Autodesk University 2008

Inventor® API:
New Features in Inventor 2009

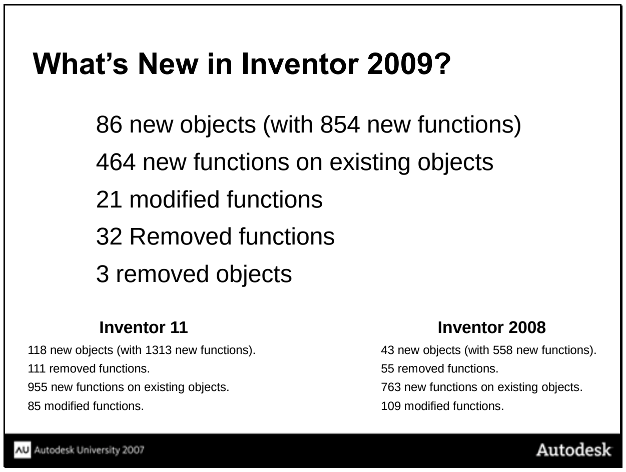
Brian Ekins – Autodesk

DE311-1 This session discusses and demonstrates the enhancements made to the Inventor programming interface for Inventor 2009.

About the Speaker:

Brian is a designer for the Autodesk Inventor® programming interface. He began working in the CAD industry over 25 years ago in various positions, including CAD administrator, applications engineer, CAD API designer, and consultant. Brian was the original designer of the Inventor® API and has presented at conferences and taught classes throughout the world to thousands of users and programmers.

brian.ekins@autodesk.com



What's New in Inventor 2009?

86 new objects (with 854 new functions)

464 new functions on existing objects

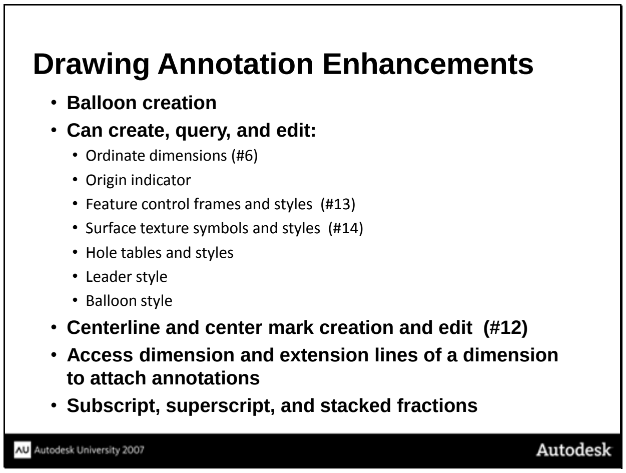
21 modified functions

32 Removed functions

3 removed objects

Inventor 11	Inventor 2008
118 new objects (with 1313 new functions).	43 new objects (with 558 new functions).
111 removed functions.	55 removed functions.
955 new functions on existing objects.	763 new functions on existing objects.
85 modified functions.	109 modified functions.

AU Autodesk University 2007 Autodesk

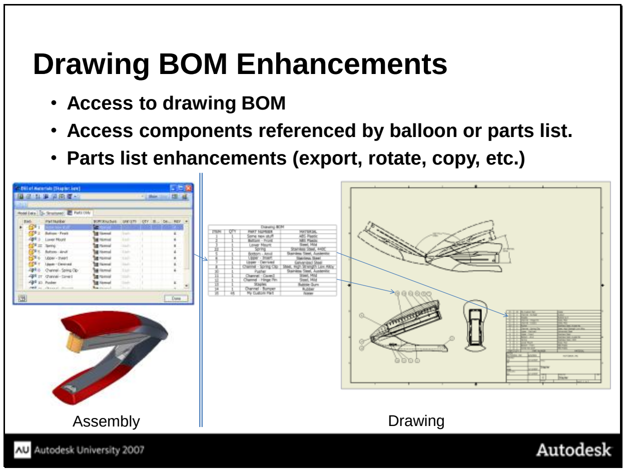


Drawing Annotation Enhancements

- **Balloon creation**
- **Can create, query, and edit:**
  - Ordinate dimensions (#6)
  - Origin indicator
  - Feature control frames and styles (#13)
  - Surface texture symbols and styles (#14)
  - Hole tables and styles
  - Leader style
  - Balloon style
- **Centerline and center mark creation and edit (#12)**
- **Access dimension and extension lines of a dimension to attach annotations**
- **Subscript, superscript, and stacked fractions**

