SolidWorks WORLD 2010

An exploration into the SolidWorks API

Leonard Kikstra
Product Designer / CAD Administrator / PDM Administrator
SolidWorks Certified Professional (CSWP)

SolidWorks WORLD 2010

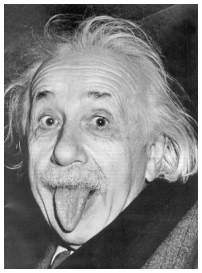
Who Am I?

- Product Designer
  - Engineering since 1982.
  - CAD user since 1991.
  - SolidWorks user since 1998.
- CAD Administrator
  - 20+ people on site.
  - Advise other sites.
- Productivity Gains - I want to:
  - Simplifying my job.
  - Make my computer do more work for me.
  - Get the most out of the tools I use to do my job.
- Certified SolidWorks Professional (CSWP)
- SolidWorks User Group
  - SMART (SolidWorks Milwaukee Area Resource Team)
  - Active member since 1999.
- Lenny's SolidWorks Resources
  - Online since Sept. 2003.
  - Free Macros, Tips & Tricks.
- LennyWorks Blog
  - Since 2007.
- Programming
  - Programmed over 40 SolidWorks macros.
  - Most of these are available for free on the Internet.

SolidWorks WORLD 2010

Quotation:

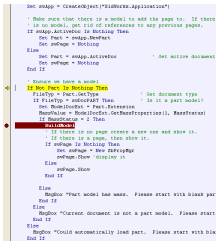
- Albert Einstein:
  - Computers are incredibly fast, accurate, and stupid.
  - Human beings are incredibly slow, inaccurate, and brilliant.
  - Together they are powerful beyond imagination.



SolidWorks WORLD 2010

What is covered here?

- EXPLORE
  - Macro Recorder.
  - Typical Macro Process.
  - Interaction With User.
  - SolidWorks API Object Model.
  - SolidWorks API Help file.
- EXPERIMENT
  - Try to duplicate SolidWorks functions or features.
  - Use macro recorder to record advanced options of common features.
- CREATE
  - Example Macros.
  - Tips.



SolidWorks WORLD 2010

EXPLORE: Macro Recorder

- How to record:
  - Plan your steps.
  - Launch the Macro Recorder.
  - Complete the steps.
  - Stop the Macro Recorder.
  - Save the macro.
- Play back recorded steps:
  - Run Macro.
  - Select macro.

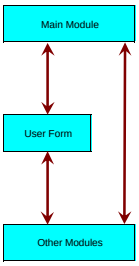
SolidWorks WORLD 2010

Macro Recorder Tips

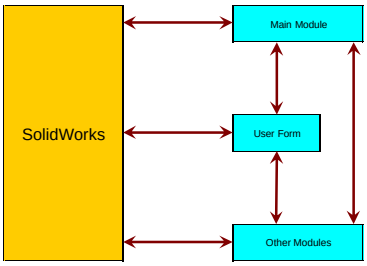
- Discovery Tool
  - Provides insight into how SolidWorks sees the documents.
  - Provides list of commands necessary to complete task as recorded.
- Programming Tool
  - Use many smaller recorded sessions to build code for larger applications.
    - Record a few steps at a time, then import into your own application.
    - Plan ahead – What do you want to record?

EXPLORE: Typical Macro Process

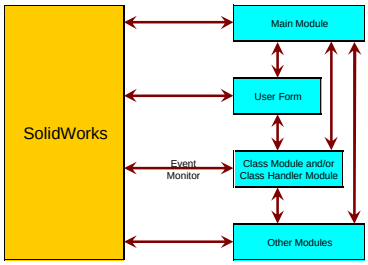
- Main Module
  - § Initialize Variables
  - § Attach to SolidWorks
  - § Attach to Document
- User Form
  - § Retrieve Data From Model
  - § Present Data To User
  - § Wait For User Input
  - § Process User Request
- Other Modules as needed
- Main Module
  - § Clean Up After Yourself
  - § Close Macro



