Introduction to ANSYS Workbench Scripting in ANSYS 12.1

November 2009

ANSYS[®]

Agenda



- Overview of ANSYS Workbench Scripting
- Recording & Replaying Journals
- Command Window
- Scripting Basics
- Project & Data Model Concepts
- Scripting with Data-Integrated Applications
- Where to Get Help
- Examples

Scripting Overview

ANSYS 12.1 fully supports Workbench journaling and scripting

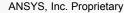
- Project concepts & operations
- Parameter management
- Native applications
 - Project Schematic, Design Exploration, Engineering Data
- File management and data models
- Python-based scripting language
 - Object-oriented
 - Platform-independent
- Fully documented & supported
- Works "hand-in-hand" with application-level scripting
 - DesignModeler, Meshing, Mechanical, Mechanical APDL, FLUENT, CFX, etc.











Journaling



- Workbench operations are recorded in a journal file
 - Only operations which modify data model
 - Files have *.wbjn extension
- Each session creates a new journal file
- Playing back the journal recreates the session
- Journals should not be confused with log files
 - Log files record specific Workbench events
 - Log files cannot be replayed

Journaling



Two types of Workbench journals

- Automatically recorded session journals
 - Restore work from a complete session
 - NOT designed to archive simulation projects
- Manually recorded journals
 - Starting points for creating custom Workbench scripts
 - Communicate sequential Workbench steps to colleagues or ANSYS Customer Support

Journaling Options (Preferences) (ANSYS)

Tools -> Options... -> Journals and Logs

| Project Management Appearance Regional and Language Options Graphics Interaction Geometry Import Journals and Logs Documentation Preferences Mechanical APDL CFX FLUENT Design Exploration | Journals and Logs Journal Files ✓ Record Journal File Journal File Directory C:\Documents and Settings\tmcdevit\Local Settin Days to Keep Journal File 10 ✓ When running a journal file, pause after each command Seconds to Pause 1 |
|--|--|
|--|--|

Manually Recorded Journals

1

New



| Start a | journal |
|---------|---------|
| record | ing |

Ctrl+N

Ctrl+O

Ctrl+S

| 2 | Open | Ctrl+O | | | |
|----------|---|---------------|---|---|--|
| | Save | Ctrl+S | | | |
| 3 | Save As | | | | |
| | Import | | | | |
| • | Archive | | | | |
| a | Restore Archive | | | | |
| _ | | | | | |
| - | Scripting | | • | ۲ | Record Journal |
| | Scripting Launch EKM Desktop | | • | • | Record Journal Run Script File |
| | | ent | • | ۲ | |
| <u>د</u> | Launch EKM Desktop | ent Ctrl+Q | • | • | Run Script File |
| <u></u> | Launch EKM Desktop Launch EKM File Transfer Clie | | • | • | Run Script File Open Command Window |

Stop a journal recording

Ctrl+N

| | Scripting | | × | ۲ | Stop Recording Journal |
|----------------|---|--------|---|---|--|
| ^ _} | Launch EKM Desktop… Launch EKM File Transfer Cli | ent | | | Run Script File Open Command Window |
| | Exit | Ctrl+Q | | | 1 C:\Users\tmcdevit\MyProjects\MyProject3.wbjn 2 C:\Users\tmcdevit\MyProjects\MyProject2.wbjn 3 C:\Users\tmcdevit\MyProjects\MyProject1.wbjn |

2

Ê

厚

R

New

Save

Open...

Save As...

Import...

Archive...

Restore Archive...

| Replay | ing J | ournals | | NSYS [®] |
|--|----------------------------|---|-------------------|------------------------------|
| interac | tive s | rnal at any ession ipting → Ru | | |
| New Open Save Save As Import Archive Restore Archive | Ctrl+N Ctrl+O Ctrl+S | | Look in: | |
| Scripting A Launch EKM Desktop Image: Launch EKM File Transfer Exit | | Record Journal Run Script File Open Command Window 1 C:\Users\tmcdevit\MyProject 2 C:\Users\tmcdevit\MyProject 3 C:\Users\tmcdevit\MyProject | s\MyProject2.wbjn | Pench Journal Files (".wbjn) |
| 2009 ANSYS, Inc. All rights reserve | d. | | 8 | ANSYS, Inc. Proprietary |

Replaying Journals



- A journal can also be replayed from the command line
 - Add –R <filename.wbjn>
 - Add –I for interactive mode or –B for batch

| Command Prompt | - 🗆 🗙 |
|---|-------|
| :\Users\tmcdevit\MyProjects>runwb2.exe -R MyProject.wbjn -I | |
| . Josels (Lincdevic (Hyriojects/Lunwb2.exe -K Hyrioject.wbjn -I | |
| | |
| | |
| | |
| | |
| | |
| | |
| | • |

