# CP9844-L: Moldflow Synergy API Training
# Part 1: An Introduction

**David Astbury** Senior Manager, Solver Development

**Matt Jaworski** Senior Moldflow Technical Specialist

Version 3.1

# Class summary

In this class we will:

- Show how to build practical API scripts using the API in Moldflow Insight
- Introduce the API functionality and conventions
- Automate MFR creation, create custom plots, interface result with other programs (Excel & PowerPoint)
- Present a mixture of lectures and hands-on exercises to ensure that we cover the theoretical and practical aspects of building API scripts

Some programming knowledge is preferred, but not required

# Key learning objectives

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

- Leave with all the required knowledge to build usable API scripts using Autodesk Moldflow Synergy
- Understand the concepts, conventions and principles behind the API
- Show how to do common tasks with the API
- Show how the API can be used to export data

# Biography – David Astbury

David Astbury

- Started developing  Moldflow Software in 1988
- B App. Sci. (Chemistry) University of Melbourne
- Roles: Developer, R&D Manager, QA Manager, Senior Manager
- Expertise:  Numerical Simulation, Optimization/Design of Experiment (DOE), Project Management, QA Management, Moldflow API, IT/Linux, Software Development, Automation

# Biography – Matt Jaworski

Matthew Jaworski

- Used Moldflow since 1996, joined Moldflow in 2000
- Roles: Support, Training, Application Engineer, Manager
- Erie Plastics, Hewlett Packard, Rubbermaid previously
- Taught at UMass Lowell & Penn State Erie
- Dual BS Degrees (Mechanical Engineering & Plastics Engineering Technology – Penn State University)
- MS Plastics Engineering from UMass Lowell
- Finishing PhD in Plastics Engineering from UMass Lowell

# Assisting Today

- Dr. Franco Costa
- Dr. Nanda Santhanam

- Assisted in preparation of material

  Caroline Dorin Hanno van Raalte

  Shishir Ray

# Related Sessions

- CP12451-L: The Autodesk Moldflow Synergy API part 2: Building real world applications
  - Monday 30th November 1:00 pm – 5:00 pm
  - San Polo  3503, Level 3

- CP10731-L: Learn to Create Professional, Stunning Reports with MS Office and the Synergy API
  - Tuesday 2nd December 1:00 pm – 2:15 pm
  - San Polo  3503, Level 3

# House Keeping

# Timetable

| Time | Activity |
|---|---|
| 8:00 - 8:15 | Introductions |
| 8:15 – 8:45 | API Basics |
| 8:45 - 9:15 | Exercise 1:  MFR Export Automation |
| 9:15 - 9:40 | API Fundamentals |
| 9:40 - 10:00 | Break |
| 10:00 - 10:30 | Exercise 2: Custom Aspect Ratio Plot |
| 10:30 - 11:00 | Exercise 3: Export Results to Excel |
| 11:00 - 11:30 | Exercise 4: Export Results to PPT |
| 11:30 - 11:45 | Useful scripts |
| 11:45 - 12:00 | Discussion / Questions |

# Demographics

- Are you a Moldflow user?
- Do you have any programming experience?
- Are you taking this afternoon’s class?
- What do people want to achieve today?

# Standing Orders

- Ask questions at any time
  - We may refer it to a break if it’s complex/lengthy
- Have Fun
  - This is Las Vegas after all
- After the class
  - Feel free to contact us to discuss your requirements/projects
    - A little advice/direction from us may save you a lot of time

# Exercise Standing Orders

- Assist each other with the exercises
  - Feel free to exchange code/ideas
  - If you are stuck, look at the solution
  - Complexity increases with each task in the exercises
  - If you finish early
    - Help others
    - Experiment with what you have learned
    - Compare your approach with others. Often there are many solutions.
  - If your programming skills are poor then pair up with somebody who has done some programming before!

# API

# What is the Autodesk Moldflow Insight API?

- API = **A**pplication **P**rogramming **I**nterface
- An Object Linking and Embedding (OLE) programming interface that allows AMI functionality to be automated
- Manipulation of AMI is done through scripts or third party software
- Functionality available since MPI 4.0 (2002)
- Additional functionality added in each major Insight release

# Why use the AMI API?

- Increase user productivity for repetitive tasks
- Customize the UI, result plots & solvers
- Support 3rd party add-on products
- Enhance integration to
  - ERP
  - Enterprise-wide applications
  - CAD/CAE
  - MS Office Excel/Word/PPT
- Support university research programs
- Standardize corporate protocols and best practices
- Support industry data formats

# Quick API Example Demonstration

# OLE Introduction

- An OLE automation client is needed to control the API though AMI. Examples include:
  - Visual Basic Script or vbScript (VBS)
    - AMI records all scripts in Visual Basic Script
  - Other Potential Interfaces
    - JScript and other programming languages
    - Visual Basic for Applications (VBA)
      - This is a fully featured client that is part of the Microsoft Office application suite
    - Visual Basic (VB)
    - Perl
    - Python
    - ActiveX scripts in Internet Explorer
  - Autodesk only provides help/support for VBS

# OLE Introduction

- The OLE automation interface will automatically use the version of Synergy (AMI's GUI) that was most recently executed on your computer

- If you last opened an early version of AMI/MPI and try to run a macro or script that uses features that are not supported, your macro or script will generate an error

# Running multiple Synergy Instances

- Prior to  the 2016 Release it was only possible to run 1 API instance across all Synergy Instances

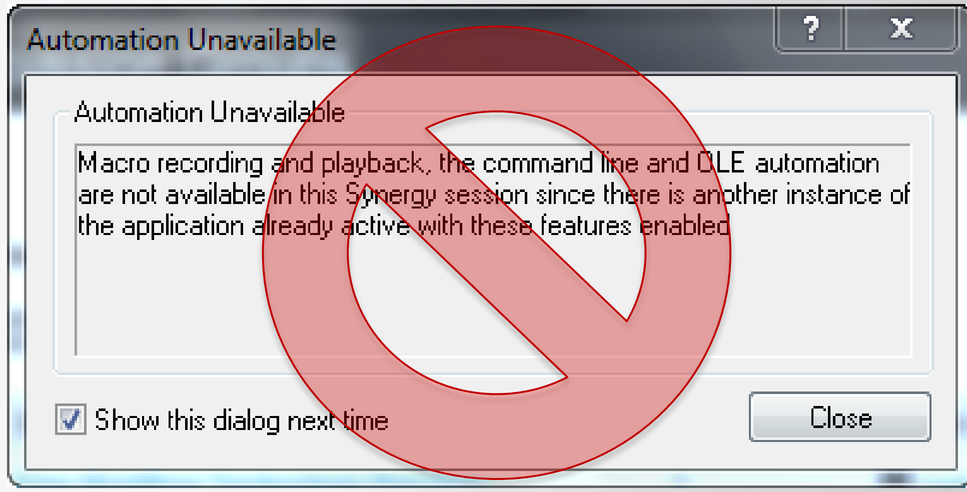
- API now works with multiple Synergy instances
  - You can launch a macro for a specific instance of Synergy on the local machine, using the ***InstanceID*** that you can find in the **About** box

# Running multiple Synergy Instances

- Edit Macros recorded/written prior to the 2016
  - Add the following code segment

```
'%RunPerInstance
```

Must be first line to avoid this message



Automation Unavailable

Automation Unavailable

Macro recording and playback, the command line and OLE automation are not available in this Synergy session since there is another instance of the application already active with these features enabled

☑ Show this dialog next time

```
Dim SynergyGetter, Synergy
On Error Resume Next
Set SynergyGetter = GetObject(CreateObject("WScript.Shell").ExpandEnvironmentStrings("%SAInstance%"))
On Error GoTo 0
If (Not IsEmpty(SynergyGetter)) Then
  Set Synergy = SynergyGetter.GetSASynergy
Else
  Set Synergy = CreateObject("Synergy.Synergy")
End If
```

Code to Ensure Correct Instance is executed

# Synergy Command Line

- Execute a script from the command line
  - Available since the 2016 release
  - Allows synergy.exe to be run from scripts or external processes
  - The synergy UI window will always start
    - No way to stop this
  - Synergy will exit when the script completes
  - Examples

```
synergy.exe /script c:\myscript.vbs
synergy.exe /silent /script c:\myscriptwitharguments.vbs 10 “fred”
```

# Synergy Command Line

- Syntax

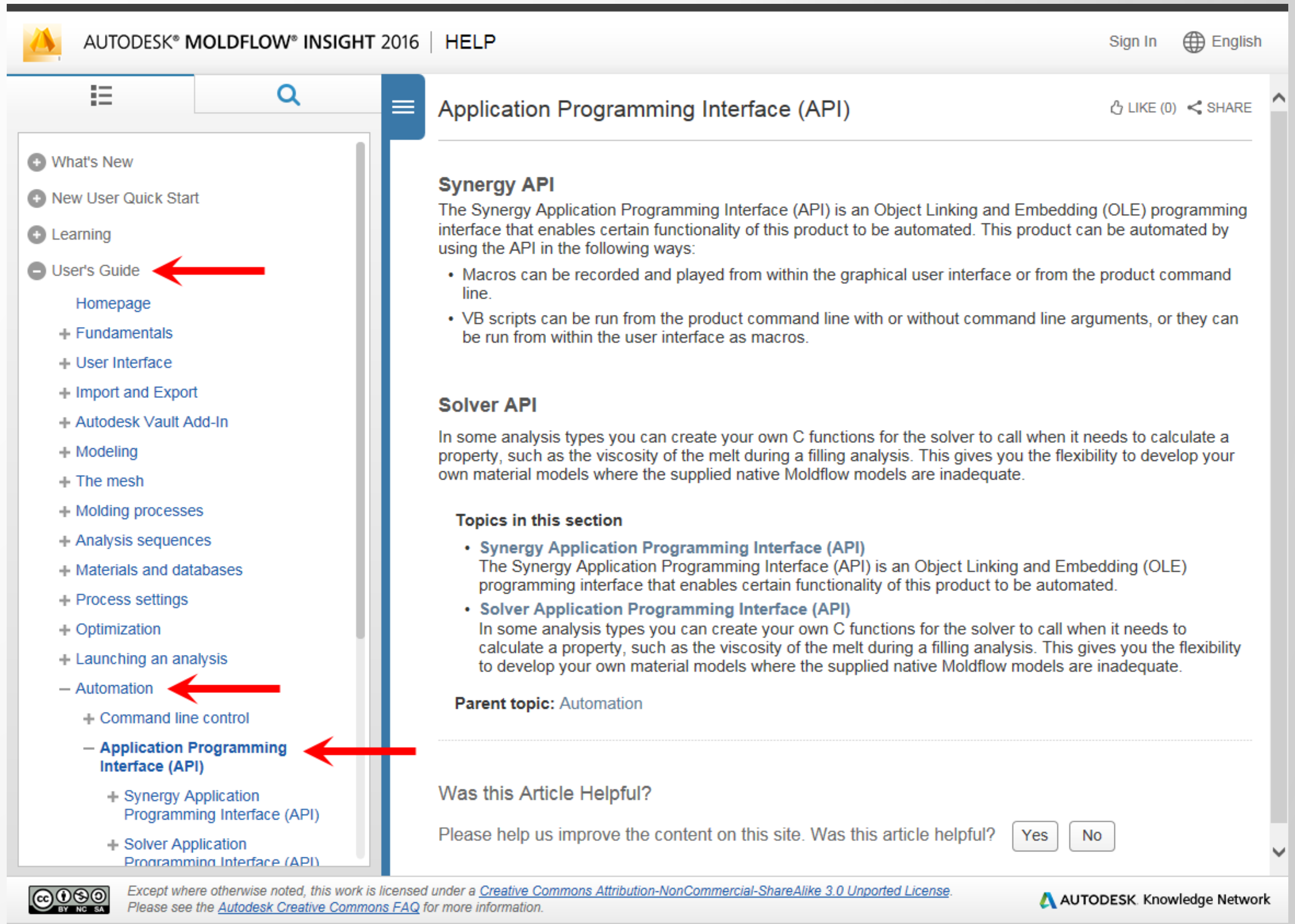
  /script SCRIPT [ARGS] : Running script in Synergy's instance