EXPLORE: Typical Macro Process

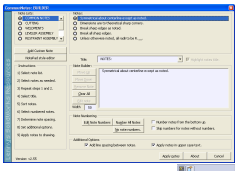


EXPLORE: Typical Macro Process with Event Monitoring

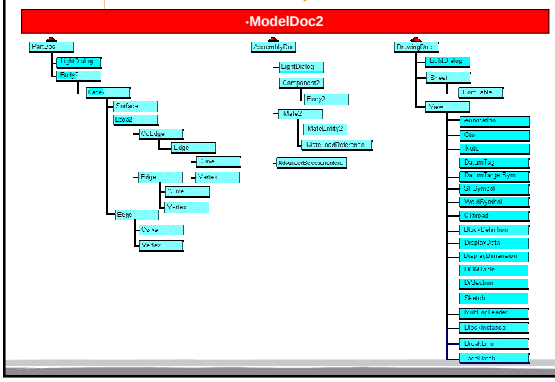


EXPLORE: Interaction With User

- Visual Basic Form Beginner
  - § User created with WYSIWYG utility.
  - § More flexible with more options.
  - § Easier to understand and program.
  - § Can display lots of information to user.
  - § Large form can get in the way.
- PropertyManagerPage Intermediate
  - § User Created.
  - § Manually setup page via programming (no form creation interface).
  - § Requires user programmed PageClass and PageHandlerClass.
  - § Requires “SolidWorks Exposed Type Library for Add-In use”.
  - § More complex to keep track when using multiple objects in form.
  - § If part of large form is hidden, user may not see all information.
  - § Out of the way.
  - § Looks and feels like it’s part of SolidWorks.
  - § See SolidWorks “VBA_PropertyManagerPage.swp” example.
  - § See “PMP” examples supplied with this presentation.



EXPLORE: SolidWorks Object Model



EXPLORE: SolidWorks API Help File

- Many Examples
- Accessing documents
  - § Open Document.
  - § Read existing data.
  - § Reload model.
- Selection Sets
- Sketches & Blocks
- Component States
  - § Resolved / Suppressed.
- Component Visibility
  - § Shown / Hidden.
- More Examples
- Traversal Routines
  - § Assembly Components.
  - § Part Features.
  - § Drawing Views.
  - § Feature Tree.
  - § Lighting.
  - § Reference Geometry.
  - § Mate Groups.
  - § Annotations.
- Add components
- Compare
  - § Documents.
  - § Features.
  - § Geometry.

EXPERIMENT: Discover and Learn

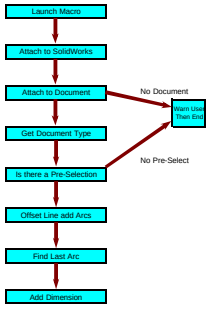
- Download examples from SolidWorks' website
- Download macros from user websites
- Step thru code to understand how the macro completes the task
  - [F5] and [F8] keys.
- Make changes and observe the effect of the changes

EXPERIMENT: Discover and Learn

- "Debug.Print"
  - Output data to "Immediates" window in macro editor.
- Macro recorder
  - Record tasks.
- Test "What If" scenarios
- Experiments
  - Do not need to complete a specific task
  - They are a learning and discovery tool
  - Save for later reference.
- Consistency enables predictability

CREATE: Quick & Dirty Macros - Sketches

- QuickSlot macro (as recorded)
  - Pre-select line in sketch.
  - Start recording macro.
  - Offset command.
    - No dimensions
    - Bi-Directional
    - Cap Ends
    - Arcs
  - Add dimension.
    - Endpoints of last arc created
  - Save macro.
- Challenges
  - Modify for flexibility as follows:
    - Work with any pre-selected line.
    - Dimension my way. (Full width)
    - Locate dimension appropriately.
  - Pre-existing entities in sketch.
- How it works:

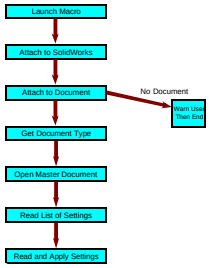


CREATE: Quick & Dirty Macros - Sketches

- QuickRectangle macro (as recorded)
  - Start recording macro.
  - Draw rectangle.
  - Draw diagonal line.
    - Opposite corners of rectangle
  - Constrain diagonal midpoint to origin.
  - Add dimensions.
- Challenges
  - Pre-existing entities in sketch.
  - Need list of last 4 points in sketch.
  - Need list of last 5 lines in sketch.
  - Locate 2 dimension appropriately.