Workbench Command Line Options



| Argument | Operation |
|--|--|
| -В | Run Workbench in batch mode. The user interface is not displayed and a console window is opened. The functionality of the console window is the same as the Workbench Command Window. |
| -R <ansys workbench<br="">script file></ansys> | Replay the specified Workbench script file on start-up. If specified in conjunction with –B, Workbench will start in batch mode, execute the specified script, and shut down at the completion of script execution. |
| -1 | Run Workbench in interactive mode. This is typically the default, but if specified in conjunction with –B, both the user interface and console window are opened. |
| -X | Run Workbench interactively and then exit upon completion of script execution. Typically used in conjunction with –R. |
| -F <ansys workbench<br="">project file></ansys> | Load the specified Workbench project file on start-up. |
| -E <command/> | Execute the specified Workbench scripting command on start-up. You can issue multiple commands, separated with semicolons (;), or specify this argument multiple times and the commands will be executed in order. |
| © 2009 ANSYS, Inc. All rights reserved. | 10 ANSYS, Inc. Proprietary |

Command Window



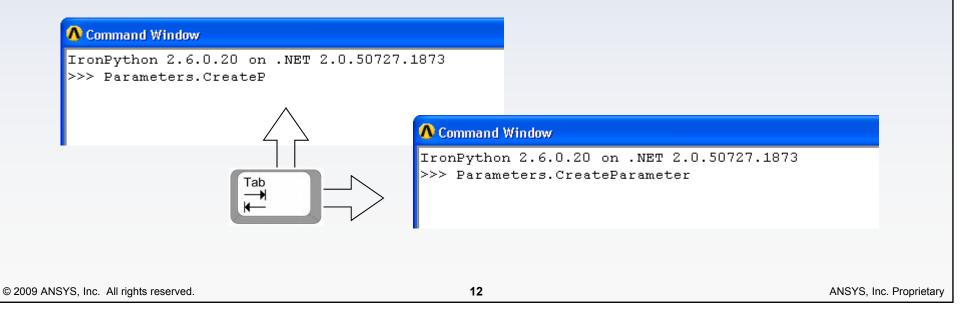
Scripting commands can be entered manually in Command Window

Command Window



Command Window supports:

- Command completion
- Command history
- Keyboard shortcuts for cursor navigation and editing



Scripting Basics



- Script: a set of instructions to be issued to Workbench
 - Can be a modified journal or a completely new set of instructions written directly
 - Scripts and journals use the same programming language—Python

Use scripts for:

- Automating repetitive tasks
- Performing Workbench operations in batch mode

Objects and Properties



 Object: combination of data and methods which act on the data

• Property:

- Data belonging to an object
- Defined by its name, type, and value
- Property types: Boolean, String, Integer, Real, etc.
- Dictionaries and Lists can also be property types

Properties are accessed via dot operator

- parameter1.Expression = 10
 - parameter1 is an object of type Parameter
 - Expression is a property of that object
 - Value of Expression is set to 10





Method: command that is bound to an object

- Can change property values, create or delete properties, or invoke complex calculations and operations
- Like properties, methods are accessed via dot operator
 - parameter1.SetQuantityUnits("m")
 - parameter1 is an object of type Parameter
 - SetQuantityUnits is a method called to set the object's units to meters