  /silent : Run the script without showing any VB errors or message boxes

  /noshutdown : Do not shutdown Synergy after running the script
  if /noshutdown is not sent, then Synergy will return an error code:
  0 = no errors were reported
  1 = could not execute script
  2 = script may not support Per-Instance execution
  any other = error returned by the script itself (by calling Wscript.Quit)

# API On-line Help

- Valuable resource for the beginner or expert
- Accessed via
  - Help > User’s Guide > Automation > Application Programming Interface (API)
  - Help > Advice > Application Programming Interface (API) under the Help home page



## Application Programming Interface (API)

### Synergy API

The Synergy Application Programming Interface (API) is an Object Linking and Embedding (OLE) programming interface that enables certain functionality of this product to be automated. This product can be automated by using the API in the following ways:

- Macros can be recorded and played from within the graphical user interface or from the product command line.
- VB scripts can be run from the product command line with or without command line arguments, or they can be run from within the user interface as macros.

### Solver API

In some analysis types you can create your own C functions for the solver to call when it needs to calculate a property, such as the viscosity of the melt during a filling analysis. This gives you the flexibility to develop your own material models where the supplied native Moldflow models are inadequate.

**Topics in this section**

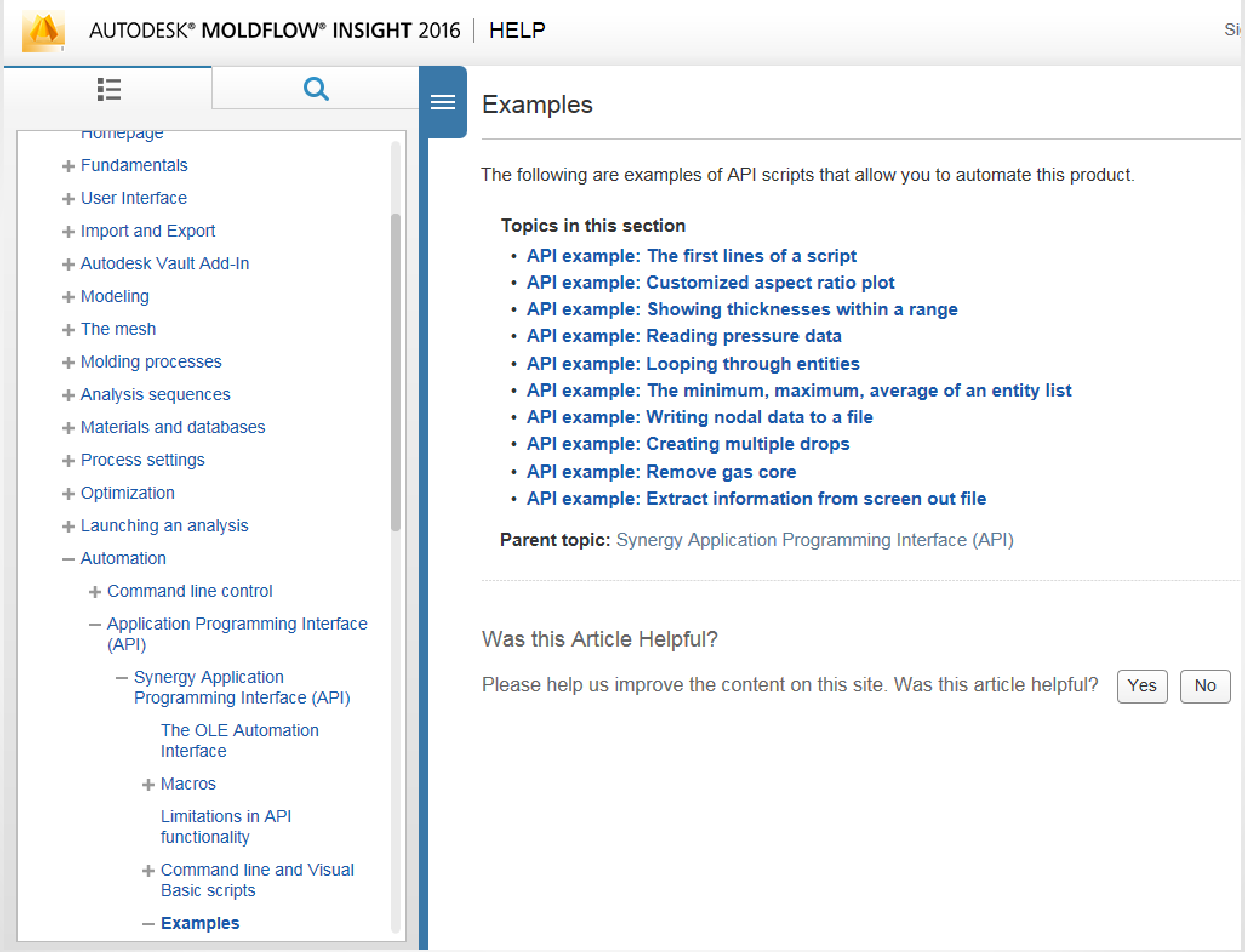
- **Synergy Application Programming Interface (API)**
  The Synergy Application Programming Interface (API) is an Object Linking and Embedding (OLE) programming interface that enables certain functionality of this product to be automated.
- **Solver Application Programming Interface (API)**
  In some analysis types you can create your own C functions for the solver to call when it needs to calculate a property, such as the viscosity of the melt during a filling analysis. This gives you the flexibility to develop your own material models where the supplied native Moldflow models are inadequate.

**Parent topic:** Automation

Was this Article Helpful?

Please help us improve the content on this site. Was this article helpful?

# Example Help VB Scripts



Homepage
+ Fundamentals
+ User Interface
+ Import and Export
+ Autodesk Vault Add-In
+ Modeling
+ The mesh
+ Molding processes
+ Analysis sequences
+ Materials and databases
+ Process settings
+ Optimization
+ Launching an analysis
– Automation
  + Command line control
  – Application Programming Interface (API)
    – Synergy Application Programming Interface (API)
      The OLE Automation Interface
      + Macros
      Limitations in API functionality
      + Command line and Visual Basic scripts
      – **Examples**

## Examples

The following are examples of API scripts that allow you to automate this product.

**Topics in this section**

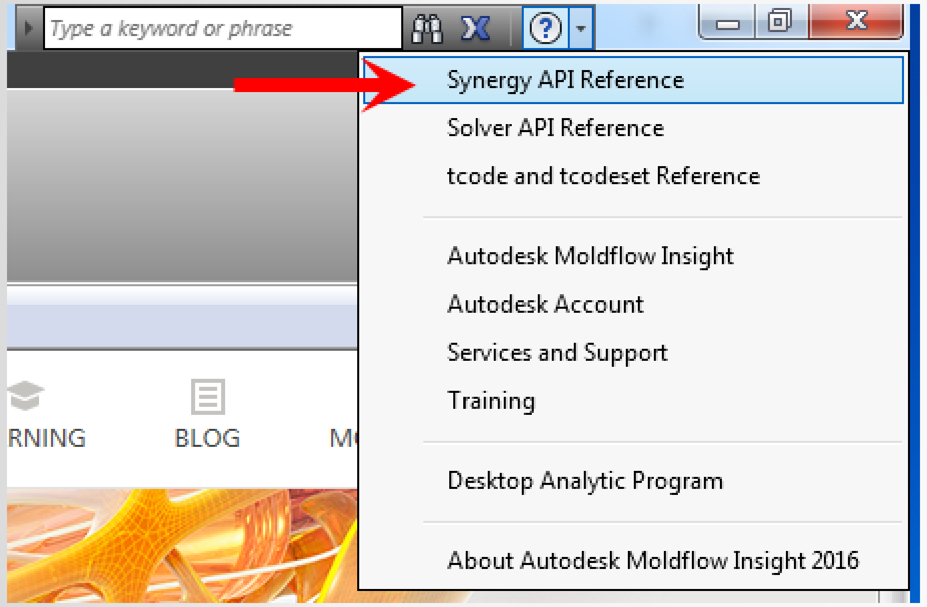
- **API example: The first lines of a script**
- **API example: Customized aspect ratio plot**
- **API example: Showing thicknesses within a range**
- **API example: Reading pressure data**
- **API example: Looping through entities**
- **API example: The minimum, maximum, average of an entity list**
- **API example: Writing nodal data to a file**
- **API example: Creating multiple drops**
- **API example: Remove gas core**
- **API example: Extract information from screen out file**

