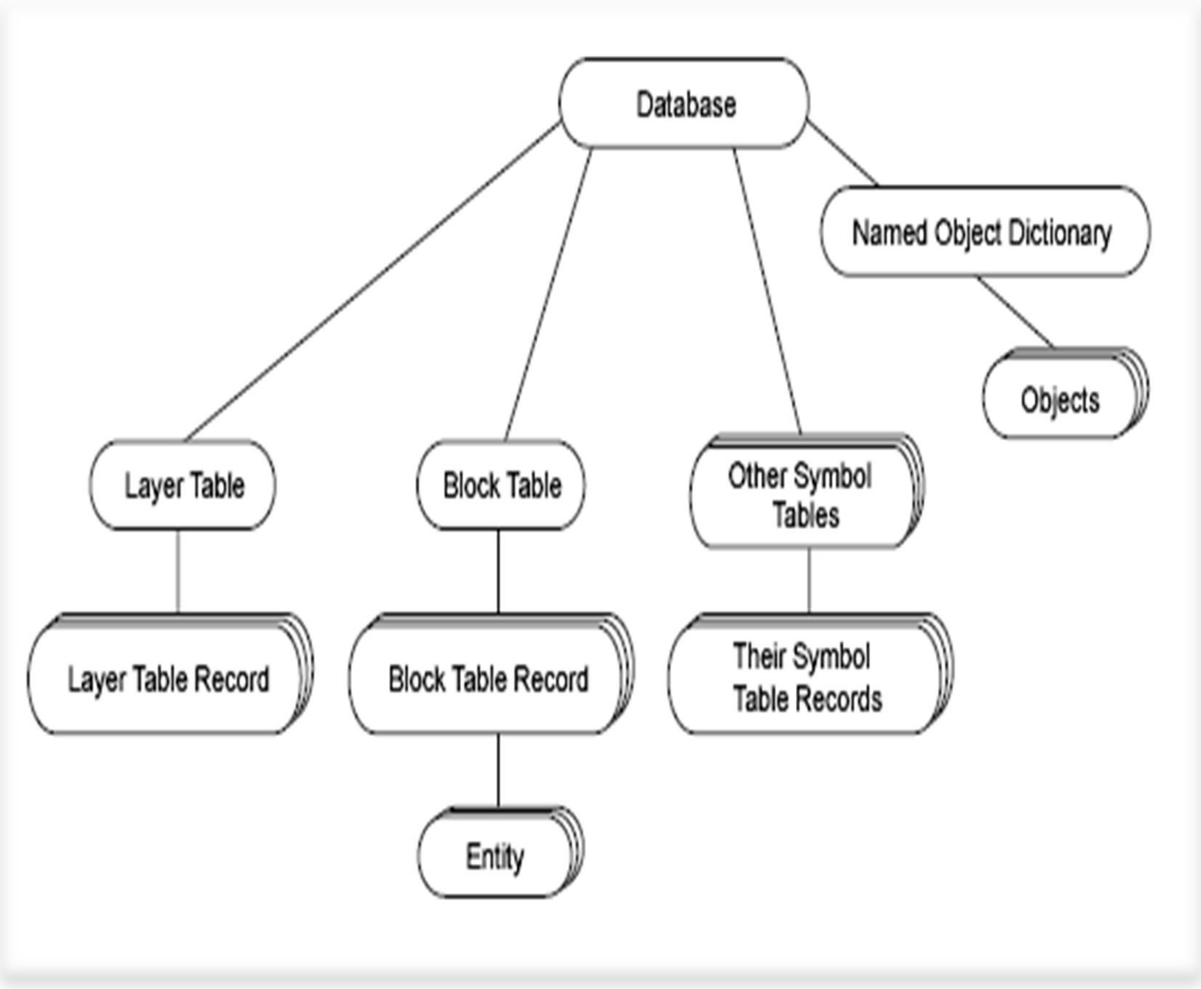
ObjectDBX is an AutoCAD-independent subset of ObjectARX

RealDWG is the license to use ObjectDBX in an own application outside of AutoCAD. RealDWG allows to read/write DWG files from your own application

What can I do with ObjectDBX?

- Create/Modify DWG/DXF files (.ARX)
- Create object enabler DLLs (.DBX)
- Can create standalone applications using RealDWG license

AcDbDatabase Owner Hierarchy



Object Identity

Handle (AcDbHandle)

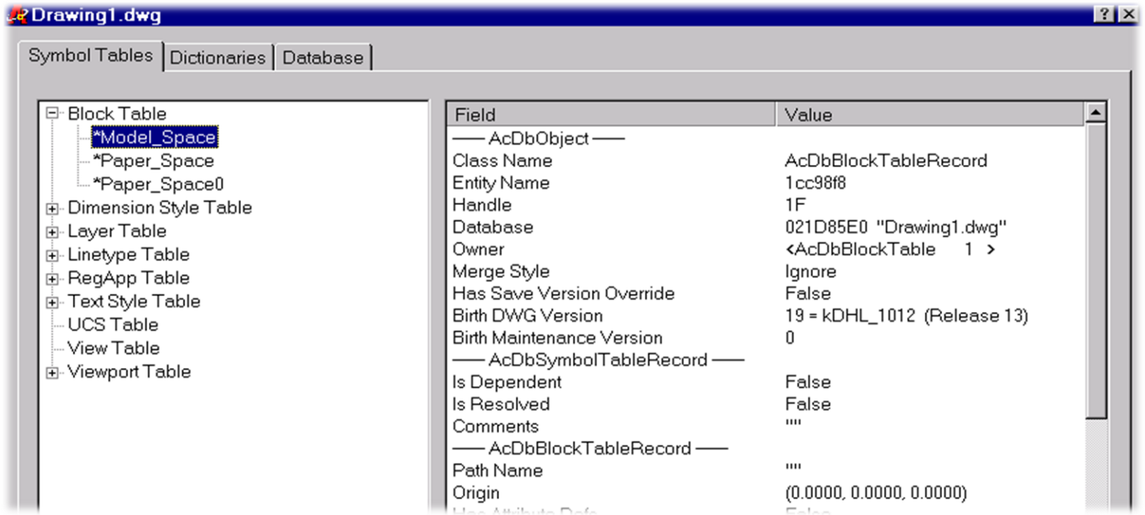
- Unique identifier of an object for the life of the drawing