Object-Based Scripting Approach ANSYS

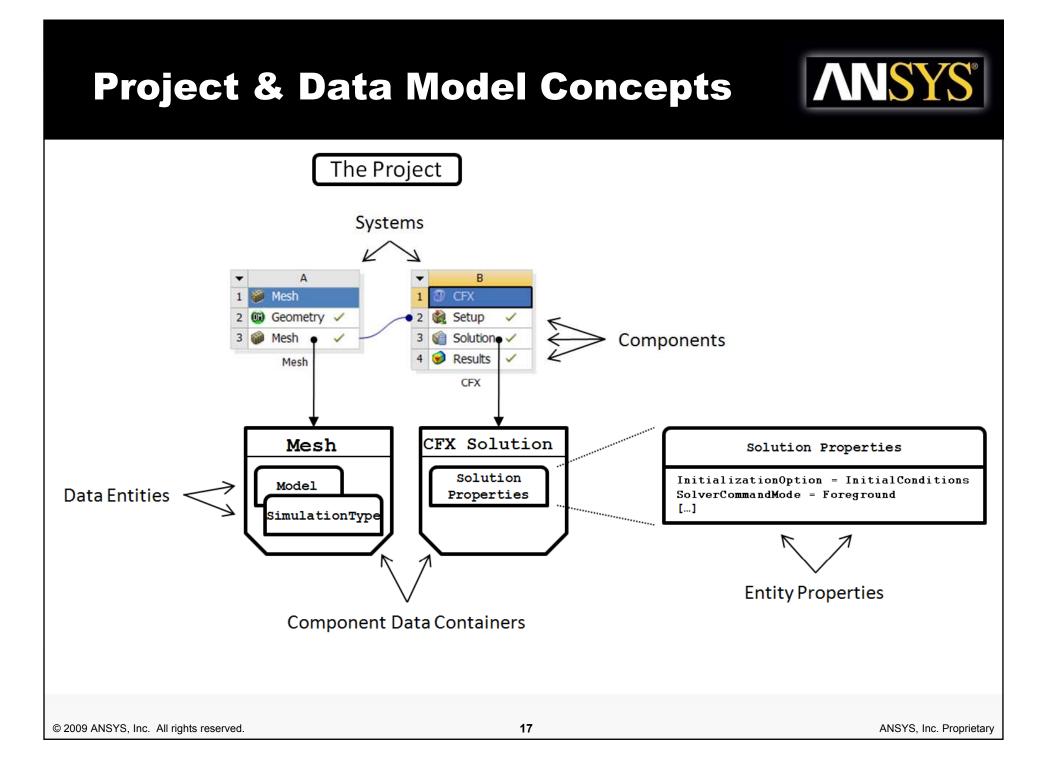
1. Query for an object

- Query methods return object references that are assigned to variables
 - Parameter = DOEModel.GetParameter(Name="P1")

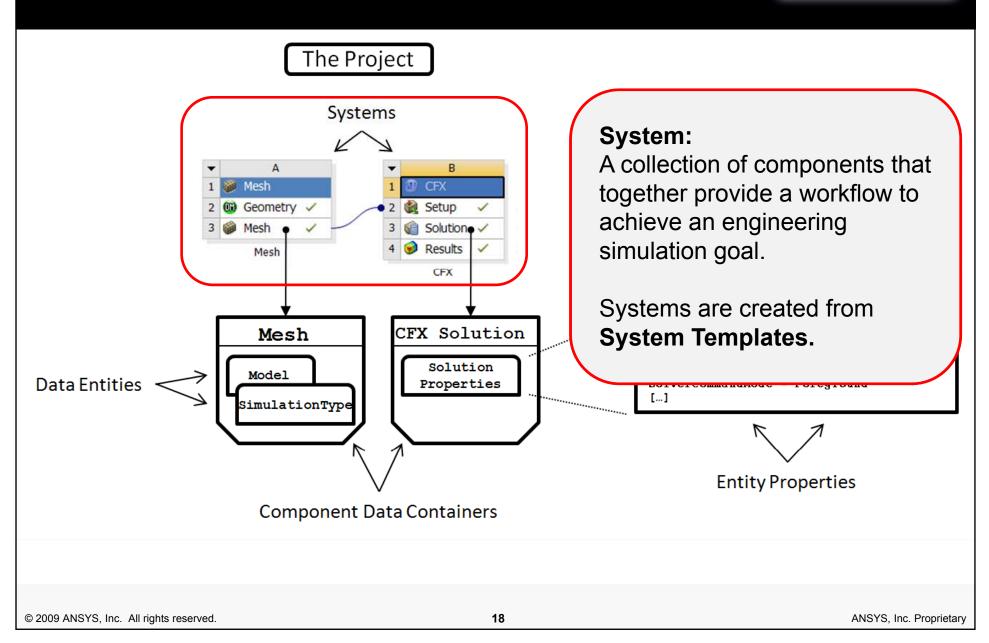
2. Interrogate/modify object properties

- Apply dot operator and property name to variable
 - Parameter.Nature = "NatureUsability"
- 3. Call methods to operate on object's internal data
 - Methods require comma-separated argument list within parentheses
 - Parameter.AddLevels(Levels=["65","70","75","80
 "])

Note: Python is a loosely-typed language; you do not need to declare variables.