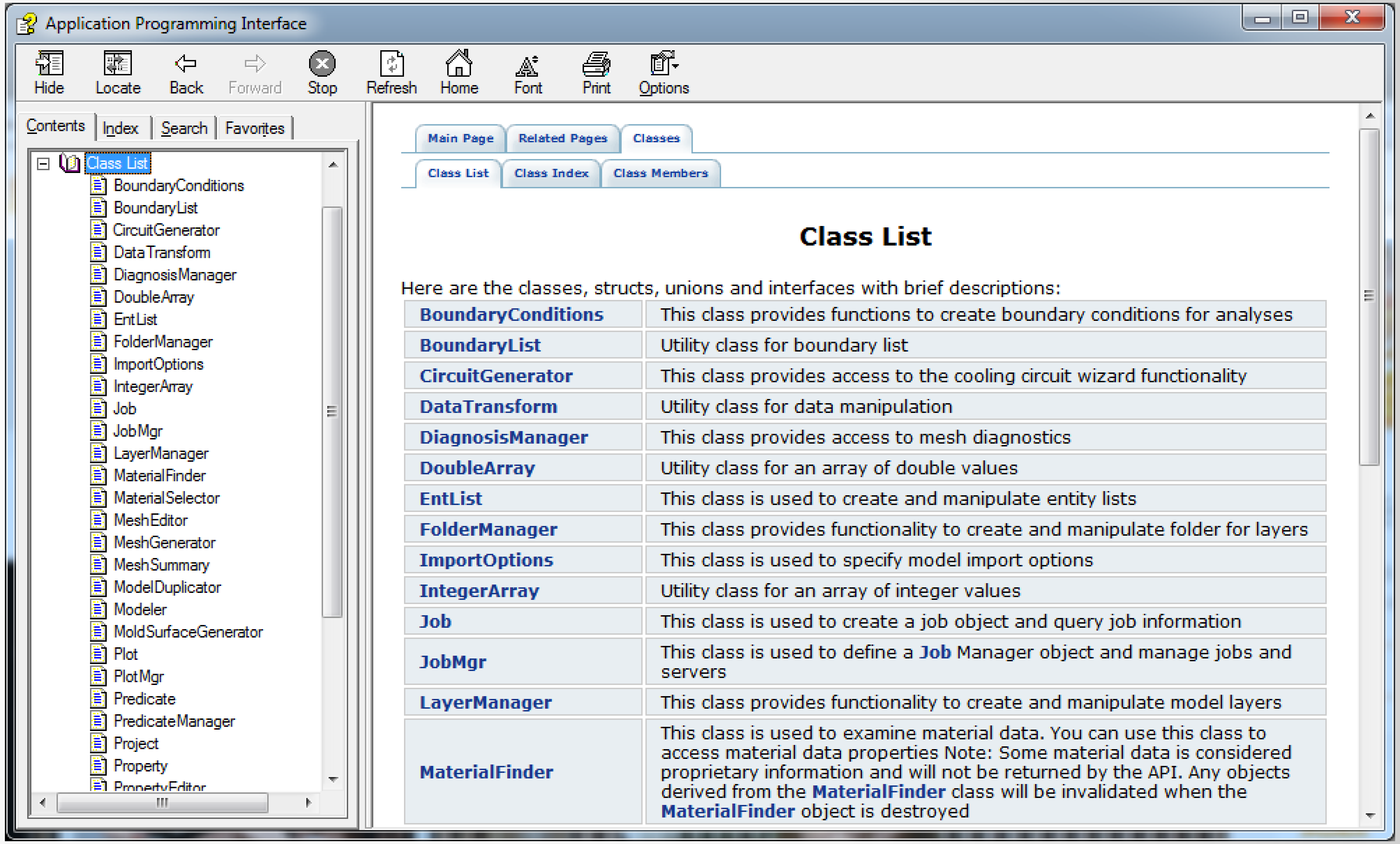
**Parent topic:** Synergy Application Programming Interface (API)

Was this Article Helpful?

Please help us improve the content on this site. Was this article helpful? Yes No

# API Reference Manual



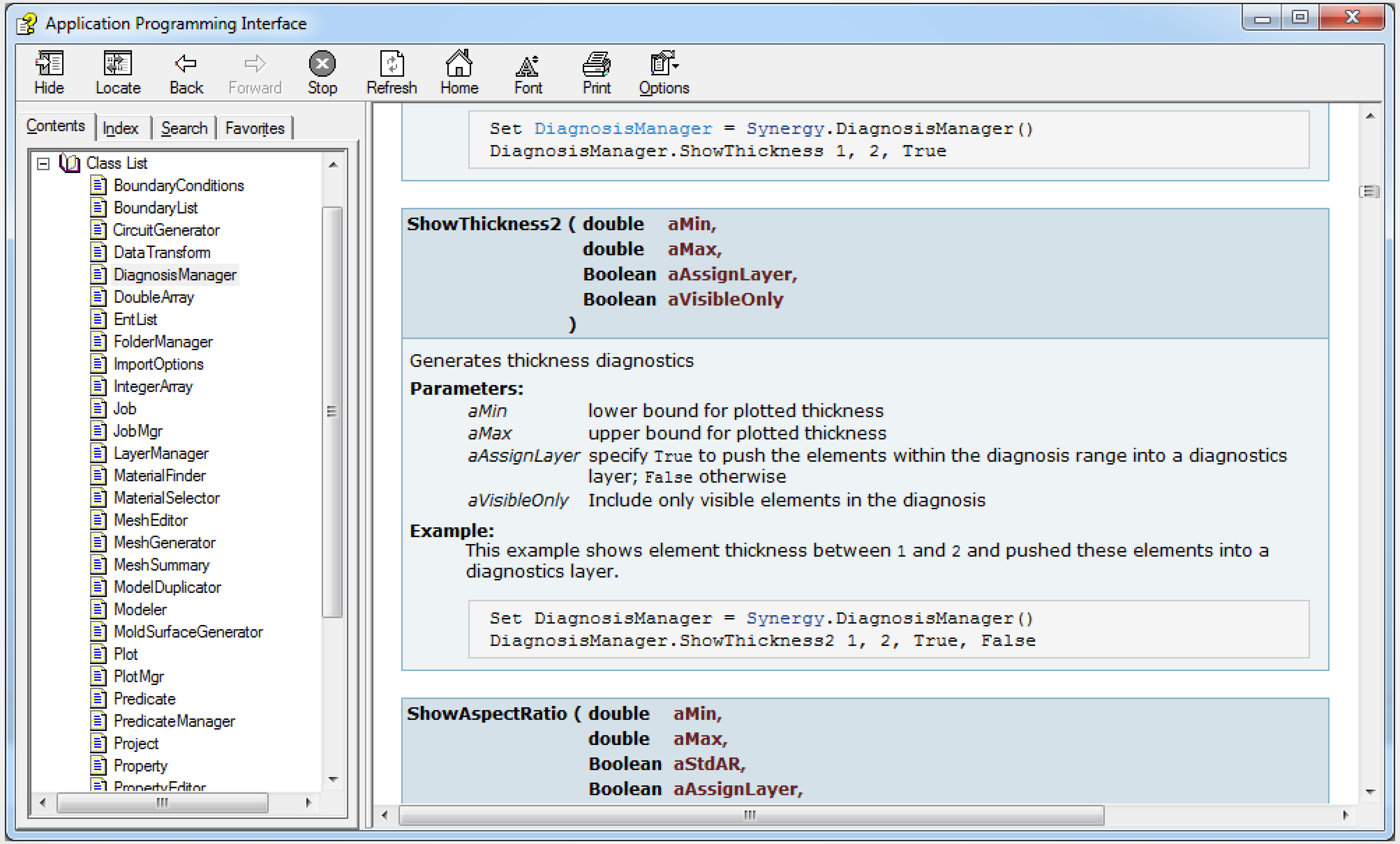


## Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

| | |
|---|---|
| **BoundaryConditions** | This class provides functions to create boundary conditions for analyses |
| **BoundaryList** | Utility class for boundary list |
| **CircuitGenerator** | This class provides access to the cooling circuit wizard functionality |
| **DataTransform** | Utility class for data manipulation |
| **DiagnosisManager** | This class provides access to mesh diagnostics |
| **DoubleArray** | Utility class for an array of double values |
| **EntList** | This class is used to create and manipulate entity lists |
| **FolderManager** | This class provides functionality to create and manipulate folder for layers |
| **ImportOptions** | This class is used to specify model import options |
| **IntegerArray** | Utility class for an array of integer values |
| **Job** | This class is used to create a job object and query job information |
| **JobMgr** | This class is used to define a **Job** Manager object and manage jobs and servers |
| **LayerManager** | This class provides functionality to create and manipulate model layers |
| **MaterialFinder** | This class is used to examine material data. You can use this class to access material data properties Note: Some material data is considered proprietary information and will not be returned by the API. Any objects derived from the **MaterialFinder** class will be invalidated when the **MaterialFinder** object is destroyed |

# Class Member



```
Set DiagnosisManager = Synergy.DiagnosisManager()
DiagnosisManager.ShowThickness 1, 2, True
```

```
ShowThickness2 ( double   aMin,
                 double   aMax,
                 Boolean  aAssignLayer,
                 Boolean  aVisibleOnly
               )
```

Generates thickness diagnostics

**Parameters:**

| | |
|---|---|
| *aMin* | lower bound for plotted thickness |
| *aMax* | upper bound for plotted thickness |
| *aAssignLayer* | specify `True` to push the elements within the diagnosis range into a diagnostics layer; `False` otherwise |
| *aVisibleOnly* | Include only visible elements in the diagnosis |

**Example:**

This example shows element thickness between `1` and `2` and pushed these elements into a diagnostics layer.

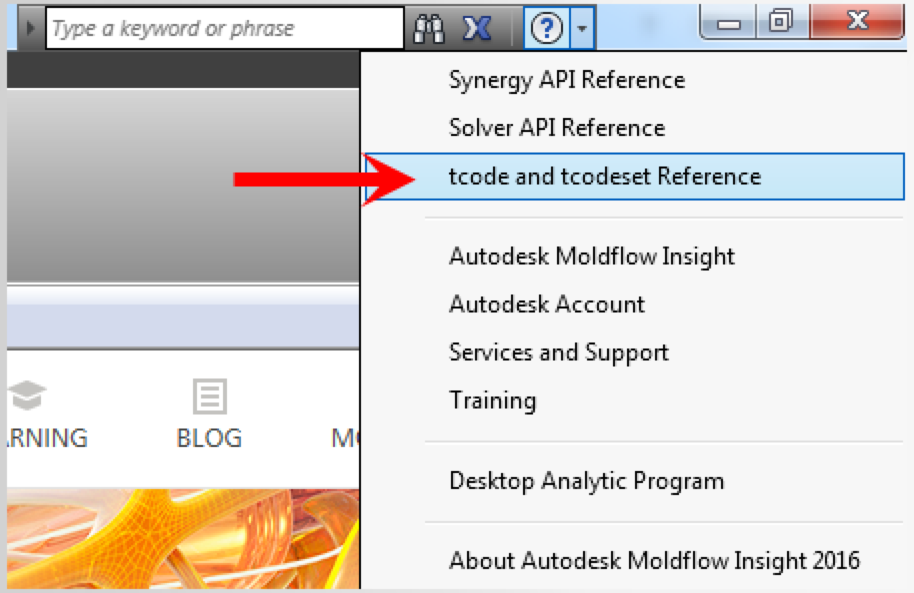
```
Set DiagnosisManager = Synergy.DiagnosisManager()
DiagnosisManager.ShowThickness2 1, 2, True, False
```