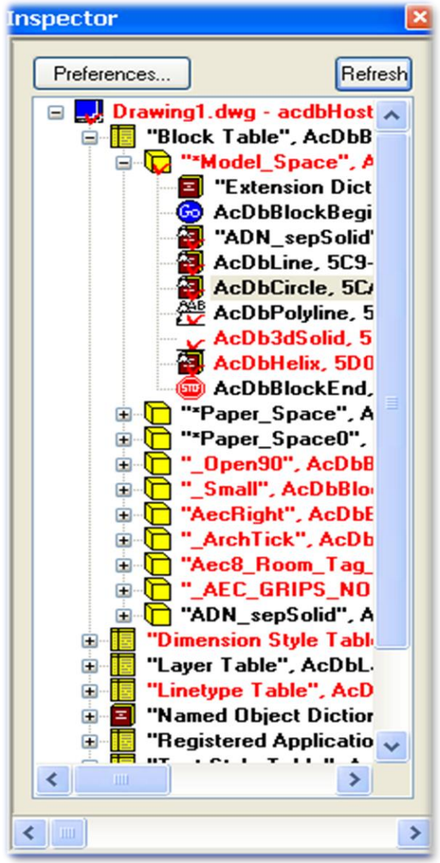
Object ID (AcDbObjectId)

- Unique identifier of an object for a session of ObjectDBX
- Multiple drawings allowed per session
- IDs unique across ALL files would require too much storage space (GUIDs)

Snoop tools for AutoCAD database

Tool	Language	Where to find
ArxDbg	C++	ObjectARX Samples
MgdDbg	C#	ADN site
Inspector	C++	ADN site





Quick Demo of ArxDbg tool

ObjectARX basics continued…

Important Objects And Ownerships

Block Table

- Three default records
  - *MODEL_SPACE
  - *PAPER_SPACE, *PAPER_SPACE0
- Only entities added to one of these is visible in AutoCAD editor
- Block table records only own entities
- AutoCAD’s “symbol tables”
- Named Objects Dictionary
  - Dictionaries own any objects (but not entities)

Create a new class

Derive from

- AcRxObject — RTTI protocol
- AcGiDrawable — + Graphics protocol
- AcDbObject — + Filing protocol
- AcDbEntity — + Persistent graphics
- AcDbCurve — + Curve entities
- …

ObjectDBX and host applications interact with your object through these interfaces

AcDbEntity Protocol

subGetOsnapPoints	-	Define osnap points
subGetGripPoints	-	Define grip points
subMoveGripPointsAt	-	Default is to call AcDbEntity::subTransformBy
subGetStretchPoints	-	Provide stretch points other than the grip points, default is to call subGetGripPoints
subMoveStretchPointsAt	-	Used by stretch, default is subTransformBy
subTransformBy	-	Needed for moving entity

AcDbEntity Protocol

subIntersectWith	- No default
subGetGeomExtents	- No default
subList	- Print specified data at command line
subExplode	- No default
worldDraw/viewPortDraw	- Defines graphical representation

Object Modeling Framework

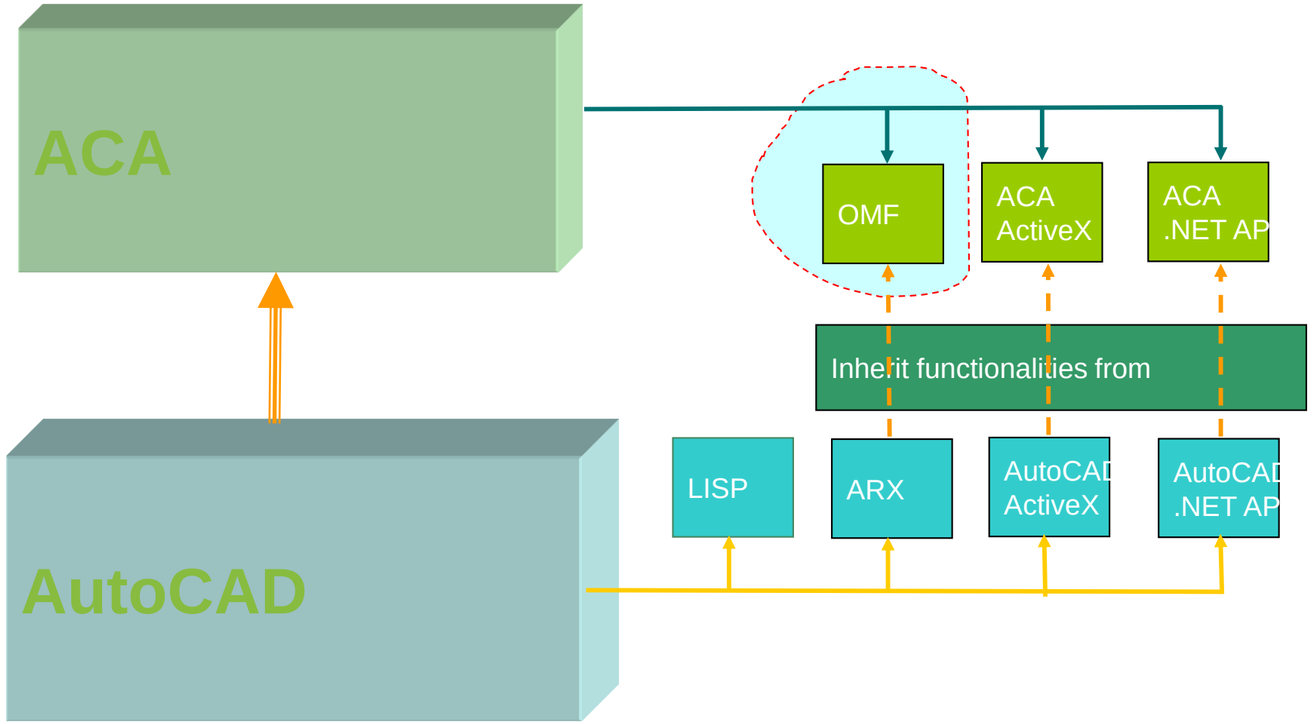
What is OMF?