AU Autodesk University 2007 Autodesk

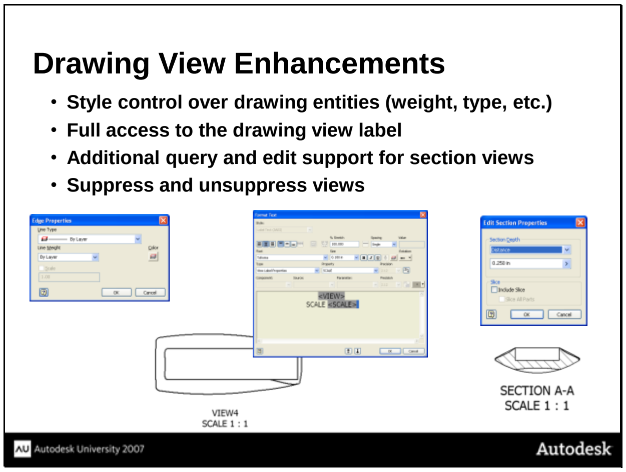
- Previous versions of the API supported query and edit of balloons, but not creation. You can now create balloons. **(Demo with AntiVibrationMount.iam and macro modDrawing.CreateBalloon)**
- There is now support for ordinate dimensions. Support is for single ordinate dimensions, not ordinate groups. **(Demo with AntiVibrationMount.idw and macro modDrawing.CreateOrdinateDimensions)** Number 6 on the ADN wishlist.
- The origin indicator is used to define the origin for ordinate dimensions and hole tables. You can now create and edit these. **(Demonstrated as part of ordinate dimension creation.)**
- Full support for feature control frames and surface texture symbols is now supported. **(Demo with AntiVibrationMount.idw and macro modDrawing.AddFeatureControlFrame and EditFeatureControlFrame)** These are number 13 and 14 on the ADN wishlist.
- Full support for hole tables including the associated style. **(Demo with sheet 2 in AntiVibrationMount.idw and macro modDrawing.CreateHoleTables)**
- Support for leader and balloon styles. The API already supported the creation and edit for leaders and balloons.
- The API supports the ability to create and do some limited editing of centerlines and marks. This is number 12 on the ADN wishlist. **(Demo with sheet 2 in AntiVibrationMount.idw and macro modDrawing.AddCenterMarks)**
- You can attach annotations directly to dimension and extension lines. **(Demo with any drawing and macro modDrawing.AddSurfaceTextureSymbol)**
- Support for more advanced text formatting. **(Demo with any drawing and macro modDrawing.StackedText)**



Drawing BOM Enhancements

- Access to drawing BOM
- Access components referenced by balloon or parts list.
- Parts list enhancements (export, rotate, copy, etc.)

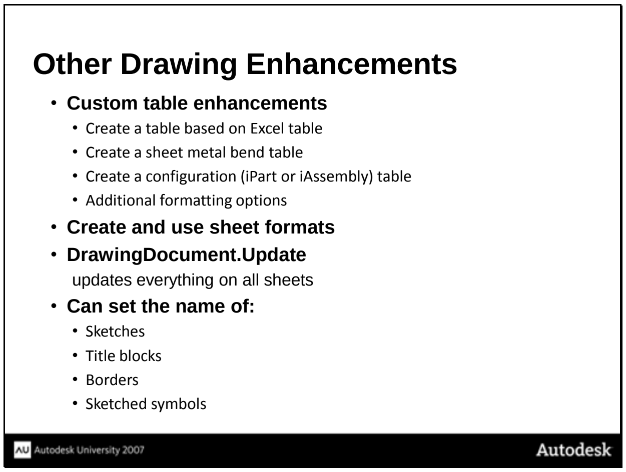
- The drawing BOM is a new API concept. It is the assembly BOM plus drawing overrides. Balloons and part list rows reference a row from the drawing BOM to determine their contents. The picture illustrates that the Assembly BOM provides the master definition of all BOM data. The drawing BOM is internal to the drawing and isn't ever directly seen by the end-user. It represents the assembly BOM plus any overrides that have been defined in the drawing. For example you can add additional rows to a parts list or modify fields of a parts list. These changes are reflected in the drawing BOM but not the assembly BOM. Any changes made to the assembly BOM are automatically reflected in the drawing BOM. **Can be demonstrated using modDrawing.ShowDrawingBOM**



Drawing View Enhancements

- Style control over drawing entities (weight, type, etc.)
- Full access to the drawing view label
- Additional query and edit support for section views
- Suppress and unsuppress views

- Can now set the line type, line weight, line scale, and color of drawing entities.
- Full access to the drawing label. There was limited access before, but now you have access to the full text and other settings associated with a drawing label.
- Full section view support. This is primarily the addition of the functionality shown in the "Edit Section Properties" dialog.
- Ability to set the visibility of a drawing view (suppress).