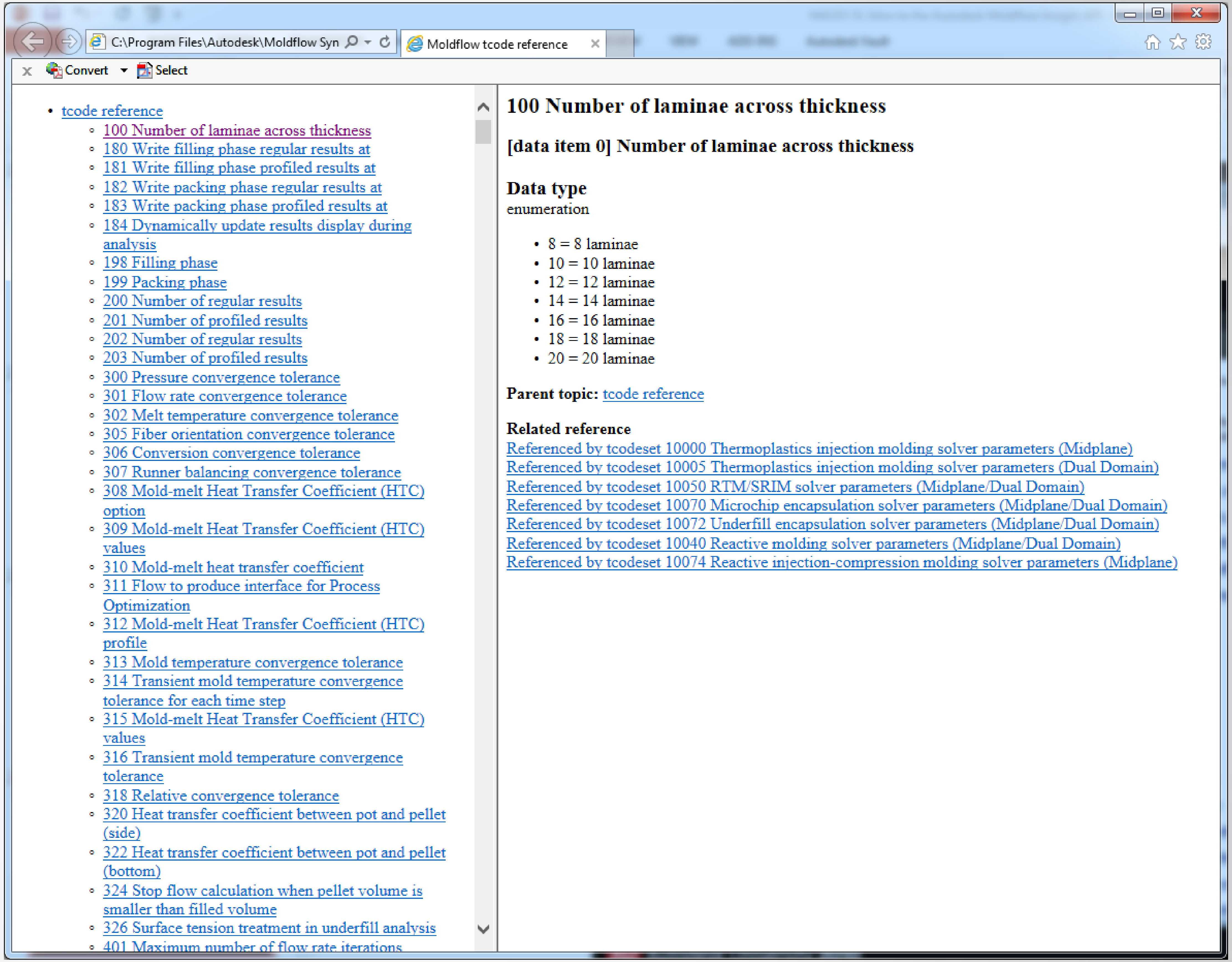
```
ShowAspectRatio ( double   aMin,
                  double   aMax,
                  Boolean  aStdAR,
                  Boolean  aAssignLayer,
```

# TCode Reference

- A **TCode** corresponds to a single feature in the process settings, geometry or solver parameters

- When programming with the Moldflow API, you need to know the numeric ID corresponding to a particular solver or modeling feature





- tcode reference
  - 100 Number of laminae across thickness
  - 180 Write filling phase regular results at
  - 181 Write filling phase profiled results at
  - 182 Write packing phase regular results at
  - 183 Write packing phase profiled results at
  - 184 Dynamically update results display during analysis
  - 198 Filling phase
  - 199 Packing phase
  - 200 Number of regular results
  - 201 Number of profiled results
  - 202 Number of regular results
  - 203 Number of profiled results
  - 300 Pressure convergence tolerance
  - 301 Flow rate convergence tolerance
  - 302 Melt temperature convergence tolerance
  - 305 Fiber orientation convergence tolerance
  - 306 Conversion convergence tolerance
  - 307 Runner balancing convergence tolerance
  - 308 Mold-melt Heat Transfer Coefficient (HTC) option
  - 309 Mold-melt Heat Transfer Coefficient (HTC) values
  - 310 Mold-melt heat transfer coefficient
  - 311 Flow to produce interface for Process Optimization
  - 312 Mold-melt Heat Transfer Coefficient (HTC) profile
  - 313 Mold temperature convergence tolerance
  - 314 Transient mold temperature convergence tolerance for each time step
  - 315 Mold-melt Heat Transfer Coefficient (HTC) values
  - 316 Transient mold temperature convergence tolerance
  - 318 Relative convergence tolerance
  - 320 Heat transfer coefficient between pot and pellet (side)
  - 322 Heat transfer coefficient between pot and pellet (bottom)
  - 324 Stop flow calculation when pellet volume is smaller than filled volume
  - 326 Surface tension treatment in underfill analysis
  - 401 Maximum number of flow rate iterations

# 100 Number of laminae across thickness

## [data item 0] Number of laminae across thickness

### Data type

enumeration

- 8 = 8 laminae
- 10 = 10 laminae
- 12 = 12 laminae
- 14 = 14 laminae
- 16 = 16 laminae
- 18 = 18 laminae
- 20 = 20 laminae

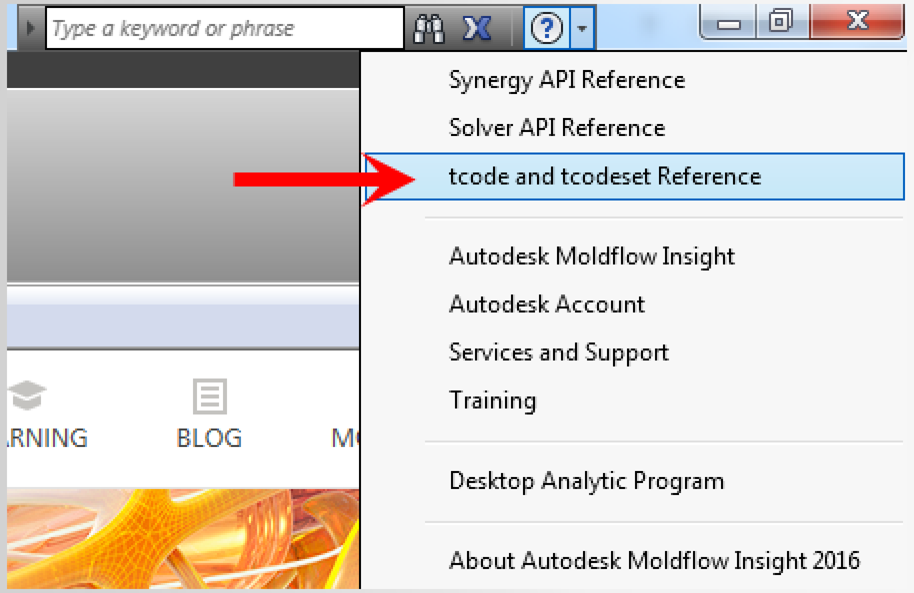
**Parent topic:** tcode reference

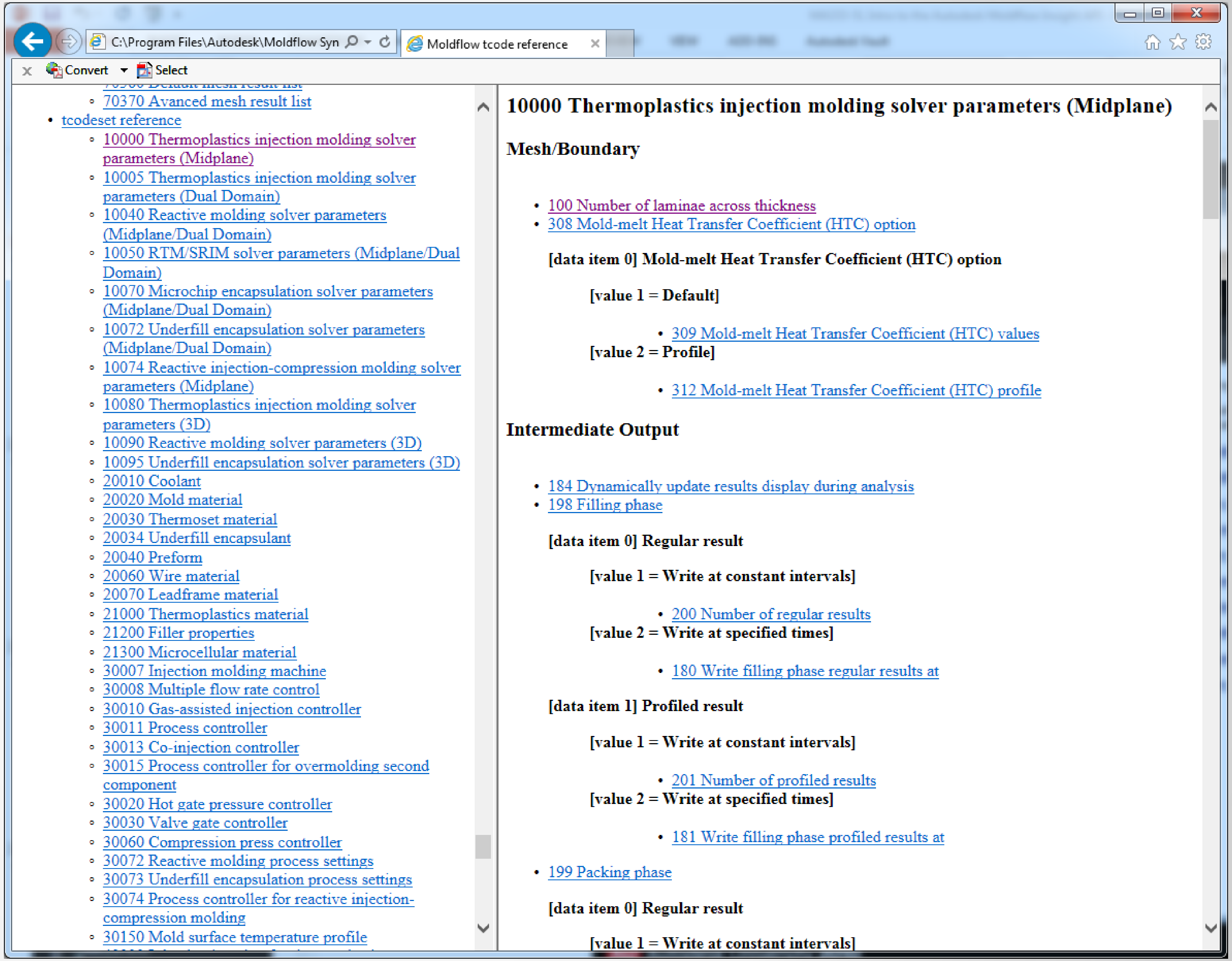
**Related reference**

Referenced by tcodeset 10000 Thermoplastics injection molding solver parameters (Midplane)
Referenced by tcodeset 10005 Thermoplastics injection molding solver parameters (Dual Domain)
Referenced by tcodeset 10050 RTM/SRIM solver parameters (Midplane/Dual Domain)
Referenced by tcodeset 10070 Microchip encapsulation solver parameters (Midplane/Dual Domain)
Referenced by tcodeset 10072 Underfill encapsulation solver parameters (Midplane/Dual Domain)
Referenced by tcodeset 10040 Reactive molding solver parameters (Midplane/Dual Domain)
Referenced by tcodeset 10074 Reactive injection-compression molding solver parameters (Midplane)