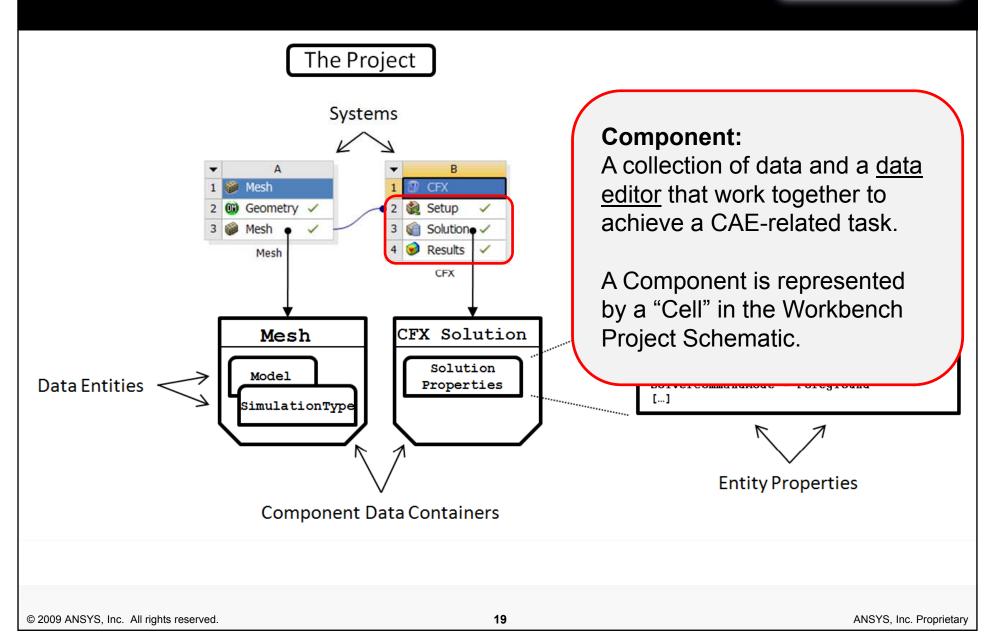


Project & Data Model Concepts



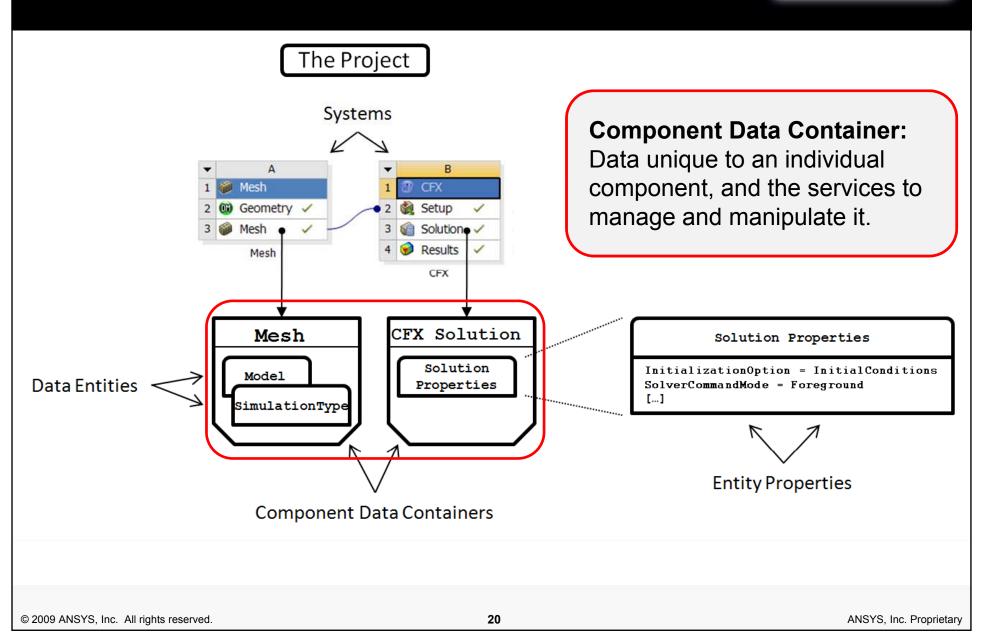
INSYS

Project & Data Model Concepts



INSYS

Project & Data Model Concepts



INNSYS®

ANSYS[®] **Project & Data Model Concepts** The Project Systems **Data Entity:** A data structure defined within a В A CFX data container. A data Mesh 1 2 00 Geometry 2 🍓 Setup container often has several data 3 Mesh Solution -3 entities. Results 4 Mesh CFX CFX Solution Mesh Solution Properties Solution InitializationOption = InitialConditions Model Data Entities Properties SolverCommandMode = Foreground [...] SimulationTvpe **Entity Properties Component Data Containers** 21 © 2009 ANSYS, Inc. All rights reserved. ANSYS, Inc. Proprietary

Example: Change a Material Property



Query for the static structural analysis template
ss_template = GetTemplate(TemplateName="Static Structural",
 Solver="ANSYS")

Create an analysis system from the template
ss system = ss template.CreateSystem()

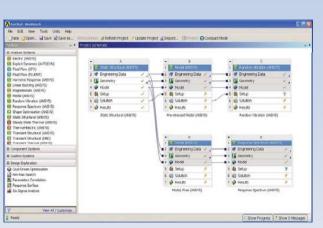
Query for the Engineering Data container ed_container = ss_system.GetContainer(ComponentName="Engineering Data")

Query for the material data entity in the data container steel = ed container.GetMaterial(Name="Structural Steel")

Query for the property data entity associated with structural steel elasticity = steel.GetProperty(Name="Elasticity")

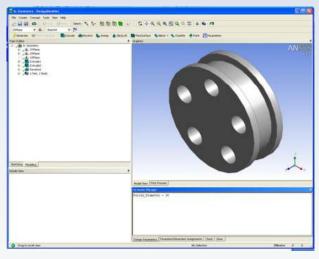
```
# Set Young's modulus
elasticity.SetData(Variables="Young's Modulus", Values=2E+11)
```