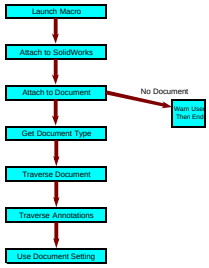
CREATE: Quick & Dirty Macros – Document Settings

- CopyDocSettings
  - Copies settings to existing document.
  - Only change document settings.
- Challenges
  - Master settings.
  - How to find master documents.
  - Work with multiple file types.
  - Separate document settings from SolidWorks settings.
- How it works:



CREATE: Quick & Dirty Macros – Annotation Settings

- AnnotationsPro
  - Change annotations to match document settings.
- Challenges
  - Accessing document settings.
  - Work with multiple file types.
  - Ignore the SheetFormat in drawing.
    - First view of each drawing sheet.
  - Incorporate special handling for view titles.
    - "Section" and "Detail"
  - Each annotation type requires different handling.
    - Notes, Dimensions, Weld Callouts, Etc…
- Future Enhancements:
  - Accommodate new annotation types.
- How it works:



CREATE: Quick & Dirty Macros - Models

- IsoViews
  - Creates alternate isometric views that can be referenced in SolidWorks drawings.
- Challenges
  - Predictably rotating model.
  - Where to start.
  - Working with Radians for model rotation.
  - Cannot create view if it already exists. Delete view before creation.
  - Works on models only.
- How it works:
  - Activate standard ISO view.
  - Rotate model about it's Y Axis.
  - Fit model to graphics window.
  - Delete named view from model.
  - Create a named view in model.
  - Repeat total of 4 times.
- Future Enhancements:
  - Diametric Views.
  - Bottom ISO Views.
  - Bottom Diametric Views.

CREATE: Quick & Dirty Macros - Drawings

- CenterlineSym
  - Create annotation at pre-selected point.
  - Use <MOD-CL> as note text.
  - Resize note to highlight in drawing.
- Challenges
  - Work with any pre-selected point.
- DelRevTriangle
  - Traverse drawing sheets.
  - Clear/Create selection set.
  - Traverse drawing views in sheet.
  - Find annotations with triangle border.
  - Add annotation to selection set.
  - Delete selection set.
- Challenges
  - Traversing drawing.
  - Finding proper annotation.
  - Adding to a selection set.
  - Selection set cleared when drawing sheet changed.

CREATE: Quick & Dirty Macros - Assemblies

- ToggleState
  - Toggle (suppress/unsuppress) state.
  - First version – For Mates Only.
  - New version – For Features, Mates and Components.
  - Works with multiple selections.
  - Suppressed and unsuppressed mates in same selection set.
    - Suppress unsuppressed objects
    - Unsuppress Suppressed objects
- Challenges
  - Evaluating objects in selection set.
    - How many items were pre-selected?
    - Features, mates and components only
    - What is suppressed or unsuppressed?
  - Build selection sets for later access.
  - Suppress first.
  - Unsuppress second.
- MateAlignToggle
  - Modify alignment setting for mates.
  - For mates only.
  - Works with multiple selections.
  - Based on example in SolidWorks Help File.
- Challenges
  - Evaluating objects in selection set.
    - Work with mates only
    - Can mate be flipped?
    - Special handling for each mate type.
  - Change each mate one at a time.

CREATE: Quick & Dirty Macros - Settings

- SolidWorksSettings
  - Faster access to settings I change frequently.
- Challenges
  - Limited to settings that can be toggles. (ON / OFF)
- WindowsExplorerPaths
  - Add or clear specific paths from SolidWorks list of search paths.
- Where Is This Useful?
  - Enable or disable searching PDM Vault for retrieving referenced documents.
- Challenges
  - Keep other paths.
  - Allow user to edit their preferred paths.

CREATE: Quick & Dirty Macros - 3D Sketches

- 3D Point Cloud
  - Read points from test file.
  - Plot points in 3D sketch.
- Challenges
  - Create new 3D sketch if not already in a sketch.
  - Read X, Y, Z coordinates from file
  - Create point.
  - Repeat.
  - Close sketch after all points created.