# TCodeset  Reference

- A **TCodeset** comprises the TCodes which together control one aspect of a solver
- When programming with the Moldflow API, you need to know the numeric ID corresponding to a particular solver or modeling feature





- 70360 Default mesh result list
- 70370 Avanced mesh result list
- tcodeset reference
  - 10000 Thermoplastics injection molding solver parameters (Midplane)
  - 10005 Thermoplastics injection molding solver parameters (Dual Domain)
  - 10040 Reactive molding solver parameters (Midplane/Dual Domain)
  - 10050 RTM/SRIM solver parameters (Midplane/Dual Domain)
  - 10070 Microchip encapsulation solver parameters (Midplane/Dual Domain)
  - 10072 Underfill encapsulation solver parameters (Midplane/Dual Domain)
  - 10074 Reactive injection-compression molding solver parameters (Midplane)
  - 10080 Thermoplastics injection molding solver parameters (3D)
  - 10090 Reactive molding solver parameters (3D)
  - 10095 Underfill encapsulation solver parameters (3D)
  - 20010 Coolant
  - 20020 Mold material
  - 20030 Thermoset material
  - 20034 Underfill encapsulant
  - 20040 Preform
  - 20060 Wire material
  - 20070 Leadframe material
  - 21000 Thermoplastics material
  - 21200 Filler properties
  - 21300 Microcellular material
  - 30007 Injection molding machine
  - 30008 Multiple flow rate control
  - 30010 Gas-assisted injection controller
  - 30011 Process controller
  - 30013 Co-injection controller
  - 30015 Process controller for overmolding second component
  - 30020 Hot gate pressure controller
  - 30030 Valve gate controller
  - 30060 Compression press controller
  - 30072 Reactive molding process settings
  - 30073 Underfill encapsulation process settings
  - 30074 Process controller for reactive injection-compression molding
  - 30150 Mold surface temperature profile

# 10000 Thermoplastics injection molding solver parameters (Midplane)

## Mesh/Boundary

- 100 Number of laminae across thickness
- 308 Mold-melt Heat Transfer Coefficient (HTC) option

**[data item 0] Mold-melt Heat Transfer Coefficient (HTC) option**

**[value 1 = Default]**

- 309 Mold-melt Heat Transfer Coefficient (HTC) values

**[value 2 = Profile]**

- 312 Mold-melt Heat Transfer Coefficient (HTC) profile

## Intermediate Output

- 184 Dynamically update results display during analysis
- 198 Filling phase

**[data item 0] Regular result**

**[value 1 = Write at constant intervals]**

- 200 Number of regular results

**[value 2 = Write at specified times]**

- 180 Write filling phase regular results at

**[data item 1] Profiled result**

**[value 1 = Write at constant intervals]**

- 201 Number of profiled results

**[value 2 = Write at specified times]**

- 181 Write filling phase profiled results at

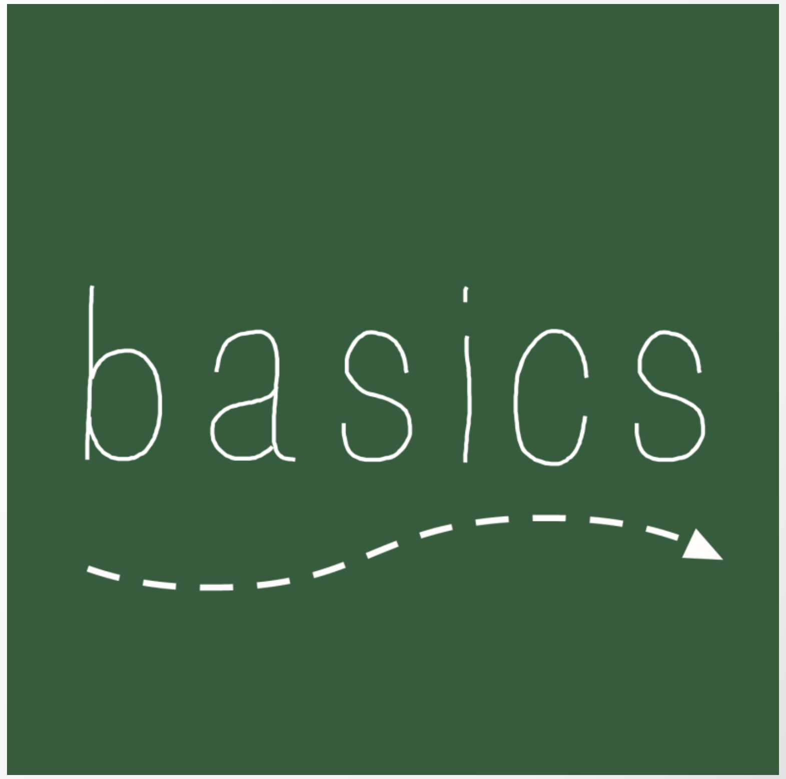
- 199 Packing phase

**[data item 0] Regular result**

**[value 1 = Write at constant intervals]**

# More to Come…

- Afternoon class will dive deeper into
  - Class members
  - TCodes
  - TCodesets
- This class we will stick with the

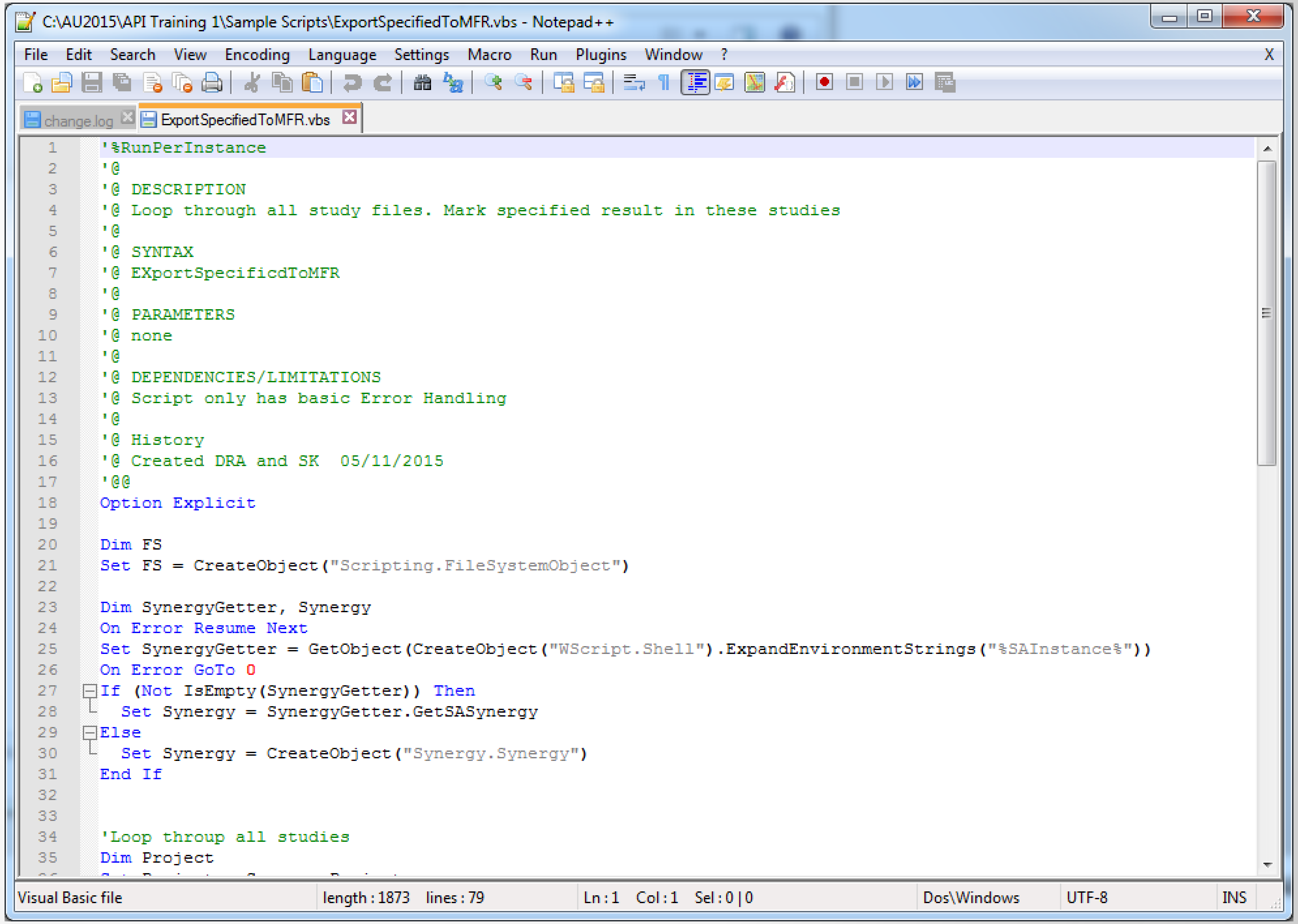


# Vbscript Editors/Debuggers Summary

| | |
|---|---|
| Notepad++ | Editor Only |
| vbsedit | Editor/Debugger |
| Microsoft Visual Studio | Editor/Debugger |