Data-Integrated Applications



Native applications

- Built entirely on new Workbench Framework
- Fully supported by Workbench scripting
- E.g., Project Schematic, Design Exploration, Engineering Data



Data-integrated applications

- Share data and parameters with Workbench, native applications, and other data-integrated applications
- Created independently from new Workbench Framework
- Often have their own scripting languages
- E.g., Mechanical, Mechanical APDL, CFX, FLUENT, DesignModeler

Data-Integrated Applications



 Application-level scripting can be embedded in a Workbench script

- APDL, CCL, Scheme, JScript, etc.
- Some data-integrated applications record their operations in the Workbench journal
 - CFX, FLUENT

Example: Create Geometry



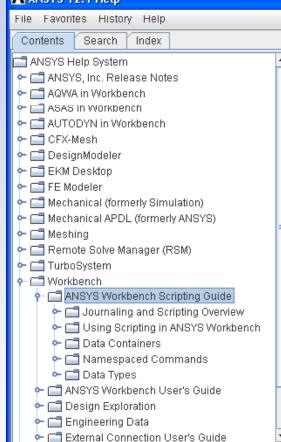
The following Workbench script sketches an elliptical curve in DesignModeler:

```
# create a geometry system
system = GetSystem(Name="Geom")
# query the data container of the geometry system
geometry = system.GetContainer(ComponentName="Geometry")
# send a JScript command to DM to create an ellipse
geometry.SendCommand(Command = """var ps1 = new Object();
  ps1.Plane = aqb.GetActivePlane();
 ps1.Origin = ps1.Plane.GetOrigin();
 ps1.XAxis = ps1.Plane.GetXAxis();
 ps1.YAxis = ps1.Plane.GetYAxis();
 ps1.Sk1 = ps1.Plane.NewSketch();
 ps1.Sk1.Name = "Sketch1";
  with (ps1.Sk1) { ps1.El7 = Ellipse( 8.0, 10.0, 9.0,
  6.0, 5.0, 12.0; }
  agb.Regen();""")
```

Documentation



ANSYS 12.1 Help



Online Help includes: Overview . • How-to Complete command reference Many examples b select help based on the product in which you are working. The sections Parameters tree can be expanded to display all related books and sections. Parameters This container hold project-level Parameters and Design Points Search: Select the Search tab to open the Search panel. You can search Methods Mechanical APDL (formerly ANSYS) command, section title, or regular GetParameter expression. Returns the set of all parameters associated with all entity properties in the given container. This method can be run on any container The set of parameters used by the given container Refurn Index: Select the Index tab to see a list of the products for which Help is Type DataReferenceSet available from this help viewer. Double-click a product name to review its Example In this example 'paramSet' becomes the set of parameters associated with the properties of any entity that resides within the Results container of system1. For detailed instructions on how to use ANSYS Help, select **Using Help** fr Help menu. Data Entities DesianPoint The data entity which describes a project-level design point Properties DisplayText The general property that defines the user-visible name of an entity. This property is defined for all Data Entities, but is used only in those entities that present a label in the User Interface Ŧ Type Read Only No equires an update

PDF version available on ANSYS Customer Portal

© 2009 ANSYS, Inc. All rights reserved.