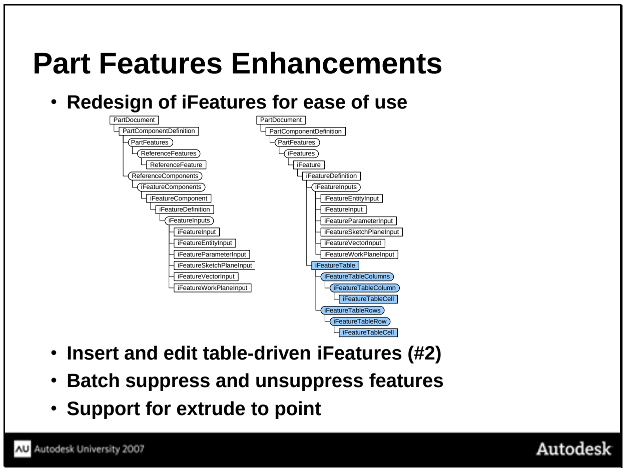


Other Drawing Enhancements

- **Custom table enhancements**
  - Create a table based on Excel table
  - Create a sheet metal bend table
  - Create a configuration (iPart or iAssembly) table
  - Additional formatting options
- **Create and use sheet formats**
- **DrawingDocument.Update**
  updates everything on all sheets
- **Can set the name of:**
  - Sketches
  - Title blocks
  - Borders
  - Sketched symbols

Autodesk University 2007 Autodesk

- Several significant enhancements to custom tables. You can create tables that are:
  - associatively tied to the contents of an Excel spreadsheet. **(Demo with any drawing and macro modDrawing.CreateExcelTable)**
  - display the bends of a sheet metal part.
  - display a configuration table for an iPart or iAssembly **(Demo with any drawing and macro modDrawing.CreateConfigurationTable)**
- The ability to create sheet formats and use them when creating a new sheet is now supported.
- The DrawingDocument.Update method will now update the contents of all sheets, whether they're active or not.
- The Name property of several objects is now read-write so that the name can be set. The ability to rename these objects is new in Inventor 2009.



Part Features Enhancements

- **Redesign of iFeatures for ease of use**

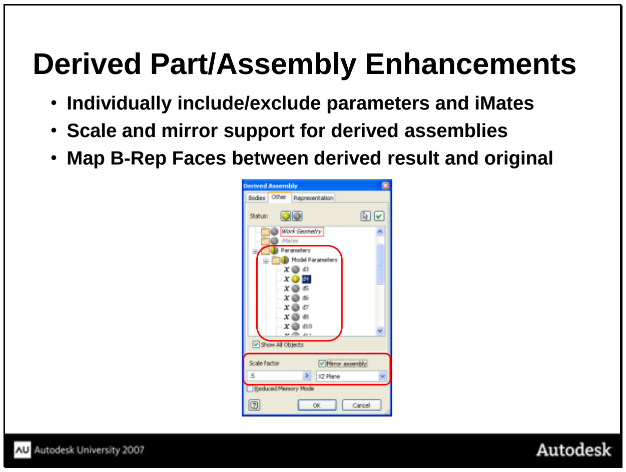
PartDocument
PartComponentDefinition
PartFeatures
ReferenceFeatures
ReferenceFeature
ReferenceComponents
iFeatureComponents
iFeatureComponent
iFeatureDefinition
iFeatureInputs
iFeatureInput
iFeatureEntityInput
iFeatureParameterInput
iFeatureSketchPlaneInput
iFeatureVectorInput
iFeatureWorkPlaneInput

PartDocument
PartComponentDefinition
PartFeatures
iFeatures
iFeature
iFeatureDefinition
iFeatureInputs
iFeatureEntityInput
iFeatureInput
iFeatureParameterInput
iFeatureSketchPlaneInput
iFeatureVectorInput
iFeatureWorkPlaneInput
iFeatureTable
iFeatureTableColumns
iFeatureTableColumn
iFeatureTableCell
iFeatureTableRows
iFeatureTableRow
iFeatureTableCell