# Notepad ++

- Great tool for writing and editing scripts
- Really an editor only
- Free

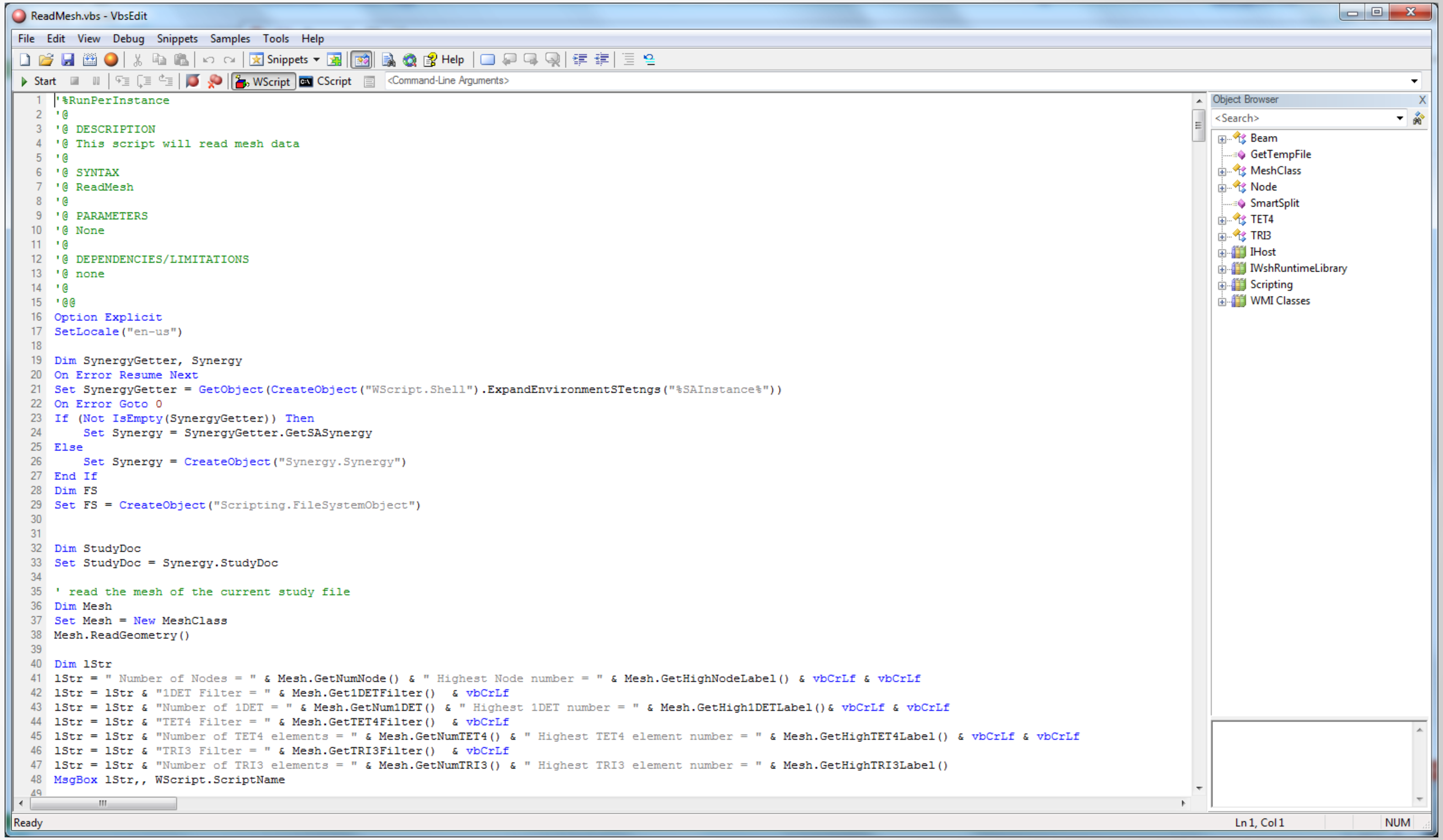


```
'%RunPerInstance
'@
'@ DESCRIPTION
'@ Loop through all study files. Mark specified result in these studies
'@
'@ SYNTAX
'@ EXportSpecificdToMFR
'@
'@ PARAMETERS
'@ none
'@
'@ DEPENDENCIES/LIMITATIONS
'@ Script only has basic Error Handling
'@
'@ History
'@ Created DRA and SK  05/11/2015
'@@
Option Explicit

Dim FS
Set FS = CreateObject("Scripting.FileSystemObject")

Dim SynergyGetter, Synergy
On Error Resume Next
Set SynergyGetter = GetObject(CreateObject("WScript.Shell").ExpandEnvironmentStrings("%SAInstance%"))
On Error GoTo 0
If (Not IsEmpty(SynergyGetter)) Then
  Set Synergy = SynergyGetter.GetSASynergy
Else
  Set Synergy = CreateObject("Synergy.Synergy")
End If


'Loop throup all studies
Dim Project
```

# VbsEdit

- Great tool for writing, debugging and editing scripts
- Free version available
- We will use this to run scripts using the Start icon

VbsEdit

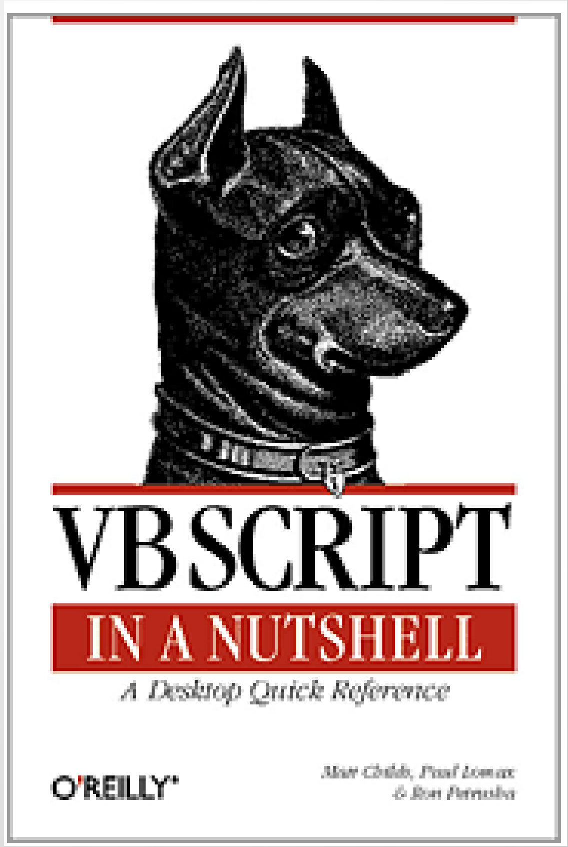


```
'%RunPerInstance
'@
'@ DESCRIPTION
'@ This script will read mesh data
'@
'@ SYNTAX
'@ ReadMesh
'@
'@ PARAMETERS
'@ None
'@
'@ DEPENDENCIES/LIMITATIONS
'@ none
'@
'@@
Option Explicit
SetLocale("en-us")

Dim SynergyGetter, Synergy
On Error Resume Next
Set SynergyGetter = GetObject(CreateObject("WScript.Shell").ExpandEnvironmentSTetngs("%SAInstance%"))
On Error Goto 0
If (Not IsEmpty(SynergyGetter)) Then
    Set Synergy = SynergyGetter.GetSASynergy
Else
    Set Synergy = CreateObject("Synergy.Synergy")
End If
Dim FS
Set FS = CreateObject("Scripting.FileSystemObject")


Dim StudyDoc
Set StudyDoc = Synergy.StudyDoc

' read the mesh of the current study file
Dim Mesh
Set Mesh = New MeshClass
Mesh.ReadGeometry()

Dim lStr
lStr = " Number of Nodes = " & Mesh.GetNumNode() & " Highest Node number = " & Mesh.GetHighNodeLabel() & vbCrLf & vbCrLf
lStr = lStr & "1DET Filter = " & Mesh.Get1DETFilter()  & vbCrLf
lStr = lStr & "Number of 1DET = " & Mesh.GetNum1DET() & " Highest 1DET number = " & Mesh.GetHigh1DETLabel()& vbCrLf & vbCrLf
lStr = lStr & "TET4 Filter = " & Mesh.GetTET4Filter()  & vbCrLf
lStr = lStr & "Number of TET4 elements = " & Mesh.GetNumTET4() & " Highest TET4 element number = " & Mesh.GetHighTET4Label() & vbCrLf & vbCrLf
lStr = lStr & "TRI3 Filter = " & Mesh.GetTRI3Filter()  & vbCrLf
lStr = lStr & "Number of TRI3 elements = " & Mesh.GetNumTRI3() & " Highest TRI3 element number = " & Mesh.GetHighTRI3Label()
MsgBox lStr,, WScript.ScriptName
```

# vbScript Programming Resources

- Online
  - Include the word vbscript in any searches
  - Searches often returns results for Visual Basic for Applications(VBA)
    - VBA is compiled, vbscript is interpreted
  - There are a number of user guides for vbscript
    - Search for vbscript Language Reference or vbscript User Guide
- Books

VBScript in a Nutshell



# API Related Utilities: StudyMod and StudyRlt

- Tools
  - **StudyMod** – Modify an existing study File
  - **StudyRlt** – Extract Result Data
- Key Features
  - Provided on both Windows and Linux
  - Command line driven
  - Can be used within API scripts
  - Typically used to interface with 3rd party optimization packages

# API Related Utilities: StudyMod

- StudyMod - Modify an existing Study file
  - Allows changes to
    - Boundary Conditions
    - Mesh
    - Processing Settings
    - Materials
  - Not designed to create/add/modify geometry
  - Modifier file is in XML format

```
NAME:
  Studymod - Modify a study File

SYNOPSIS:
  studymod  <InputStudy> <OutputStudy> <ModifierFile>
```

# API Related Utilities: StudyMod

- Example Modifier File

```
<?xml version="1.0" encoding="utf-8"?>
<StudyMod title="Autodesk StudyMod" ver="1.00">
    <UnitSystem>Metric</UnitSystem>
    <Property>
             <TSet>
                  <!--Process controller-->
                  <ID>30011</ID>
                  <SubID>1</SubID>
                  <!--Melt temperature-->
                  <TCode>
                         <ID>11002</ID>
                         <Description>Melt temperature</Description>
                         <Value>240</Value>
                  </TCode>
          </TSet>
    </Property>
</StudyMod>
```

Set the Melt Temperature to 240C

# API Related Utilities: StudyRlt

- StudyRlt – Extract Result Data
  - Data which can be extracted
    - Individual message output (-message)
    - Sequence screen output (-exportoutput)
    - Result in XML format (-xml)
    - Model in Patran format (-exportpatran)
    - Value from Result Data (-result)
      - By Region/Layer
      - Calculation: Min/Max/Average