- Object Modelling Framework
- A layer on top of ObjectARX just like MFC vs. Win32
- A powerful tool to extend ACA and also create your own ACA-like objects and entities
- Still can use any functionalities of ObjectARX
- More Object Oriented
- Latest version: 2017
- Available only to ADN members.

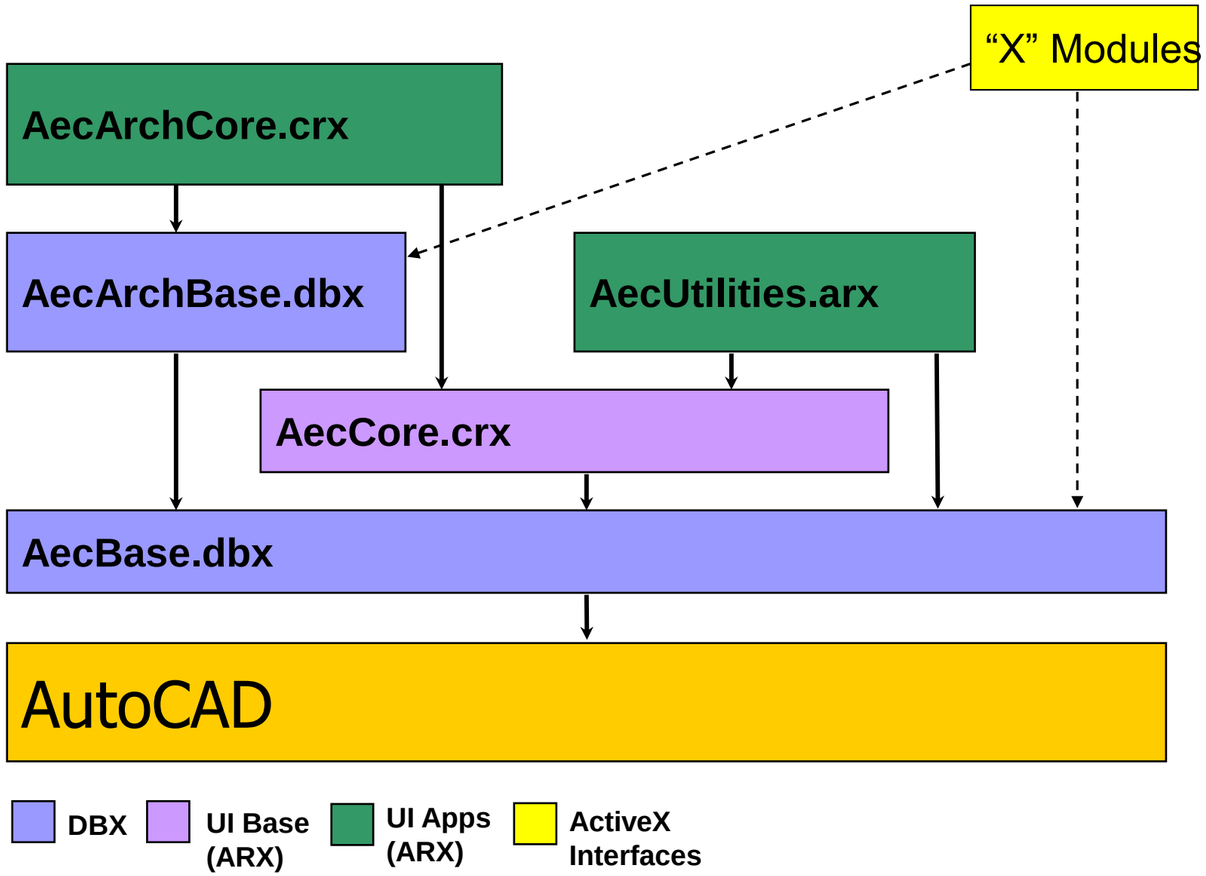
What is OMF for?

- Access ACA building model
- Create new ACA like objects and entities
- Extend ACA UI
- Separate physical and graphical data
- Build relation graph of all ACA and OMF objects
- Fully leverage object-oriented design
- Easy to extend and localize

Different ACA API



Component Architecture of OMF



OMF Vs ObjectARX

OMF Vs ObjectARX – App classes

In OMF, we need to derive from AecAppDbx for DBX projects and AecAppArx for ARX projects:

- Support DBX, ObjectEnabler strategy
- Support internationalization
- Plug into AEC OMF framework

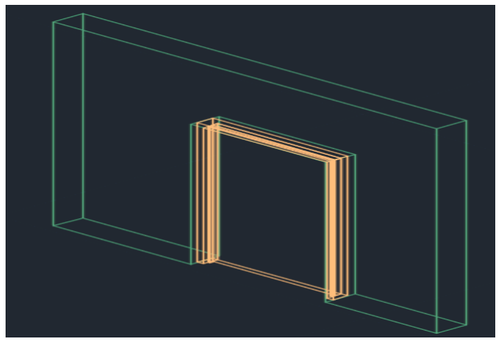
OMF Vs ObjectARX – draw()