- **Insert and edit table-driven iFeatures (#2)**
- **Batch suppress and unsuppress features**
- **Support for extrude to point**

Autodesk University 2007 Autodesk

- The API interface for iFeatures has been redesigned to make them easier to understand and use. For backward compatibility, the old interface still works but has been hidden. With the new interface, an iFeature is like all other part features; it is represented by a single object and is accessible from the PartFeatures collection. **(Can be demonstrated using modPartFeatures.PlaceiFeature)**
- In addition to simplifying the iFeature interface, support for table driven iFeatures has also been added. This was item number 2 from the ADN Wishlist. **(Can be demonstrated using modPartFeatures.PlaceTableDriveniFeature And modPartFeatures.EditTableDriveniFeature)**
- To help with performance there is now the ability to batch suppress and unsuppress a list of features. (SuppressFeatures method supported by the various ComponentDefinition objects.)
- The extrude command now supports the ability to extrude to a point. This is also supported through the API.



Derived Part/Assembly Enhancements

- Individually include/exclude parameters and iMates
- Scale and mirror support for derived assemblies
- Map B-Rep Faces between derived result and original

☑ Show All Objects

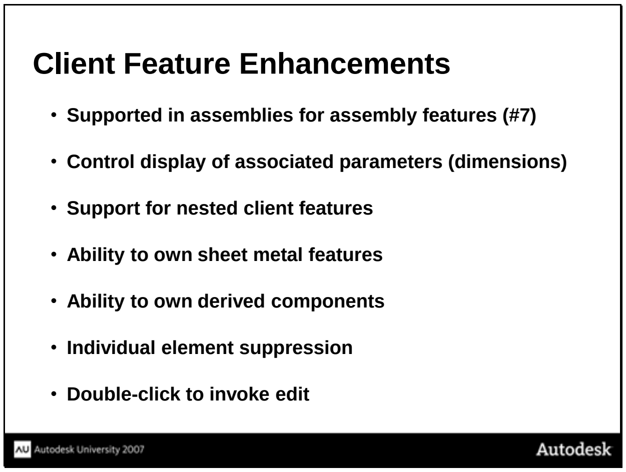
Scale factor .5

☑ Mirror assembly XZ Plane

□ Reduced Memory Mode

AU Autodesk University 2007 Autodesk

- The interface to derived parts supports some additional capability to control using parameters and iMates. There is also support to control scaling and mirroring when creating a derived assembly.
- Can map faces from the derived result to the original part using the Face.ReferencedEntity property.

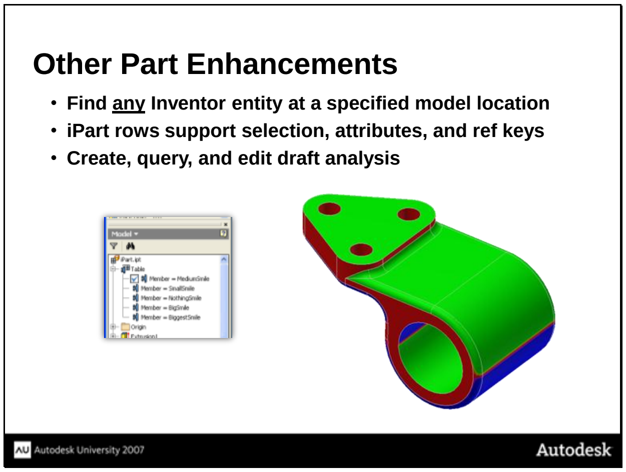


Client Feature Enhancements

- Supported in assemblies for assembly features (#7)
- Control display of associated parameters (dimensions)
- Support for nested client features
- Ability to own sheet metal features
- Ability to own derived components
- Individual element suppression
- Double-click to invoke edit