# API Related Utilities: StudyRlt

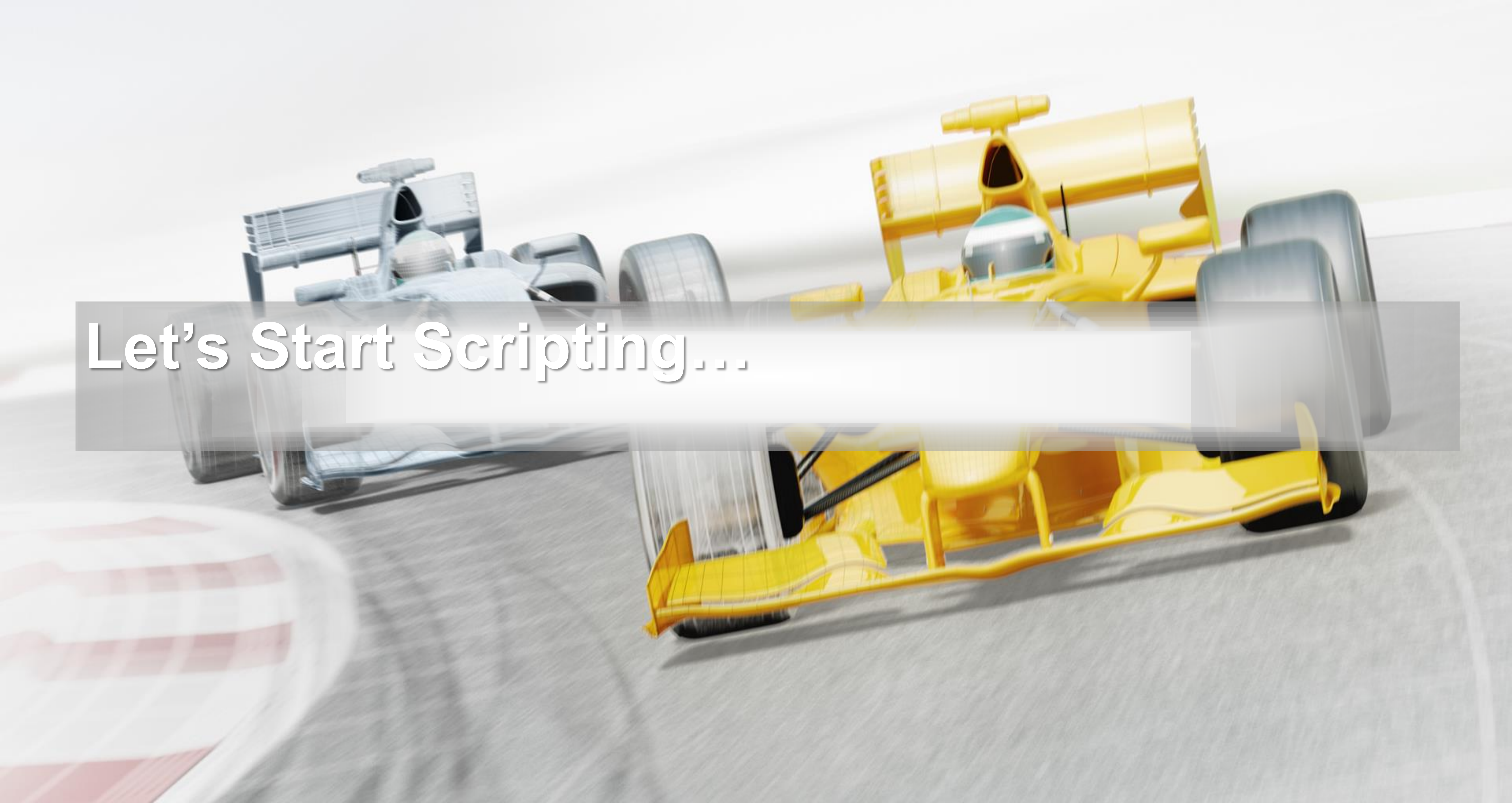
NAME:

Studyrlt - Result Extraction Utility

SYNOPSIS:

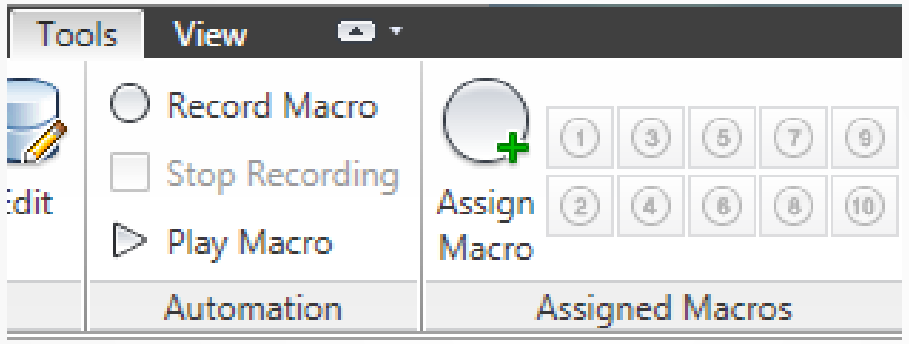
```
 studyrlt  <study> -message <sequence>  <message ID> <occurrence> <item> [-unit SI|Metric|English]
     <study> -exportoutput [<sequence>] [-output <filename>] [-unit SI|Metric|English]
     <study> -exportwarp <result ID> -actual|-opposite -scale <x> -output <filename> [-unit SI|Metric|English]
     <study> -xml <result ID>
     <study> -exportpatran
     <study> -result <result ID>
                      -min|-max|-average|-stddev|-count|-node <node number>|-element <element number>
                      [-layer <layer name>|-cavity|-gate|-runner|-sprue]
                      [-component <number> [-anchor <node1> <node2> <node3>]
                      [-unit SI|Metric|English]
```

# Let’s Start Scripting…

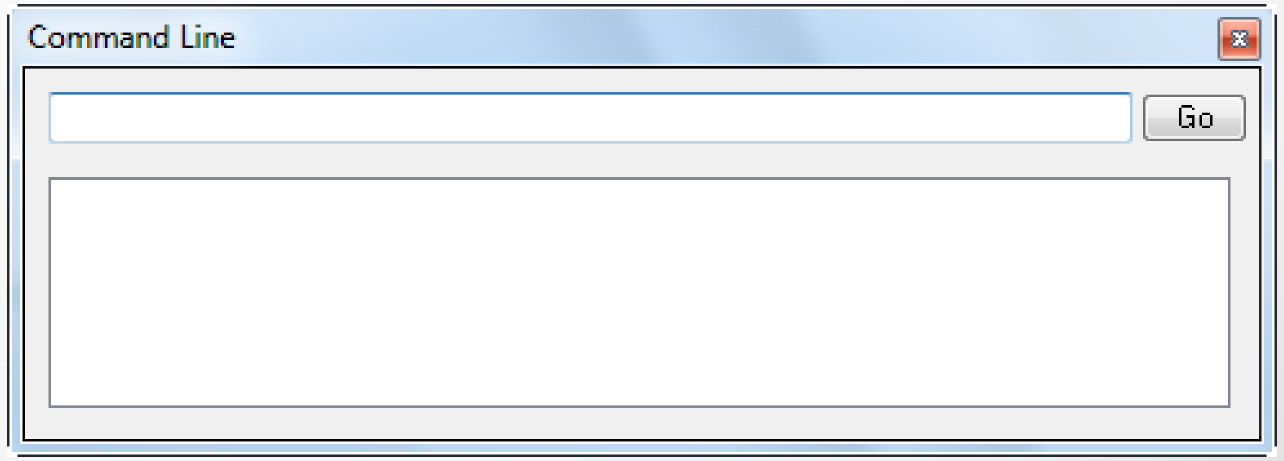


# Two Ways to Automate AMI Using the API

1. Macros can be recorded and played from within the interface (Play Macro) or from the AMI command line



2. VB scripts can be run from the AMI command line with or without command line arguments, or they can be run from within AMI as macros

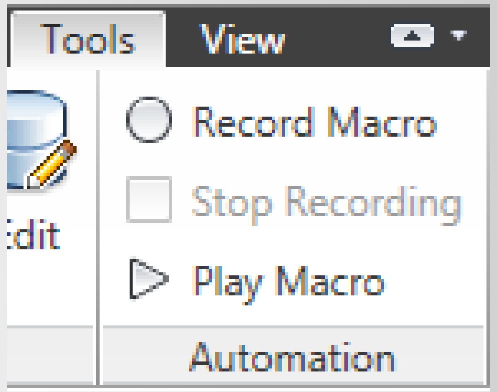


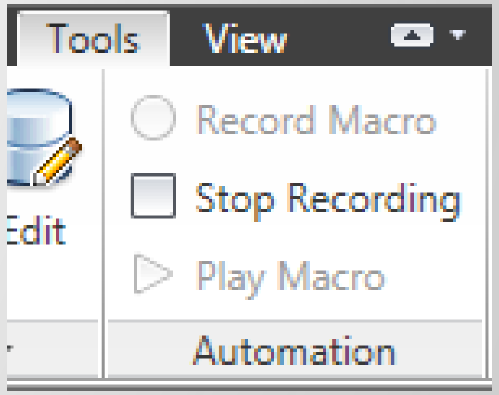
# Macros

- Macro recording and playback allows you to repeat user-interface actions to automate common or repetitive tasks
- Macro recording and playback is built on top of the basic OLE automation interface
- Uses Visual Basic script as the recording and playback language
- By default, recorded macros are saved with a .vbs extension in the folder specified during installation:
  - My Documents\My AMI 20xx Projects\scripts,
    (where xx is the software release number, i.e. 2016)

# Creating Macros (Recording)

1. Select the **Tools** tab at the top of the Ribbon interface
2. In the **Automation** panel, click **Record Macro** to start the macro recording
3. Perform a sequence of actions in the Autodesk Moldflow Insight Synergy interface
4. In the **Automation** panel, click **Stop Recording** to stop the macro recording
5. The **Save Macro** dialog appears, enter a name in the **File Name** box and click **Save**

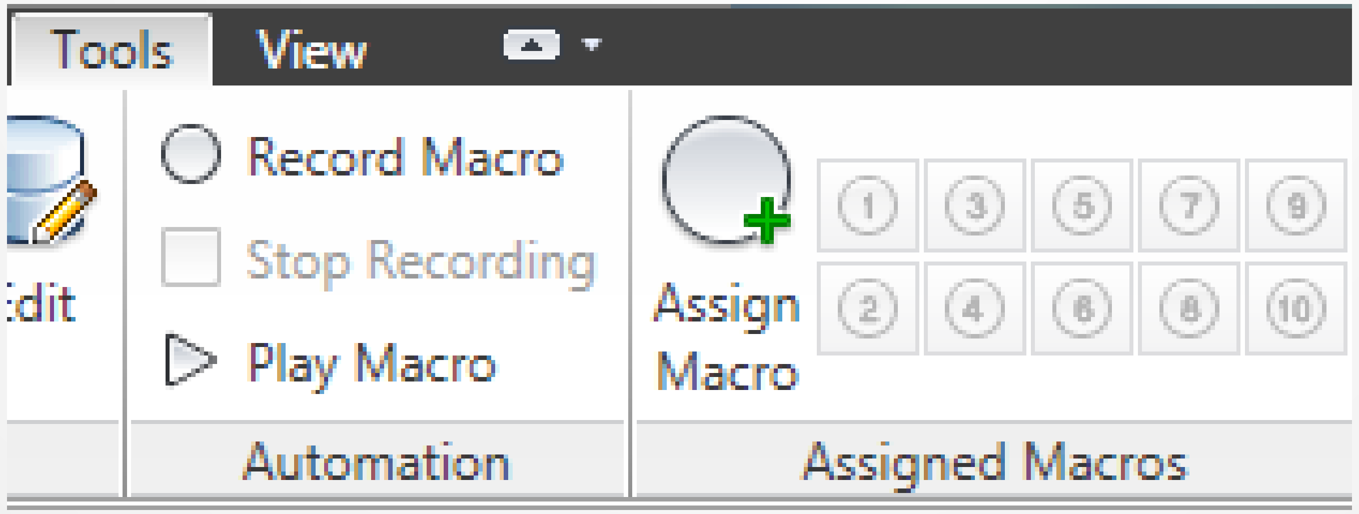




# Macro Playback (In Synergy)

1. Select the **Tools** tab at the top of the Ribbon interface
2. In the **Automation** panel, click **Play Macro** and the **Open Macro** dialog appears
3. Select a macro file of your choice and click **Open**. The macro plays