ANSYS, Inc. Proprietary

277

Example & Demo: Parametric Update from MS Excel

| Paste V | A ≡ % | | - # | A+ Share As | | | | Image: A mage: A mage |
|--|---|--|--|--|------------------|--|----------|---|
| Clipboard 🕞 | | | Editing | Applicatio We | n Setting bEx | 32 | | clr.AddReference("Microsoft.04 import Microsoft.Office.Inter |
| F7 | • (0 | f _x | | | | | × | workingDir = "C:/Users/tmcdevi |
| Α Δ | В | С | | DI | E | F | - | def updateHandler(): |
| Bar Lenj 2 [cm] 3 | | laximum Defor [m] 4.1 | mation | Upda Workb | | | | <pre># Define key ranges in the lengthCell = worksheet.Rang loadCell = worksheet.Range defCell = worksheet.Range['</pre> |
| 4 rometerExample - Workber | nch | | | | | | | # Get the Workbench Paramet lengthParam = Parameters.Ge |
| | Save As 🗿 Import 🖓 Reconnect 🔒 | 😸 Refresh Project 🏓 Update Pro | yeat @Prover @Co | ampact Mode | | | | loadParam = Parameters.GetP defParam = Parameters.GetPa |
| ew Open Save X - X alysis Systems Indifice (ANSYS) kid Flow (CRX) kid Flow (CRX) kid Flow (CRX) | Save As Depend Schematic | ×. | жа (Флона Ф Са | ampact Mode | | | < Nordez | |
| ever Copen, Save A Save | Save A A A Project Schemalic A A 1 State: Structure of (MOLTIN) A 2 | | жа (С Рожа Ф Со | impact Mode | | - | | defParam = Parameters.GetPa # Assign values to the inpu lengthParam.Expression = 10 |
| New Copen Save & S - X nalysis Systemic Bechtic (ANSTS) Exploit Dynamics (ANSTS) Faid Film (CNS) Faid Film (CNS) Faid Film (CNS) Model (ANSTS) Model (ANSTS) Model (ANSTS) Model (ANSTS) Model (ANSTS) Shape Optimication (ANSTS) Shape Optimication (ANSTS) Shape Optimication (ANSTS) Shape Optimication (ANSTS) | Save A A A Project Schemalic A A 1 State: Structure / (Motivs) A 2 Trajecting Data A 3 Go Gometry A 4 Model A 5 Stup: School A 6 Solution A | × . × . × . | ₩a @rowa @co | mpact Mode | | | | defParam = Parameters.GetPa # Assign values to the inpu lengthParam.Expression = 10 loadParam.Expression = load # Mark the deformation para |
| Herr Connect Sector (ANSTS) Herr Connect (A | Seve A Import Period Project Schemade Import Period 1 State Sunctural (Sector) 1 2 Import Period 3 Import Import 4 Model Import 5 Import Import 6 Import Import 7 Import Import 8 Import Import 9 Import Import 1 Type A 2 ACO'S Revers. ACO'S Releases./ | Ansolt Academic Product Band | B Text Be for Next-Generation | | | 2 C Date(Time 1104/54 PM 1029/2009 | | <pre>defParam = Parameters.GetPa # Assign values to the inpu lengthParam.Expression = le loadParam.Expression = load # Mark the deformation para defCell.Value2="Updating # Run the project update</pre> |
| New Copen Save & S - X nalysis Systems & A Excite (AVSYS) Explicit Dynamics (AVSYS) Explicit Dynamics (AVSYS) Had Film (CFX) Flid Film (CFX) Flid Film (CFX) Modal (AXSYS) Modal (AXSYS) Modal (AXSYS) Modal (AXSYS) Modal (AXSYS) Satic Structural (AXSYS) Static Structural (AXSYS) Static Structural (AXSYS) Transiert St | Seve A Import Period Project Schemade Import Period 1 State Sunctural (Sector) 1 2 Import Period 3 Import Import 4 Model Import 5 Import Import 6 Import Import 7 Import Import 8 Import Import 9 Import Import 1 Type A 2 ACO'S Revers. ACO'S Releases./ | Ansoft Academic Product Band | B Test Ille for Nest-Generalion nies List | a Engineera. | | C 11/3/2009 1:04:54 PM 10/29/2009 2:04:23 PM 10/27/2009 2:04:23 PM | | <pre>defParam = Parameters.GetPa # Assign values to the input lengthParam.Expression = let loadParam.Expression = load # Mark the deformation para defCell.Value2="Updating # Run the project update Update() # Update the workbook value defCell.Value2 = defParam.W # Open the Workbench Project</pre> |
| New Conn Save A. S. Save A. Save | Sere A (2) Incort Pointermed (2) Project Schenold: | Ansoft Academic Product Bund Forbes 200 Best Small Compare is ANSYS Software Verifies Desi | B Text Ref Nort-Generation Inies List D | 2 Ensineers wative Retractable Roof | F | C Date/Time 11/3/2009 1:04:54 PM 10/29/2009 2:04:23 PM 10/27/2009 2:05:49 PM | 1 | <pre>defParam = Parameters.GetPa # Assign values to the input lengthParam.Expression = let loadParam.Expression = load # Mark the deformation para defCell.Value2="Updating # Run the project update Update() # Update the workbook value defCell.Value2 = defParam.</pre> |
| | Sere A (1) Inport Pointermed (1) Project Schender | Ansoft Academic Product Bund Forbes 200 Best Small Compare is ANSYS Software Verifies Desi Cell D * See * A3 013 | 8 Text Be for Next-Generation next List Ion of Wimbledon Innor D Type • Engineering Data File | 2 Ensineers wative Retractable Roof | F | C Data/Time 11/3/2009 110/35 PM 10/29/2009 2:04:23 PM 10/27/2009 2:05:49 PM 0.027/2009 2:05:49 PM 0.027/2009 | 1 | <pre>defParam = Parameters.GetPa # Assign values to the input lengthParam.Expression = let loadParam.Expression = load # Mark the deformation para defCell.Value2="Updating # Run the project update Update() # Update the workbook value defCell.Value2 = defParam.W # Open the Workbench Project</pre> |

ripting\Training) - GVIM elp 🖓 🔁 🔁 🃥 🏦 👕 🗿 💶 📍 🏝 Excel interop ice.Interop.Excel") .Excel as Excel /Work/Scripting/Training/ orkbook ["A3"] B3''1 3"] rs Parameter(Name="P1") rameter(Name="P2") ameter(Name="P3") parameters qthCell.Value2.ToString() ell.Value2.ToString() + " [N]" eter as updating in the workbook from the WB parameter lue arameterExample.wbpj") rkingDir + "ParameterExample.xlsx")

© 2009 ANSYS, Inc. All rights reserved.