AU Autodesk University 2007 Autodesk

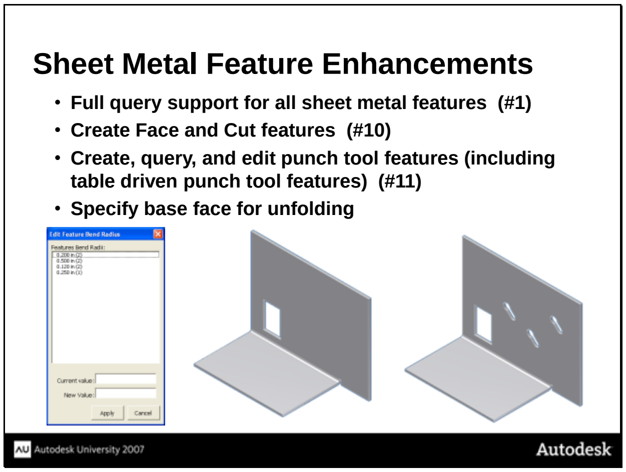
- Client features have been enhanced for Inventor 2009. One big enhancement is the ability to create client features within an assembly. Client features in an assembly support all of the same functionality as client features in a part. They are limited to grouping assembly features and do not provide any capabilities for working with occurrences.
- The enhancements for this release are:
  - Specify parameters that are to be hidden. This uses the ClientFeature.HiddenParameters property.
  - Client features can now contain other client features.
  - Client features can now contain sheet metal features and derived components.
  - You can control suppression of the individual components within the client feature. Previously the suppression was controlled as a group. **You can demo this using the Fillet_Chamfer.ipt file and the modClientFeature.FilletChamfer macro.**
  - There's now an event to notify when the client feature is double-clicked. You can provide edit capabilities in response to this event. **Much of this can be demonstrated using PocketFeature.ipt and the macro modClientFeature.PocketClientFeature macro.**



Other Part Enhancements

- Find any Inventor entity at a specified model location
- iPart rows support selection, attributes, and ref keys
- Create, query, and edit draft analysis

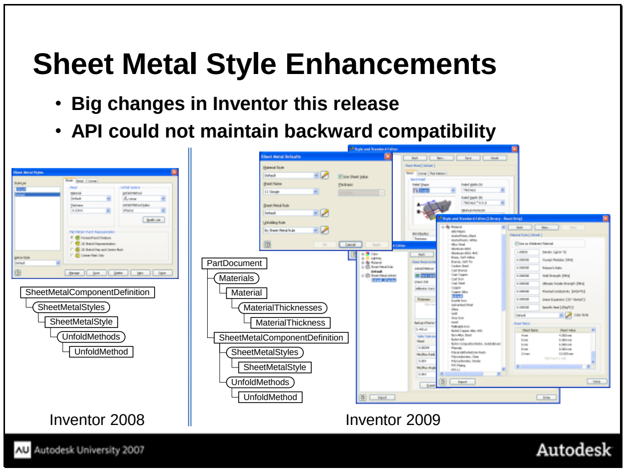
- The FindUsingPoint method of the part and assembly component definition objects supports a programmatic selection by finding any type of object that is within a tolerance of a given point.
- iPartRows are now a full fledged object and supports selection within the browser and the ability to add attributes and create reference keys.



Sheet Metal Feature Enhancements

- Full query support for all sheet metal features (#1)
- Create Face and Cut features (#10)
- Create, query, and edit punch tool features (including table driven punch tool features) (#11)
- Specify base face for unfolding

- Have full query access to all sheet metal features. Even though this is limited to query it's still possible to perform some limited edits because the query functions provide access to the associated parameters and their values can be edited. Query support for sheet metal features was the number 1 requested enhancement from the ADN wishlist. **The sample modSheetMetal.QuerySheetMetalBends can be used to demonstrate some of this.** It looks for any features that support bends and then displays them in a dialog where you can change the bend radius.
- Support for the creation of Face and Cut features. Because Face features will create automatic bends where the face butts up to an existing model, it's possible to get some bends as a side effect. This was item number 10 from the ADN Wishlist. **This is demonstrated with the sample modSheetMetal.FaceAndCutFeatureCreation**
- There's full support for placing, querying, and editing punch tool features. This includes table driven punches. This was item number 11 from the ADN Wishlist. **This is demonstrated with the sample modSheetMetal.PlacePunchFeature**



Sheet Metal Style Enhancements

- Big changes in Inventor this release
- API could not maintain backward compatibility

SheetMetalComponentDefinition
SheetMetalStyles
SheetMetalStyle
UnfoldMethods
UnfoldMethod