CREATE: Quick & Dirty Macros - Parts

- ChannelPMP
  - Need interface to modify parts with "Nominal" callouts.
  - Want to use SolidWorks' PropertyManagerPage interface
- Challenges
  - Storing data for Channel.
  - Understanding and implementing SolidWorks' PropertyManagerPage interface.
- Reference:
  - SolidWorks example: VBA_PropertyManagerPage.swp.

CREATE: Quick & Dirty Macros - Assemblies

- AssyComponentCopy
  - Copy referenced documents to current directory.
  - Useful for bringing documents into PDM Vault.
- Challenges
  - Traversing assembly.
  - Don't waste time.
    - Do need to copy files that already in same directory as assembly.
  - Exclude files in PDM Vault.
    - Ignore files in pre-defined directories.

- AssyComponentDensities
  - Retrieve list of density settings of components in assembly.
  - For weight verification.
- Challenges
  - Traversing assembly.
  - Access density setting of each component.
  - Display settings in list with model filename and description.
    - Export to text file

CREATE: Useful Examples - Other

- DataCharter
  - Change 2 "Driving" dimensions in sketch then retrieve and chart values of 1 "Driven" dimension.
  - Each "Driving" dimensions can have multiple values.
  - Permit exporting data to text file or Excel.
- Challenges
  - Allow user to select "Driving" and "Driven" dimensions.
  - Allow user to enter multiple values for 2 "Driving" dimensions.
  - Unit conversion
    - Angle and Linear dimensions have different conversion factors.

- DataRipper
  - Read dimensions on demand and chart values.
  - Keep interface on screen while user edits geometry. Do not react to user geometry editing.
  - Permit exporting data to text file or Excel.
- Challenges
  - Allow user to select dimensions to chart.
  - Keep interface on screen
  - Unit conversion
    - Angle and Linear dimensions have different conversion factors.

CREATE: Useful Examples - Other

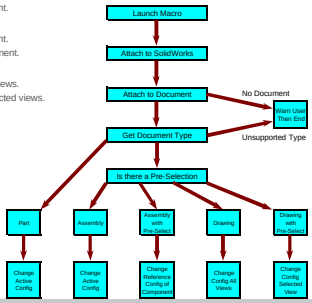
- RebuildAllConfigs
  - Ensure all configurations in model have been properly rebuilt.
  - Retrieve a list of configurations in the model.
  - Activate each configuration.
- RotateViewPro
  - Create multiple saved views in model.
  - Axis of rotation.
    - Model axis
    - Screen Axis
  - Degrees of rotation.

- DeleteConfigs
  - Parts and Assemblies only.

CREATE: Useful Examples - Multiple Document Types

- ConfigurationManager
  - Part
    - Configuration of current document.
  - Assembly
    - Configuration of current document.
    - Configuration of selected component.
  - Drawing
    - Referenced configuration in all views.
    - Referenced configuration in selected views.

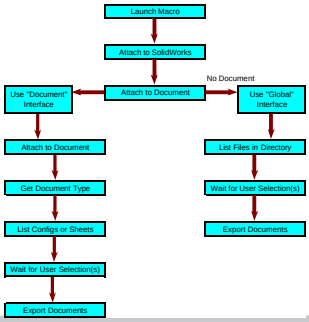
- How can this be done in a single macro? Plan Ahead!



CREATE: Useful Examples - Multiple Document Types

- DocExport
  - Drawings to DXF, PDF, etc…
  - Models to ACIS, STL, etc…
  - Different interfaces
    - With document open
    - Without document open (Batch Mode)
  - Time comparison / benefit
    - 8 hours to write.
    - Save 10-12 hours each use.
    - Saved 100's of hours.

- How it Works:



CREATE: Useful Examples - Other Stuff

- Custom Properties Editor
  - Type 1: Only edit your properties.
  - Type 2: Allows editing all properties.
  - Work with multiple document types.
  - Ensure your properties exist in. document.
- Global Property Editor
  - Rules based.
  - Edit properties in multiple documents.