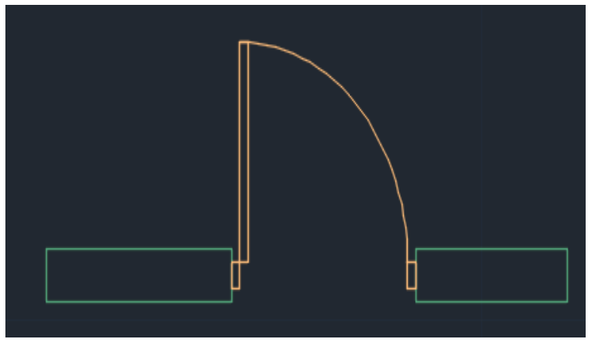
No need to implement graphics related functions in your custom entities:

- worldDraw(), viewportDraw()
- explode()
- getOnsapPoints()
- getGeomExtents()
- intersectWith()
- saveAs()

Instead, need to Implement the AecDbDispRep derived classes

- Draw()
- preDraw()
- postDraw()
- ...

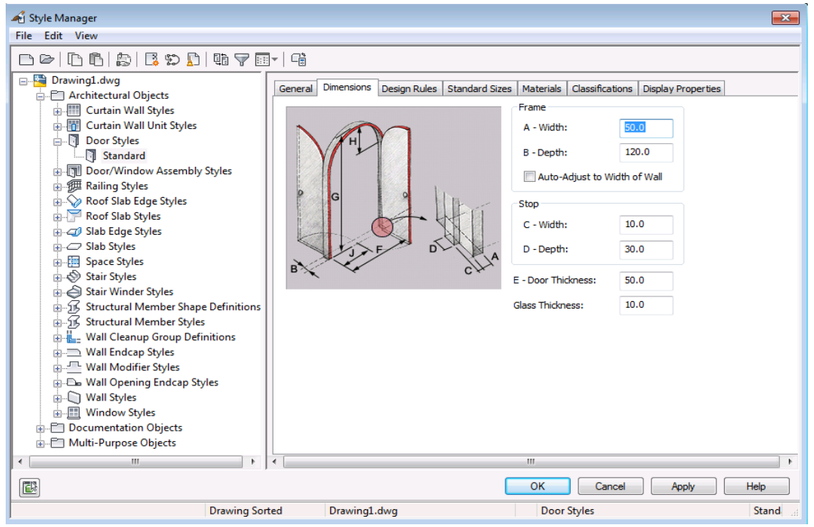




Quick Demo

OMF Vs ObjectARX – Dictionaries

- No need to care about:
  - Record merging
  - Name mangling
  - Purge
  - User Interface



- Derive from AecDbDictRecord and everything will be taken care of by OMF and ACA for you

- Users can manipulate all of these with a friendly and powerful UI, ACA Style Manager.

OMF Vs ObjectARX – Reactors

- All reactor use is “automatic” in AEC OMF
- Implement getReferencedObjects() to report who you are related to
- Implement message() or one of its dispatched functions to respond to objects you are related to.

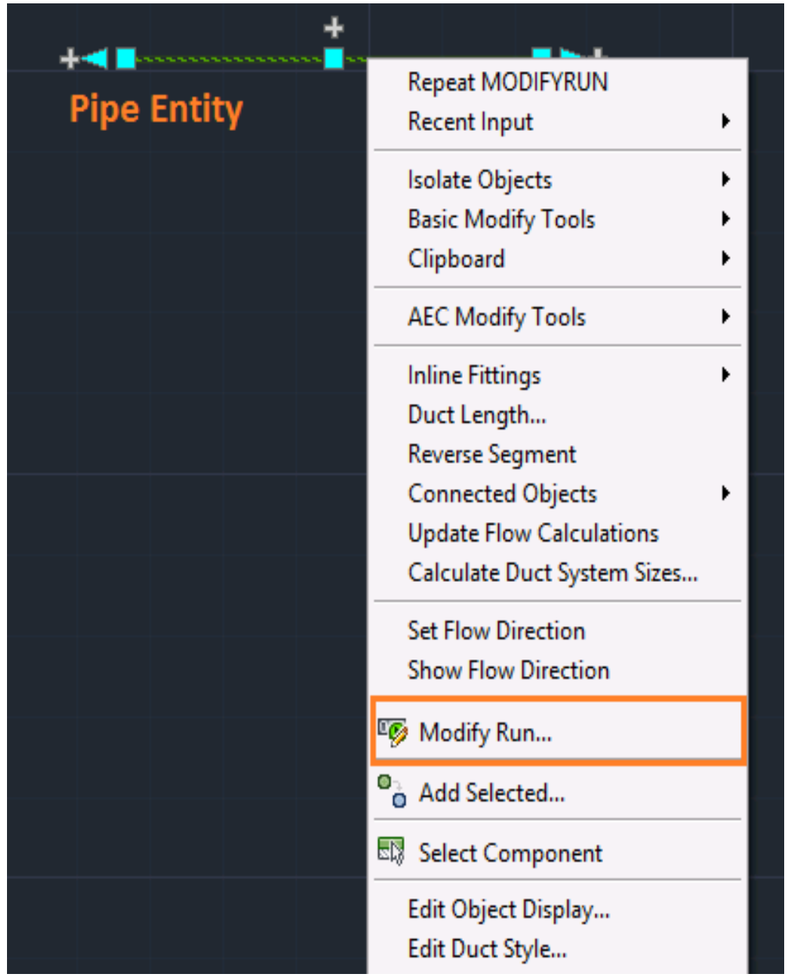
OMF Vs ObjectARX – transformBy()