Inventor 2008

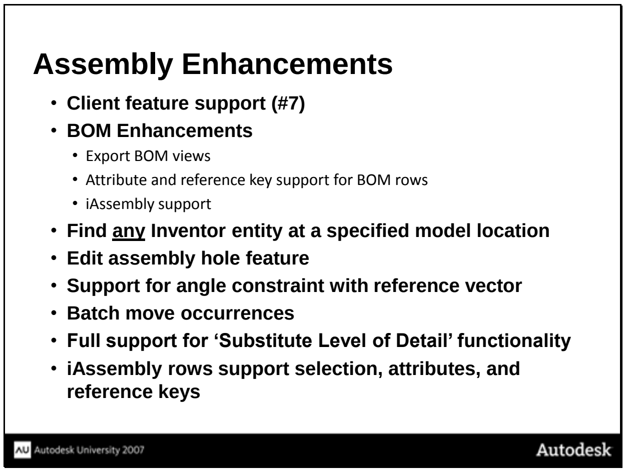
PartDocument
Materials
Material
MaterialThicknesses
MaterialThickness
SheetMetalComponentDefinition
SheetMetalStyles
SheetMetalStyle
UnfoldMethods
UnfoldMethod

Inventor 2009

Autodesk University 2007

Autodesk

- The concept of sheet metal styles has been radically changed with Inventor 2009. The changes were big enough that we were unable to provide any compatibility with the previous API. If you used sheet metal styles you'll need to rewrite that portion of your code.
- In Inventor 2008 there were sheet metal styles that encapsulated everything. These were stored within the sheet metal document. In Inventor 2009 the functionality of a sheet metal style has been broken up into two separate concepts; sheet metal rules and sheet metal unfold definitions. Both of these are now stored as styles external to any sheet metal document, like all other styles.
- The sheet metal rules are essentially the equivalent of the previous sheet metal style except some things have been removed. The API still uses the SheetMetalStyle object to represent this information. A major change was made because of the fact that these styles are now defined outside of a sheet metal document. Previously, many of the settings returned Parameter objects that you would then use to set the desired value. Now these values are stored as Strings, so the API reflects this. The parameters that are set by the active style and drive the sheet metal part are now available on the SheetMetalComponentDefinition object.
- Unfold methods are similar to before except they're no longer associated with a sheet metal style but can be created and used independently of any style.
- **The samples modSheetMetal.CreateSheetMetalStyle, modSheetMetal.SheetMetalStyleDisplay, and modSheetMetal.SetSheetMetalThickness can be used to demonstrate these.**

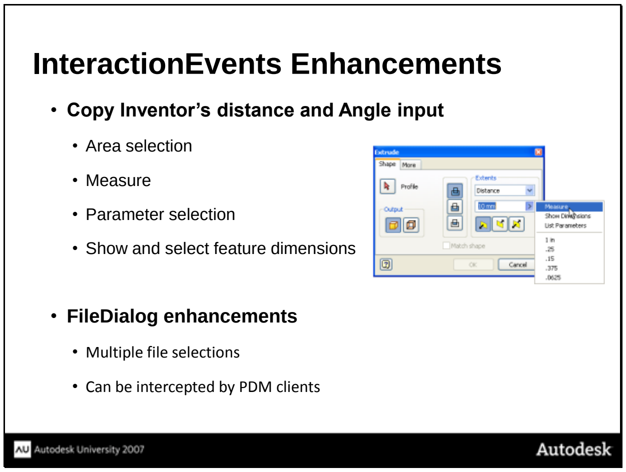


Assembly Enhancements

- Client feature support (#7)
- BOM Enhancements
  - Export BOM views
  - Attribute and reference key support for BOM rows
  - iAssembly support
- Find any Inventor entity at a specified model location
- Edit assembly hole feature
- Support for angle constraint with reference vector
- Batch move occurrences
- Full support for 'Substitute Level of Detail' functionality
- iAssembly rows support selection, attributes, and reference keys

AU Autodesk University 2007 Autodesk

- **The sample modAssembly.BOMExport demonstrates writing the BOM to an Excel spreadsheet.**
- Support for substitute parts within an assembly. The API provides support for this functionality.
- The angle constraint was enhanced to allow an extra input to define a reference vector to provide consistent solutions in all cases. The API now provides the ability to also specify this reference vector.
- Can edit assembly hole features in the same way you can edit part hole features.
- In the case you need to move many occurrences you can use the new AssemblyComponentDefinition.TransformOccurrences method to move them all at once. This will provide significant performance benefits over moving them one at a time.
- The FindUsingPoint method of the part and assembly component definition objects supports a programmatic selection by finding any type of object that is within a tolerance of a given point.