CREATE: Interfaces

- CommonNotes
  - Interaction between objects on form
    - Selection in one field affects another field.
- C-Channel Selector
  - Using PropertyManagerPage

CREATE: SolidWorks Examples – FramedBox (Series)

- Start with recorded macro
  - Create sketch on front plane.
    - Add rectangle to sketch.
    - Add diagonal line.
    - Center diagonal on Origin.
    - Dimension sketch.
  - Extrude midplane.
  - Select top face.
    - Create Sketch.
    - Offset preset distance.
    - Cut thru all.
  - Repeat for 2 other visible faces.
  - Save macro.
- Add example interface to edit values.
  - SolidWorks' PropertyManagerPages
  - Before model creation.
- Challenges
  - Determine if part is active.
    - If not, start a new part
  - Temporarily disable automatic inferencing during sketching
    - Reduces unexpected constraints
  - Be aware of previous geometry created in model
    - Sequential feature naming
    - Sequential sketch naming
    - Dimension parameters include feature or sketch names
  - Different interface types
    - Visual Basic Form
    - SolidWorks PropertyManagerPages

CREATE: External Applications / Documents

- Launch other macros.
  - MacroLaunch
    - List and menu of available macros
    - Different list based on document type or no document open
  - MacroSequencer
    - Launch macros in specific order
      - Copy Settings From Master Document
      - Update Annotations to Document settings
- Launch applications.
  - Microsoft Word
  - Microsoft Excel
  - PDM Search Tool
  - Etc…
- Open Windows Explorer
  - Directories of last opened files
  - Directories of user specified projects
- Open web pages.
  - Local or Internet

CREATE: Code Re-Use

- Retrieving Model Data
  - DataCharter 12 hours
    - Modify values of 2 dimensions
    - Read / chart values of 1 dimension
  - DataRipper <2 hours
    - Allows user to modify dimensions
    - Reads / charts up to 5 dimension values upon request.
- Each macro you write can become a base for many more macros.

CREATE: Code Re-Use

- Globally Retrieve Data or Modify Documents
  - InfoFileBuild 8 hours
    - Retrieve / chart custom properties of all documents in current directory.
    - Opens models and reads data.
  - FixRevision 2 hours
    - Ensure "Revision" custom property exists and contains a value.
    - Opens models, checks current conditions, modifies model if needed
    - First in series to modify documents
  - FixAltPartNo 1/2 hour
    - Clear "Alternate part number on BOM" setting
    - Opens models, checks current conditions, modifies model if needed
  - AddDocumentNo 1/2 hour
    - Add "Document Number" custom property to all documents in current directory.
    - Opens models, checks current conditions, modifies model if needed
  - PropertyEditorGlobal 18 hours
    - ~30% of macro built with existing code from FixRevision macro
    - Remaining development time spent on new capabilities and interface.
- Each macro you write can become a base for many more macros.

Why won't some recorded macros work?

- NOTES:
  - The recorded macro will record the same objects and commands you used while recording the macro.
  - An unmodified recorded macro will only work with the objects you accessed during recording.
- While recording your macro, you may be modifying the document, so remember the following:
  - If the work is already done, there is nothing else for the macro to do.
  - When running a recorded macro that modifies documents, the SolidWorks documents must be in the same state, now, as they were when the macro was first recorded.

Why won't a macro work?

- Pre-selections:
  - Required pre-selections may not be done before running the macro.
  - Improper pre-selections may cause problems for the macro.
- Macro incorrectly started
  - Automatic starting point.
    - Last procedure of the last module inserted into the macro.
- Written for different version of SolidWorks
  - Incorrect Type Library References.
    - Version and macro specific.
  - Changes in the SolidWorks API.
    - Macro may be referencing obsolete API calls.

Why won't a macro work?

- SolidWorks settings
  - These settings may affect macro performance
    - Automatic Relationships
    - Snap to Points
    - Insert dimension value on creation
    - Automatic Solve / No Solve Move
    - Use Fully Defined Sketches
    - Lightweight components
    - Others…
  - How to check for these effects
    - Toggle various settings on/off
    - Re-test macro
    - Verify results are consistent
  - How to bypass these effects
    - Use variable to hold current setting
    - Change setting at beginning of macro
    - Reset setting (saved variable) after completion

Programming Strategies

- Start small and work your way up
  - Write and test small portions of code before continuing to larger portions.
  - Start with a "basic" program, then add more functionality and flexibility later.
    - Each feature added, requires more programming and increases macro complexity and potential for problems.
  - **KISS: Keep It Short and Simple.**

Programming Strategies

- Know that there are multiple ways to complete tasks
  - Conditional statements.
    - If … Then, Else, End If True/False
    - Select Case …., End Case Response based on value.
    - For …. Next Set number of times to repeat.
    - Do While …. End Loop Don't know how many times.
  - Multiple conditions may require nested statements.
  - The best method may not be the first one you develop or discover.