- Inherit from AecDbGeo, which manages an object's ECS (Entity Coordinate System)
- Handles all transformBy() requests by forwarding them to an anchor
- You can completely change object's movement "rules" by simply swapping types of anchors attached
- transformBy() is implemented by the Anchor, not the entity

User interface customization in OMF

Commands

- Command classes, AecUiCmd - derived
  - Can implement worksWith for specific class(es) – for context menus and Ui customization
  - Must provide local name in the resources



Command line prompts

- AecUiPrBase - derived classes encapsulate acedGetXxx functions

- AecUiPrInt, AecUiPrDouble, AecUiPrString, AecUiPrPoint, AecUiPrDist(Def), AecUiPrAngle, ...

```
AecUiPrDistDef prDD("Radius", NULL, AecUiPrDist::kNoNeg, 55.0);
if ( AecUiPr::kOk == prDD.go()) {
  // do something with the double:
  prDD.value();
  // ...
}
```

Entity selection

- AecUiPrEntity - wraps acedEntSel()
- AecUiPrEntitySet - wraps acedSsGet(NULL, NULL,...)
- AecUiPrEntitySetSingle - same but one entity only
  - Filter based on type (isKindOf or isA)
  - Filter out entities on locked layers
  - Last two support pickfirst selection
  - Access object Id(s) directly - no need to handle ads_name(s)

Quick Demo

Selection sets

- AecAcadSelSet encapsulates acedSSGet
  - No need to free selection set when reusing the instance
  - Return AcDbObjectIdArray using asArray (no ads_name)
  - Methods (fenceSelect, boxSelect, lastSelect,...) replace the old selection code.
- E.g. acedSSGet(“F”, ptlist, NULL, NULL, ss) becomes:

```
AecAcadSelSet ss;
ss.fenceSelect(ptList, NULL);

// Do something with:
AcDbObjectIdArray selectedEnts;
ss.asArray(selectedEnts);
```

Helper functions

- extern Acad::ErrorStatus addNewSymbolAndClose(AcDbSymbolTable* symTbl, AcDbSymbolTableRecord* newRecord);
- extern Acad::ErrorStatus addNewLayer(LPCTSTR layerName, AcDbDatabase* db);

- extern Acad::ErrorStatus addToModelSpace(AcDbEntity* newEnt, AcDbDatabase* db);
- extern Acad::ErrorStatus addToModelSpaceAndClose(AcDbEntity* newEnt, AcDbDatabase* db);

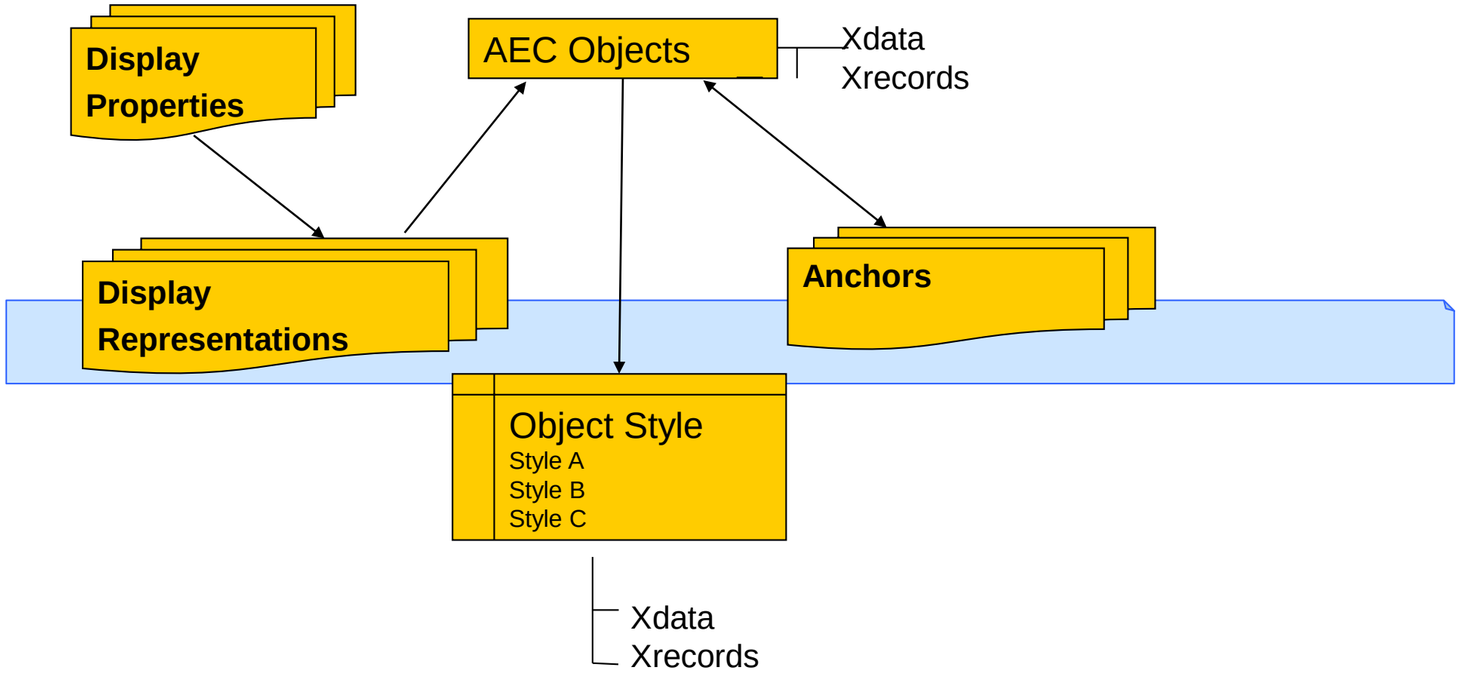
- extern Acad::ErrorStatus addToPaperSpace(AcDbEntity* newEnt, AcDbDatabase* db);
- extern Acad::ErrorStatus addToPaperSpaceAndClose(AcDbEntity* newEnt, AcDbDatabase* db);