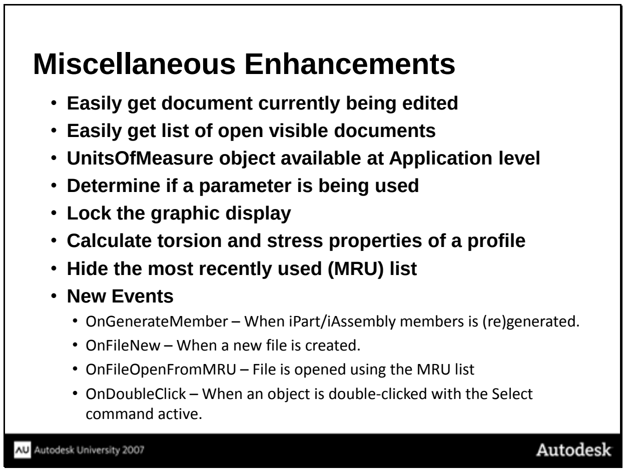


InteractionEvents Enhancements

- Copy Inventor's distance and Angle input
  - Area selection
  - Measure
  - Parameter selection
  - Show and select feature dimensions
- FileDialog enhancements
  - Multiple file selections
  - Can be intercepted by PDM clients

AU Autodesk University 2007 Autodesk

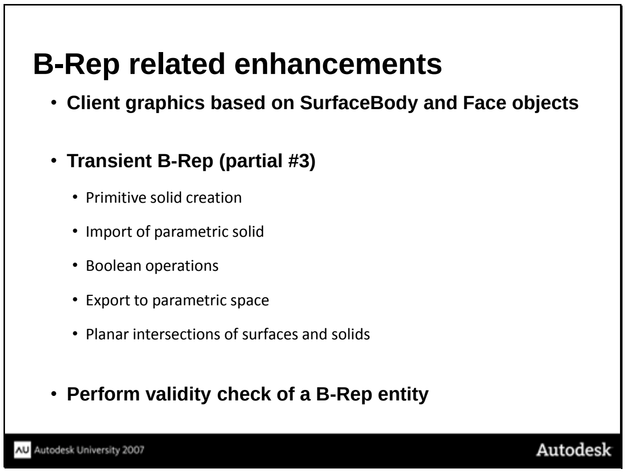
- InteractionEvents has been enhanced in several ways. You can now specify that you want to do area selection which allows the user to click and drag to select a group of entities. This is demonstrated in the **modMisc.TestWindowSelection** macro.
- There are also new options to allow you to perform measurements, show a list of parameters, and show and select feature dimensions. These capabilities allow you to fairly easily mimic the behavior of many Inventor dialogs as is shown on this slide to allow the user to specify the depth of the extrusion in one of several different ways. This is demonstrated in the **modMisc.InteractiveMeasureDistance**, **modMisc.InteractiveMeasureAngle**, and **modMisc.GetParameterInput** samples.
- The FileDialog object has been enhanced to allow the option of selecting multiple files. When the FileDialog is displayed it now fires the OnFileOpenDialog or OnFileSaveAsDialog events to allow a PDM application to override the dialog and provide their own interface to specify the file.



Miscellaneous Enhancements

- **Easily get document currently being edited**
- **Easily get list of open visible documents**
- **UnitsOfMeasure object available at Application level**
- **Determine if a parameter is being used**
- **Lock the graphic display**
- **Calculate torsion and stress properties of a profile**
- **Hide the most recently used (MRU) list**
- **New Events**
  - OnGenerateMember – When iPart/iAssembly members is (re)generated.
  - OnFileNew – When a new file is created.
  - OnFileOpenFromMRU – File is opened using the MRU list
  - OnDoubleClick – When an object is double-clicked with the Select command active.

- Get document currently being edited using the Application.ActiveEditDocument property.
- Get visible documents. Documents.VisibleDocuments
- Determine if a parameter is being used. Parameter.InUse
- Lock the graphics display. Application.ScreenUpdating
- Perform various 2D analysis on a profile. Profile.RegionProperties
- Turn off the MRU list. Application.MRUDisplay
- Respond to general double-click