Тор

6,56

ANSYS[®]

Example & Demo: Stress Analysis via APDL



| | | apdl.wbjn - Notepad File Edit Format View Help | |
|--|---|---|-----------|
| 🔨 myBar - Workbench | | # Import the 'os' module, which provides a portable way of using | ~ |
| File View Tools Units Help | | <pre># operating system dependent functionality ' import os</pre> | |
| 👔 New 🚰 Open 🛃 Save 🔣 Save As 👔 Import 🍣 Reconnect 湕 Refresh | Project 🥖 Update Project 🕜 Compact Mode | # Specify the Mechanical APDL Input file to be processed | |
| Toolbox _ X Project Schematic | _ × _ | inputFile = AbsUserPathName("Demo/ScriptExample4/bar.dat") | |
| 🗈 Analysis Systems | <u> </u> | <pre># Provide a list of bar length (Z) values to solve and write CDB files for. # Note: We expect '0' to fail.</pre> | |
| Electric (ANSYS) | | # Note: We expect 0 to fail. zvalues = [3,5,0,12,15] | |
| Explicit Dynamics (ANSYS) | | # Open a log file to record script progress | |
| C Fluid Flow-BlowMolding (P 1 Mechanical APDL | | logFile = open(AbsUserPathName("my_bar_script.log"),"w") | |
| Image: Second secon | | <pre># Start a new project and create the Mechanical APDL system Reset()</pre> | |
| G Fluid Flow (FLUENT) | = | <pre>template1 = GetTemplate(TemplateName="Mechanical APDL") system1 = template1.CreateSystem()</pre> | |
| G Fluid Flow (POLYFLOW) Mechanical APDL | | | |
| Narmonic Response (ANSYS | | <pre># Read the input file into the Mechanical APDL Setup setup1 = system1.GetContainer(ComponentName="Setup")</pre> | |
| Hydrodynamic Diffraction (# | | mapdlInputFile1 = setup1.AddInputFile(FilePath=inputFile) | |
| Linear Buckling (ANSYS) | | # Create Workbench parameters from two of the Mechanical APDL parameters # in the input file | |
| Magnetostatic (ANSYS) Modal (ANSYS) | | <pre>mapdlInputFile1.PublishMapdlParameter(Name="ZLEN") parameter1 = Parameters.GetParameter(Name="P1")</pre> | |
| View All / Customize | _ | mapdlInputFile1.PublishMapdlParameter(| |
| | × | Name="OUT_UY". | |
| Ø Double-click to view and edit all project parameters. | 💷 Show Progress 🥵 Show 19 Messages 🛒 | IsDirectOutput=True) parameter2 = Parameters.GetParameter(Name="P2") | |
| | | # Save the initial project definition. | |
| 🔿 myBar - Workbench | | Save(FilePath=AbsUserPathName("Demo/ScriptExample4/myBar.wbpj"), | |
| File View Tools Units Help | | Overwrite=True) | |
| New 🚰 Open 🚽 Save 🔍 Save As 👔 Import 🖓 Reconnect 🖉 Refresh Project 🦩 Update Project 🤇 | | # Loop through all provided bar lengths for zVal in zValues: | |
| Toolbox X Outline of Schematic A3: Parameters □ Parameter Charts ✓ A B C D | X Table of Design Points X ▼ A B C D E | # Set the Z (length) parameter expression | |
| Parameters Parallel Chart (all) 1 ID Parameter Name Value Uni | t 1 Name VP1-ZLEN VP2-OUT_UY VExported Note V | parameter1.Expression = str(zVal) | |
| M Parameters Chart 2 ⊡ Input Parameters M Parameters Chart P1 vs ? 2 □ Input Parameters | 2 Current 15 -0.12118 | logFile.write("Updating for z = %s∖n" % zVal) | |
| Design Points Vs ? | | # Update the project for the new parameter value, and report # success or failure to the log file. | |
| * \$ New input parameter New name New expression 5 Image: Output Parameters Image: Output Parameters Image: Output Parameters Image: Output Parameters | - | try: Update() | |
| 6 P2 OUT_UY -0.12118 | | except: logFile.write(" Update failed.\n") | |
| * New output parameter New expression | | else: logFile.write(" Update succeeded. UY = %s\n" % parameter2.Value) | |
| 8 Charts | Vio data | # Generate the name of the CDB file to save | |
| Properties of Outline A3: P1 | - X | cdbName = os.path.join(GetUserFilesDirectory(), "my_bar_" + str(zVal) + | ".cdb") |
| A B 1 Property Value | | cdbNameForCmd = cdbName.replace("\\","/") | |
| 2 | | # Delete the cdb file if it already exists, to prevent # Mechanical APDL from promting us about overwrite. | |
| 3 Description | | if os.path.exists(cdbName): os.remove(cdbName) | |
| 4 Error Message 5 Expression 15 | - | # Generate the APDL command to save the CDB file and send it. | |
| 6 Usage Direct Input | | apdlcmd = "cdwr, db,%s" % cdbNameForCmd setup1.Sendcommand(command=apdlcmd) | |
| View All / Customize 7 Quantity Name | | logFile.write(" CDB written to %s\n" % cdbName) | |
| Ready | 📼 Show Progress 🚇 Show 18 Messages 🛒 | # Save the final project state. | |
| | | Save() logFile.close() | ~ |
| © 2009 ANSYS, Inc. All rights reserved. | 28 | ANSYS, Inc. Pr | oprietary |
| a 2000 ANO 10, Inc. All rights reserved. | 20 | | ophetary |