- extern Acad::ErrorStatus addToCurrentSpace(AcDbEntity* newEnt, AcDbDatabase* db);
- extern Acad::ErrorStatus addToCurrentSpaceAndClose(AcDbEntity* newEnt, AcDbDatabase* db);

- // the following two functions assume acdbCurDwg() for the database (which is usually the case for jigs and add routines)
- extern Acad::ErrorStatus addToCurrentSpace(AcDbEntity* newEnt);
- extern Acad::ErrorStatus addToCurrentSpaceAndClose(AcDbEntity* newEnt);

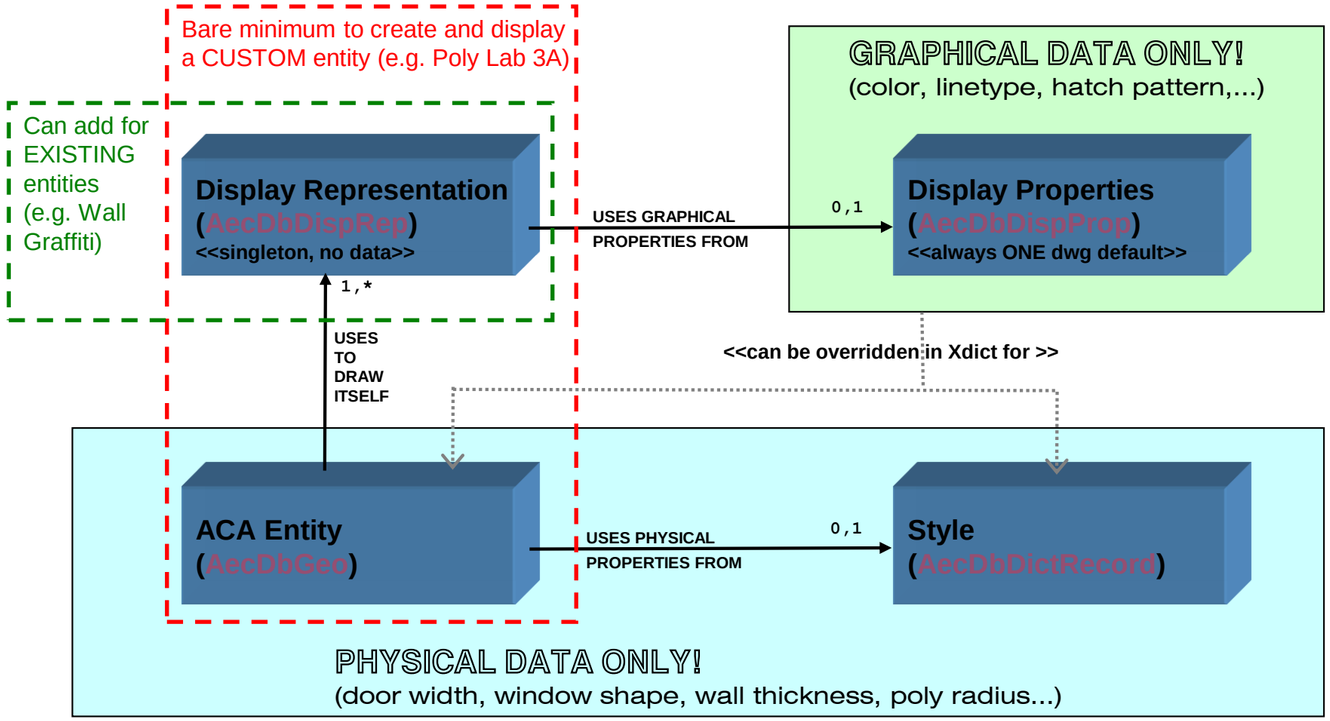
AEC Entities And Objects

- AEC Entities
- Anchors
- Styles
- Display Representations
- Display Properties

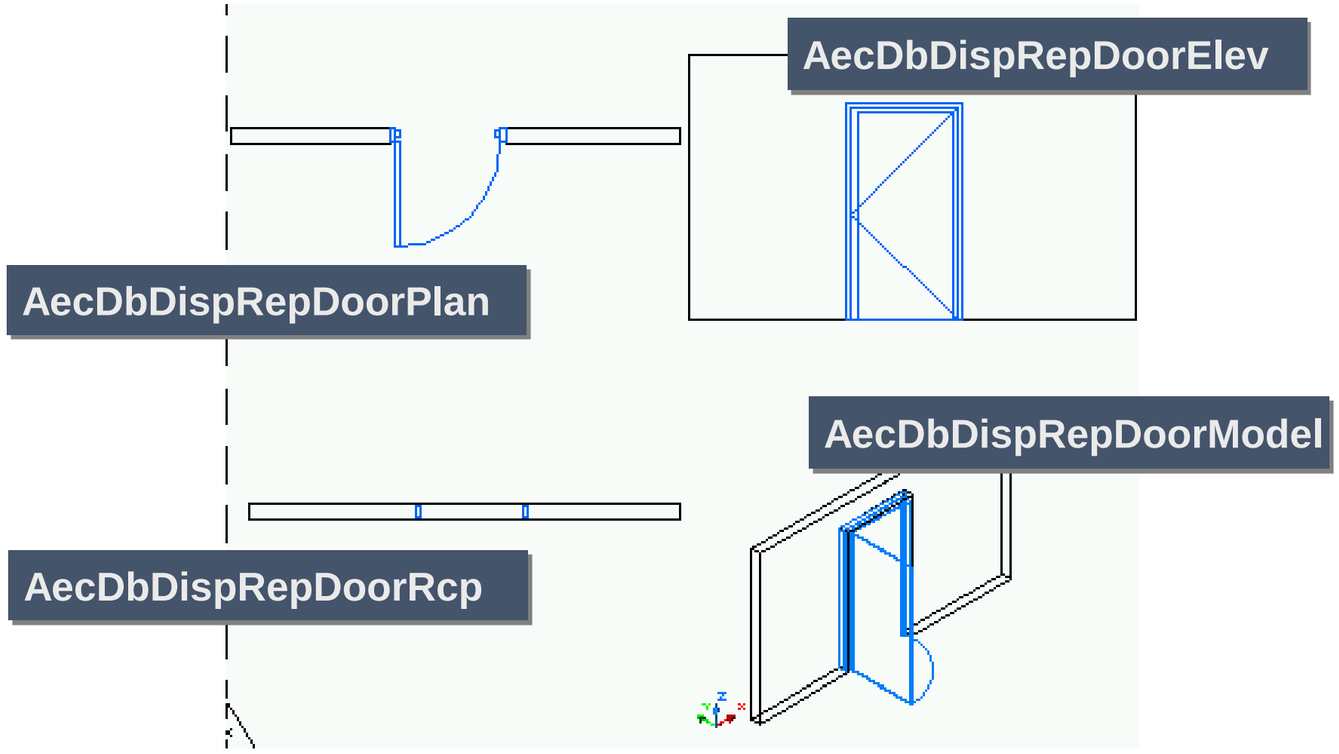


ACA Drawing File

Interaction of AEC objects



Display Representations



Display Properties

- Provide graphical data for DispReps
- Always one “System default” instance (per dwg)
- Can be overridden on style or entity level
  - Attached to specific Extension Dictionary
  - Automatic UI: right-click Y “Edit Object Display...”
- Contain:
  - AecDbDispCompEnt (Visibility, ByMaterial, Layer, Color, Linetype, Lineweight, LT Scale, Plotstyle)
  - AecDbDispCompHatch (Type, Scale, Pattern Angle)
  - Any other custom data (e.g. Cut Plane Height)

Display Manager - AecDispRepMgr