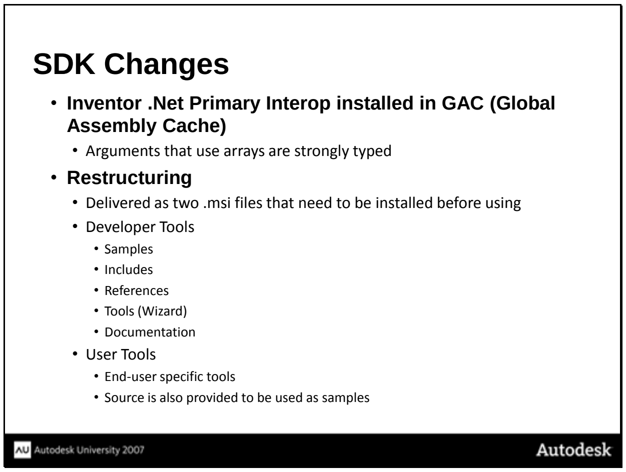


B-Rep related enhancements

- **Client graphics based on SurfaceBody and Face objects**
- **Transient B-Rep (partial #3)**
  - Primitive solid creation
  - Import of parametric solid
  - Boolean operations
  - Export to parametric space
  - Planar intersections of surfaces and solids
- **Perform validity check of a B-Rep entity**

Autodesk University 2007
Autodesk

- Surface based client graphics. This greatly simplifies the creation of client graphics for many cases. Instead of constructing triangles to approximate a shape you can now supply a Face or SurfaceBody object and have it drawn as client graphics.
- Transient B-Rep. This introduces the new concept of transient B-Rep objects. This area of the API provides the ability to do modeling operations outside the context of an Inventor document. Work performed in this transient area is not transacted, displayed, or saved. The results can be used to create standard features or displayed using client graphics. The functionality within this area is currently very limited but will be expanded over later releases of Inventor. It is accessed through the Application.TransientBRep property. **The samples modMisc.ClientGraphics3dPrimitives and modMisc.CreateBrep demonstrate this and the client graphics functionality talked about above.**
- Check a SurfaceBody object to see that it is valid. This can be useful for imported models. SurfaceBody.IsEntityValid

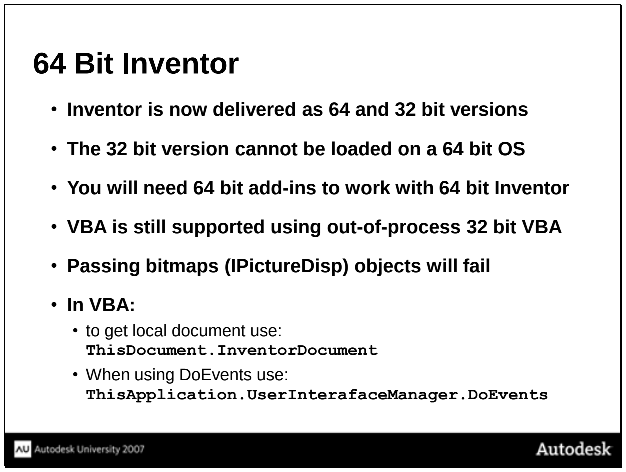


SDK Changes

- **Inventor .Net Primary Interop installed in GAC (Global Assembly Cache)**
  - Arguments that use arrays are strongly typed
- **Restructuring**
  - Delivered as two .msi files that need to be installed before using
  - Developer Tools
    - Samples
    - Includes
    - References
    - Tools (Wizard)
    - Documentation
  - User Tools
    - End-user specific tools
    - Source is also provided to be used as samples

Autodesk University 2007 Autodesk

- For better support of .Net programming we now deliver a primary interop assembly with Inventor. Programs written for Inventor 2009 and later should reference the primary interop. This primary interop is slightly different than the interop that was automatically being generated by Visual Studio when referencing the Inventor Object Library. The difference being that arguments of methods and properties that are arrays are now strongly typed (i.e. an array of Doubles) whereas before they were defined as the general System.Array class.
- The Inventor SDK is now being delivered as two .msi files that will need to be installed to access their contents. This was done to make last minute additions and changes to the samples but it also enables the ability to deliver updates of the SDK between releases, if we should choose to take advantage of that capability.



64 Bit Inventor

- **Inventor is now delivered as 64 and 32 bit versions**
- **The 32 bit version cannot be loaded on a 64 bit OS**
- **You will need 64 bit add-ins to work with 64 bit Inventor**
- **VBA is still supported using out-of-process 32 bit VBA**
- **Passing bitmaps (IPictureDisp) objects will fail**
- **In VBA:**
  - to get local document use: `ThisDocument.InventorDocument`
  - When using DoEvents use: `ThisApplication.UserInterafaceManager.DoEvents`

Autodesk University 2007 Autodesk