Programming Strategies

- Use subroutines and functions
  - Simplifies code.
  - Allows repeated use of common code.
  - Makes code easier to read / follow.
- Use existing code
  - Don't re-invent the wheel.
  - Saves time and your effort.
  - Keep sample code on hand for later use.
  - Export your own code for later use in other macros.
  - Create and use "Templates" for creating similar macros in the future.
  - Sample programs / macros.
    - Identify use of code borrowed.
    - Give credit where credit is due.

Programming Strategies

- Add comments to your code
  - Helps you and others understand what you did, and why.
  - Helps you remember what you did or why you did it that way.
  - Identify what the routines are doing.

Programming Strategies

- Use variable/constant/enumeration names that make sense.
  - Do not encrypt names
    - Use myCounter insteaf of myCnt.
  - Makes code more “readable”.
  - Keep your variable names unique and separate from Visual Basic and SolidWorks variable names.

    myCounter
    vbCounter
    swxCounter
  - Be careful that you don't use reserved keywords, commands or system variables as your own variable names

Programming Strategies

- Plan for multiple users or distribution
  - Think about other people who may be using your macros.
  - Macro may be used for needs other than initially intended.
  - User customization of existing macro.
    - Not every user will know how to modify a macro.
    - External settings file for modifiable default settings.
    - Macro defaults if settings file is not found.
  - Different companies have standards, procedures and needs.
  - Capable of running on different versions of SolidWorks

Programming Strategies

- Plan for use in multiple versions of SolidWorks
  - API can check what version of SolidWorks is in use.
- Error trapping.
  - Think of ways that users may (unintentionally) misuse your macro.
    - Wrong document type, Improper selections, etc…
  - Error trapping can be one of the toughest part of programming.

TIPS: Getting Help

- SolidWorks API Support.
  - SolidWorks API functionality only.
  - Not for learning.
- Be specific!
  - Repeatable problem.
    - Identify steps leading up to problem.
    - Identify specific conditions that lead to problem.
  - Identify specific problem you are experiencing.
  - Don't send complete macro.
    - Provide simple code to demonstrate.
- Don't expect help with programming techniques.

TIPS:

- EXPLORE
  - SolidWorks API Help file.
    - Hundreds of examples.
  - SolidWorks Website.
    - Many more examples.
    - Development Kits.
  - SolidWorks Forums.
  - Learn from existing examples.
    - Many available on internet.
    - 3D Content Central.
  - Reseller.
    - SolidWorks API Training.
  - User Websites
    - Many examples.
- EXPERIMENT
  - Macro recorder.
    - Record steps.
    - Great discovery tool.
  - Step thru macro.
    - Learn how the macro works.
  - Revise macro.
    - Observe what happens.
  - Duplicate existing functionality.
  - Learn from your experiments.
- CREATE
  - Create your own macros.
  - Use macro recorder.
    - Obtain pieces of code you need.
  - Create your own functionality.
    - How do you want it done?

Resources - Examples (Users)

- Lenny’s SolidWorks Resources — Macros, Tips, Blog
  - http://www.lennyworks.com/solidworks
- Mike Wilson’s Modeling Techniques — Macro Feature
  - http://www.mikejwilson.com/solidworks/solidworks_files.htm
- Matt Lombard — Macros, Tips, Blog
  - http://mysite.verizon.net/mjlombard/
- Joe Jones at New Hampshire CAD — Tutorials
  - http://www.nhcad.com/sw_macros/index.html
- Stefan Berlitz’s SolidWorks Tools — Macros, Tips, Blog
  - http://www.swtools.de
- SolidWorks Tips & Things — API Tips
  - http://www.solidworktips.com/


THANK YOU !!!
Good Luck with your project !!!
SolidWorks WORLD 2010
An exploration into the SolidWorks API
Leonard Kikstra
PDM Admin
Download this presentation and examples from:
http://www.lennyworks.com/solidworks