- Center of the Display System
- Have many helper functions to access DispRepConfig, DispReps and so on.
- All calls for draw are routed through this class.
- Add new display representations through it.
- Draw entities through it.
- Query which DispRep for a specific entity in a specific view direction.

Quick Demo

What are streams?

- Set of related classes that accept graphical input and process it into particular form
- Each derived stream optimized for a particular purpose
- All streams are derived from AecStreamAcad base class

How ACA uses streams?

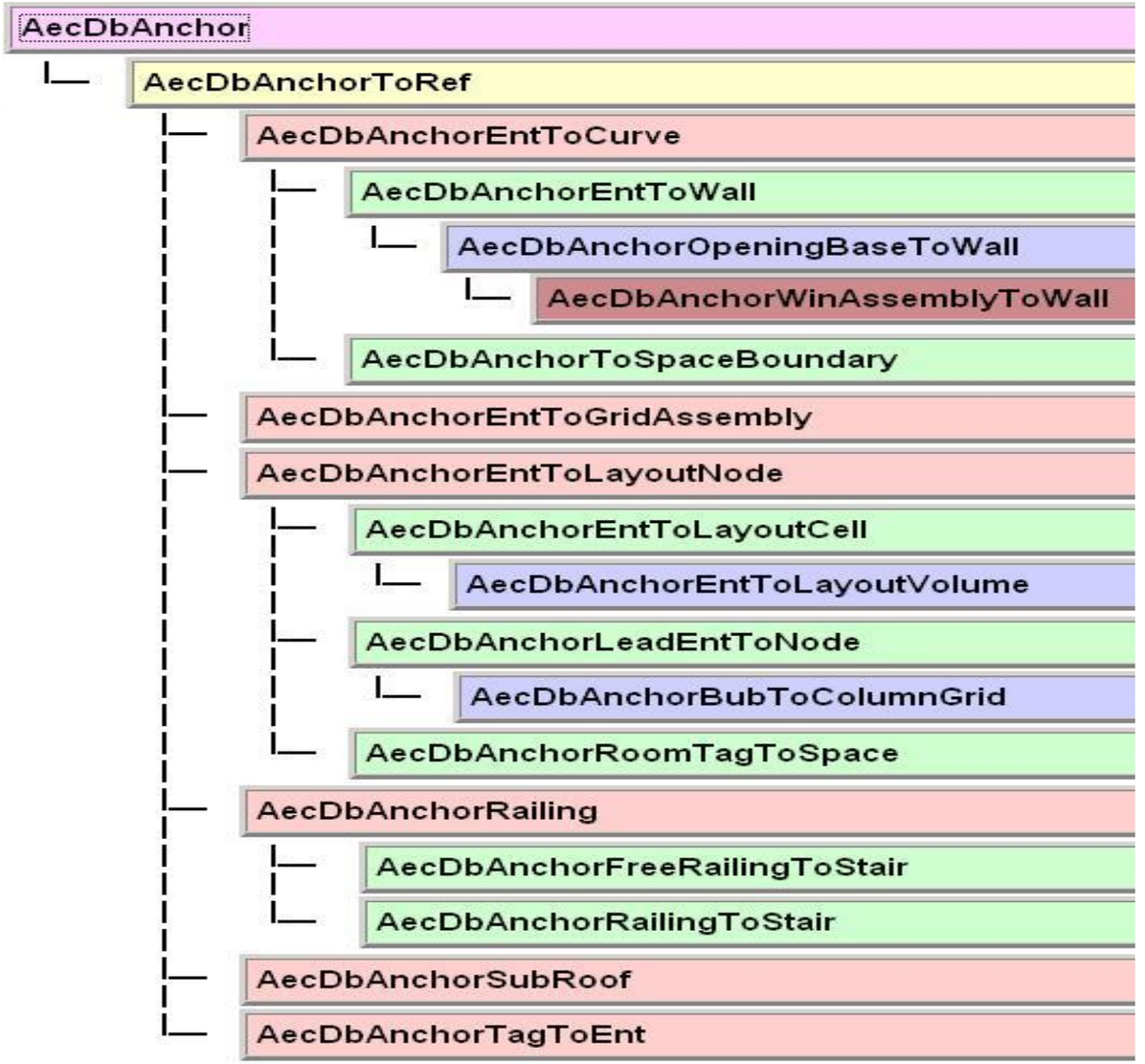
- AecDbEntity::viewPortDraw()
  - Draw (stream) ourselves into **AecStreamAcGi**
  - Collect resulting AcGi geometry
- AecDbEntity::intersectWith()
  - Draw (stream) ourselves into **AecStreamIntersect**
  - Perform intersectWith with another AecStreamIntersect
- AecDbEntity::explode()
  - Draw (stream) ourselves into **AecStreamExplode**
  - Collect anonymous block or AcDb entities
- AecDbEntity::getOsnapPoints()
  - Draw (stream) ourselves into **AecStreamOsnap**
  - Collect points based on filter
- AecDbEntity::getGeomExtents()
  - Draw (stream) ourselves into **AecStreamExtent**
  - Collect bounding points