Example & Demo: Plane Creation in CFD-Post



| 🔉 SaveJou - Workbench | |
|--|--|
| File View Tools Units Help | Ccl.wbjn - Notepad |
| 👘 New 🔀 Open 📕 Save 🕺 Save As 👔 Import 🏼 🍣 Reconnect 🛛 Refresh Project 🦩 Update Project 🤇 | |
| polbox _ X Project Schematic | - × # Create the Results system |
| Analysis Systems | <pre>template1 = GetTemplate(TemplateName="Results")</pre> |
| Electric (ANSYS) | <pre>system1 = template1.CreateSystem(Position="Default")</pre> |
| Explicit Dynamics (ANSYS) | # Edit the Results cell and load the Results file (StaticMixer_001.res) |
| G Fluid Flow-Blow Molding (P 🗐 1 💿 Results | results1 = system1.GetContainer(ComponentName="Results") |
| 🔋 Fluid Flow - Extrusion (POLY 2 🔗 Results 🗸 | results1.Edit() results1.Sendcommand(Command=r"""DATA READER: |
| Related Flow (CFX) | Clear All Objects = false |
| Fluid Flow (FLUENT) Results | Append Results = true Edit Case Names = false |
| Fluid Flow (POLYFLOW) | Open to Compare = false |
| Harmonic Response (ANSYS | Multi Configuration File Load Option = Separate Cases |
| Hydrodynamic Diffraction (# | Open in New View = true Keep Camera Position = true |
| Linear Buckling (ANSYS) Magnetostatic (ANSYS) | Load Particle Tracks = true |
| Modal (ANSYS) | Files to Compare = END |
| A 100G1 (All 5-15) | DATA READER: |
| © AZ : Kesults - CFD-Post File Edit Session Insert Tools Help | Domains to Load= |
| °°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°° | END > load filename=C:/users/sdg/ScriptingDemos/StaticMixer_001.res, multifile=append' |
| Outline Variables Expressions Calculators Turbo | # Set the camera and define a plane colored with a constant color |
| B (a) Cases ^ View 1 + | # Set the camera and define a plane colored with a constant color results1.Sendcommand(Command="""VIEW:View 1 |
| e 🖬 Default Domain Pressure | Camera Mode = User Specified |
| Plane 1 | CAMERA: Option = Pivot Point and Quaternion |
| -]]‡ in2 | Pivot Point = 0, 0, 0 |
| Contemporary | Scale = 0.226146 Pan = 0, 0 |
| © Diser Locations and Plots | Rotation Quaternion = 0.279848, -0.364705, -0.115917, 0.880476 |
| Stobel Cool | Send To Viewer = False |
| | |
| - 5.780e+003 | > autolegend plot=/PLANE:Plane 1. view=VIEW:View 1""") |
| | results1.Sendcommand(Command="""PLANE:Plane 1 Apply Instancing Transform = On |
| - 2.160e+003 | Apply Texture = Off |
| | Blend Texture = On |
| | Bound Radius = 0.5 [m] Colour = 0.75, 0.75, 0.75 |
| -1.460e+003 | Colour Map - Default Colour Map |
| [Pa] | Colour Mode = Variable Colour Scale = Linear |
| | Colour Variable = Pressure |
| | Option = Point and Normal |
| | # (Lines omitted for brevity) |
| | # |
| | END"") results1.SendCommand(Command="""# Sending visibility action from ViewUtilities |
| | >show /PLANE:Plane 1, view=/VIEw:view 1""") |
| 0 2,000 (| m (m) # Save the project |
| | save [i] each = r"SaveJou.wbpj", Overwrite=True) |
| 1,000 | |
| 30 Viewer Table Viewer Chart Viewer Comment Viewer Report Viewer | |
| | |
| 2009 ANSYS, Inc. All rights reserved. | 29 ANSYS, Inc. Proprieta |
| ., | |