Anchors

- AecDbAnchor-derived object (not an entity!) that establishes a spatial relationship between:
  - Anchored object (must be an AecDbGeo)
  - Referenced object(s) (can be any AcDbEntity)
- Automatically records “kOwnedBy/kOwnerOf” between the anchored and referenced objects
- AecDbAnchor::updateEcs called whenever the owned entity or owner geometrically modified
  - can veto or adjust any change in anchored entity’s ECS

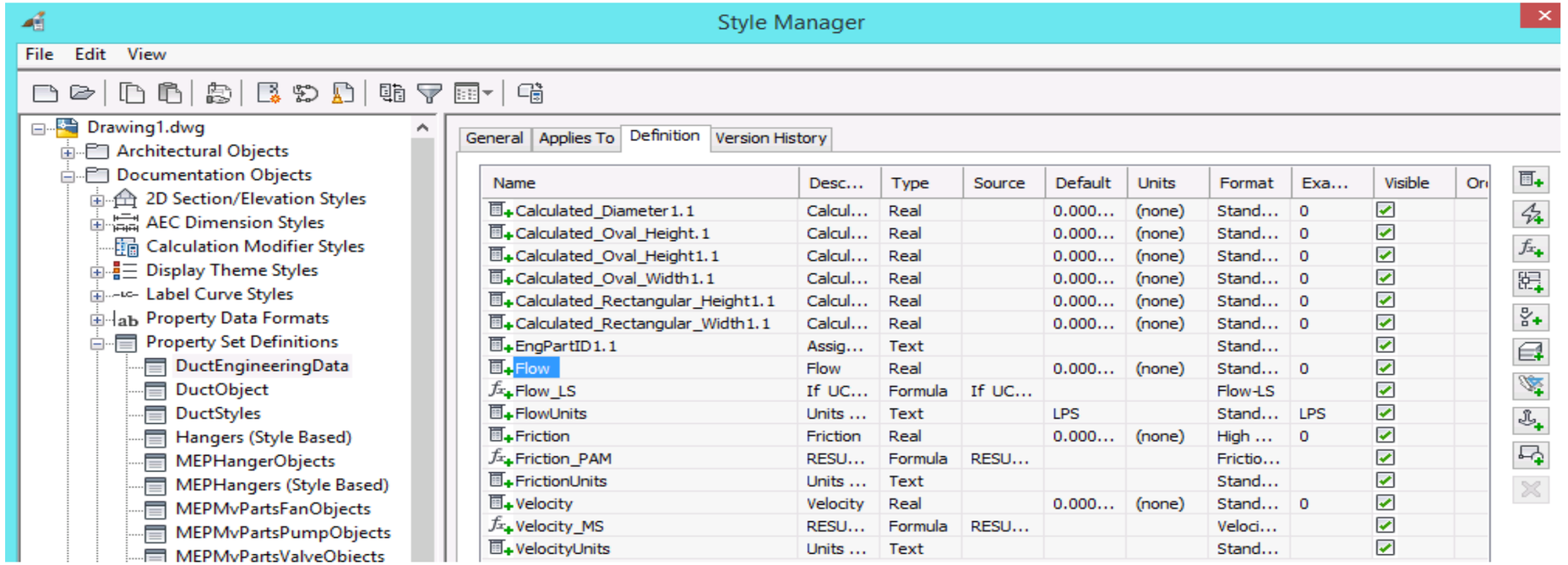
Anchor types

- Derived classes can decide to control the movement in any specific way
- You can design custom anchors in OMF:



Schedule and property sets

- Everything concerning Property Set Definitions and Property Sets is exposed
- Schedule tables have been exposed since OMF 2004



How to get OMF SDK?

- OMF is available for download to ADN members (Please visit adn.autodesk.com)

- .NET API (wrapper on OMF) are available for free and is installed along with ACA/MEP.


Q&A

More Questions? Visit the AU Answer Bar

- Seek answers to all of your technical product questions by visiting the **Answer Bar**.
- Open daily from **8am-6pm Tuesday** and **Wednesday**; **8am-4:30pm Thursday**.
- Located outside **Hall C, Level 2**.
- Meet Autodesk developers, testers, & support engineers ready to help with your most challenging technical questions.