- Also the ability to assign macros to a GUI icon button



# Macro Playback (Windows Explorer)

1. Locate the macro you want to run in Window Explorer
2. Double-click the required script.  The software will start if it is not currently running, and the script will open and run

- Scripts that assume a particular study is already open, or that a particular result is already displayed, are not likely to work
- If your script uses parameters that are entered as command line arguments, you must run the script from the software command line

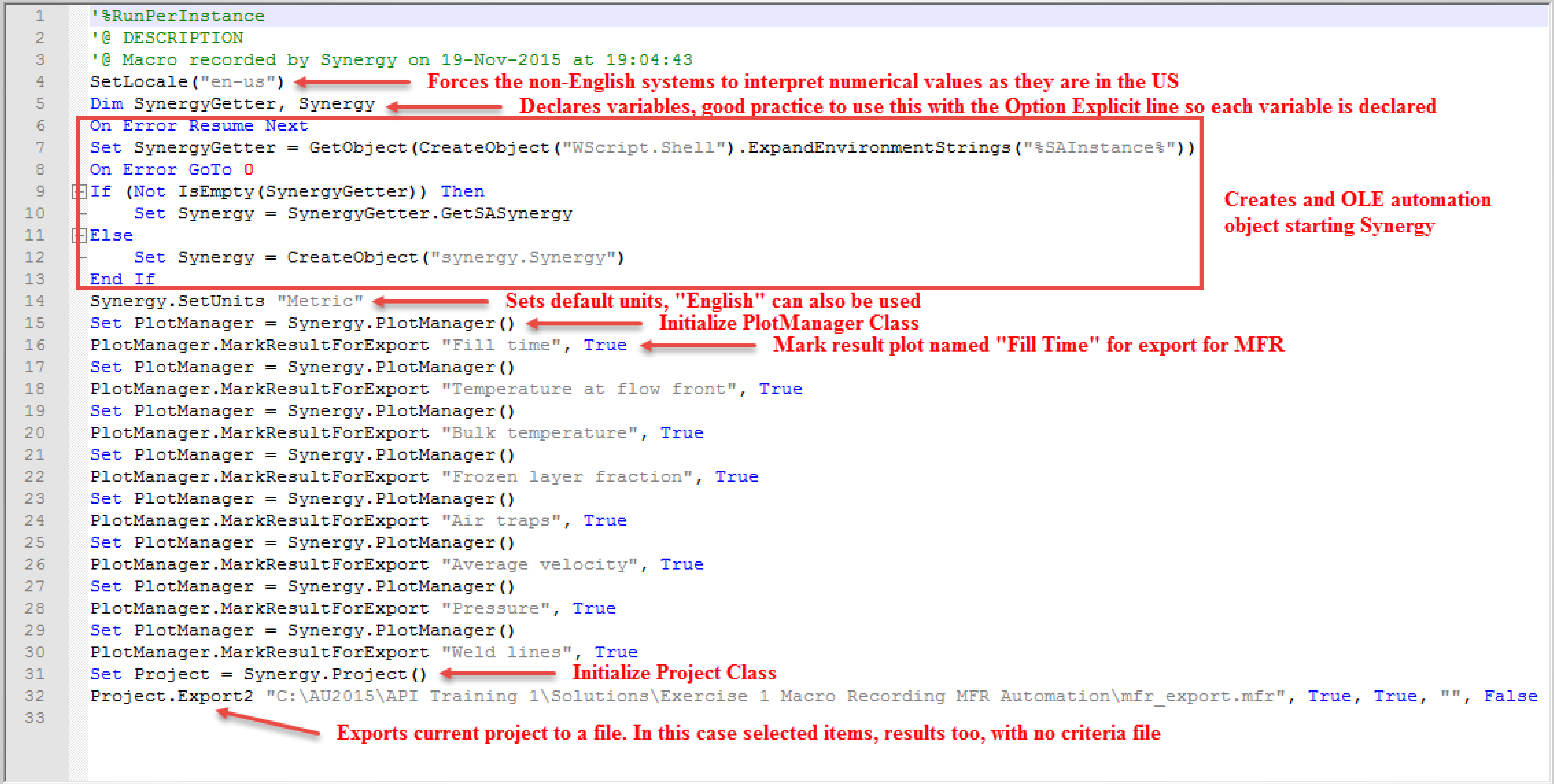
# Issues with Macro Record/Playback

- State/Version Dependency
  - Macros are recorded from a defined starting point
    - Unless the starting point is always the same, behavior may be unpredictable
    - Be careful when recorded macros use
      - Selection Lists
      - Results
      - Path Names to Files Projects
- Not all events are recorded
  - Some commands do not have API support (See Help for list)
    - Mouse clicks associated with selection are not explicitly recorded
    - The API records only the final entities selected

# Exercise 1: Macro Recording Automating Communicator MFR File Creation

# Exercise 1 General Steps:

1. Open EX1.mpi project
2. Open tutorial_model study
3. Start recording macro
4. Select the following results for Communicator mfr export
   - Fill time, Temperature at flow front, Bulk temperature, Frozen layer fraction, Air traps, Average velocity, Pressure, Weld lines
5. Export and publish MFR to a location
6. Stop recording macro and save script as “mfr_export.vbs”
7. Open and review saved macro script in Notepad ++
8. Try to use the saved script on next study (results_study)
9. Review MFR file in Communicator



```
'%RunPerInstance
'@ DESCRIPTION
'@ Macro recorded by Synergy on 19-Nov-2015 at 19:04:43
SetLocale("en-us")
Dim SynergyGetter, Synergy
On Error Resume Next
Set SynergyGetter = GetObject(CreateObject("WScript.Shell").ExpandEnvironmentStrings("%SAInstance%"))
On Error GoTo 0
If (Not IsEmpty(SynergyGetter)) Then
    Set Synergy = SynergyGetter.GetSASynergy
Else
    Set Synergy = CreateObject("synergy.Synergy")
End If
Synergy.SetUnits "Metric"
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Fill time", True
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Temperature at flow front", True
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Bulk temperature", True
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Frozen layer fraction", True
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Air traps", True
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Average velocity", True
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Pressure", True
Set PlotManager = Synergy.PlotManager()
PlotManager.MarkResultForExport "Weld lines", True
Set Project = Synergy.Project()
Project.Export2 "C:\AU2015\API Training 1\Solutions\Exercise 1 Macro Recording MFR Automation\mfr_export.mfr", True, True, "", False
```

Forces the non-English systems to interpret numerical values as they are in the US

Declares variables, good practice to use this with the Option Explicit line so each variable is declared

Creates and OLE automation object starting Synergy

Sets default units, "English" can also be used

Initialize PlotManager Class

Mark result plot named "Fill Time" for export for MFR

Initialize Project Class

Exports current project to a file. In this case selected items, results too, with no criteria file

Issues? Questions?

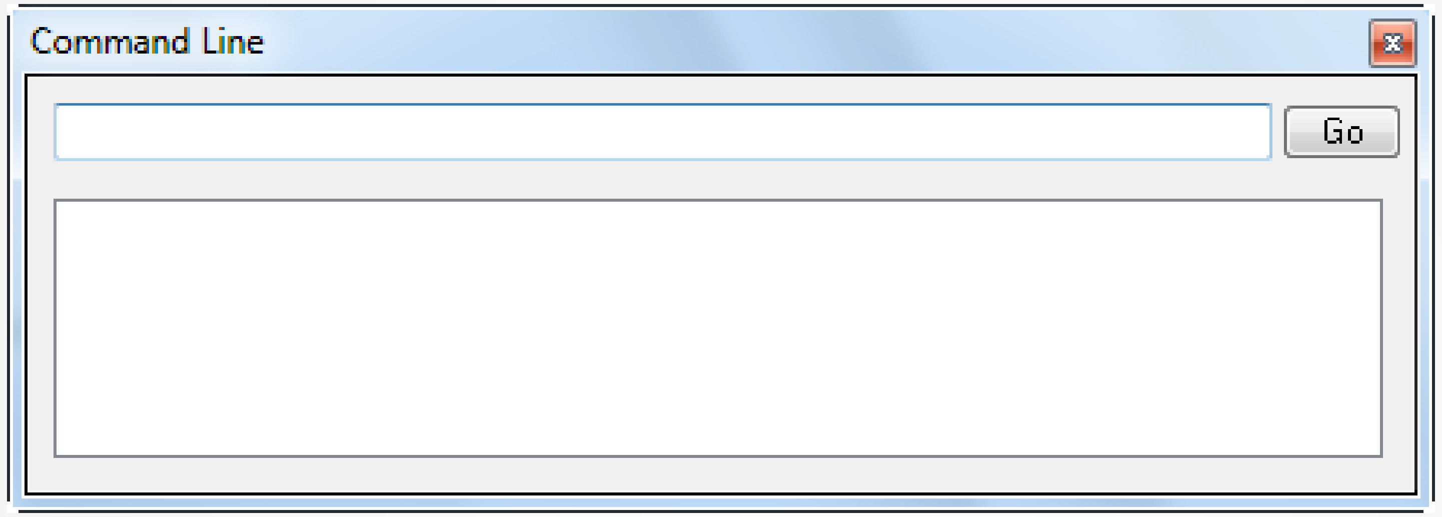
Hint: Look in Solutions Folder

# AMI Command Line

- The AMI command line allows you to invoke Visual Basic Scripts (VBS) and macros from a command line user interface within Synergy
- This allows you to run scripts that require you to provide parameters
- Note that this capability is dynamic and extensible.
  - You can modify existing scripts to better suit your needs or extend them in order to add new capability
  - You can add new scripts (containing multiple commands) that can be used just like the "built-in" commands that come with your AMI installation

# Command Line

- **View** tab > **User Interface** > **Command Line** to open the dialog



- Enter the name of the script without its file extension followed by any command line arguments the script requires
  - Example: dr 25 25 30 rotates the model to these X, Y, Z angles in the Synergy graphics window

- Select **Go** or hit the **Enter** key and the script plays

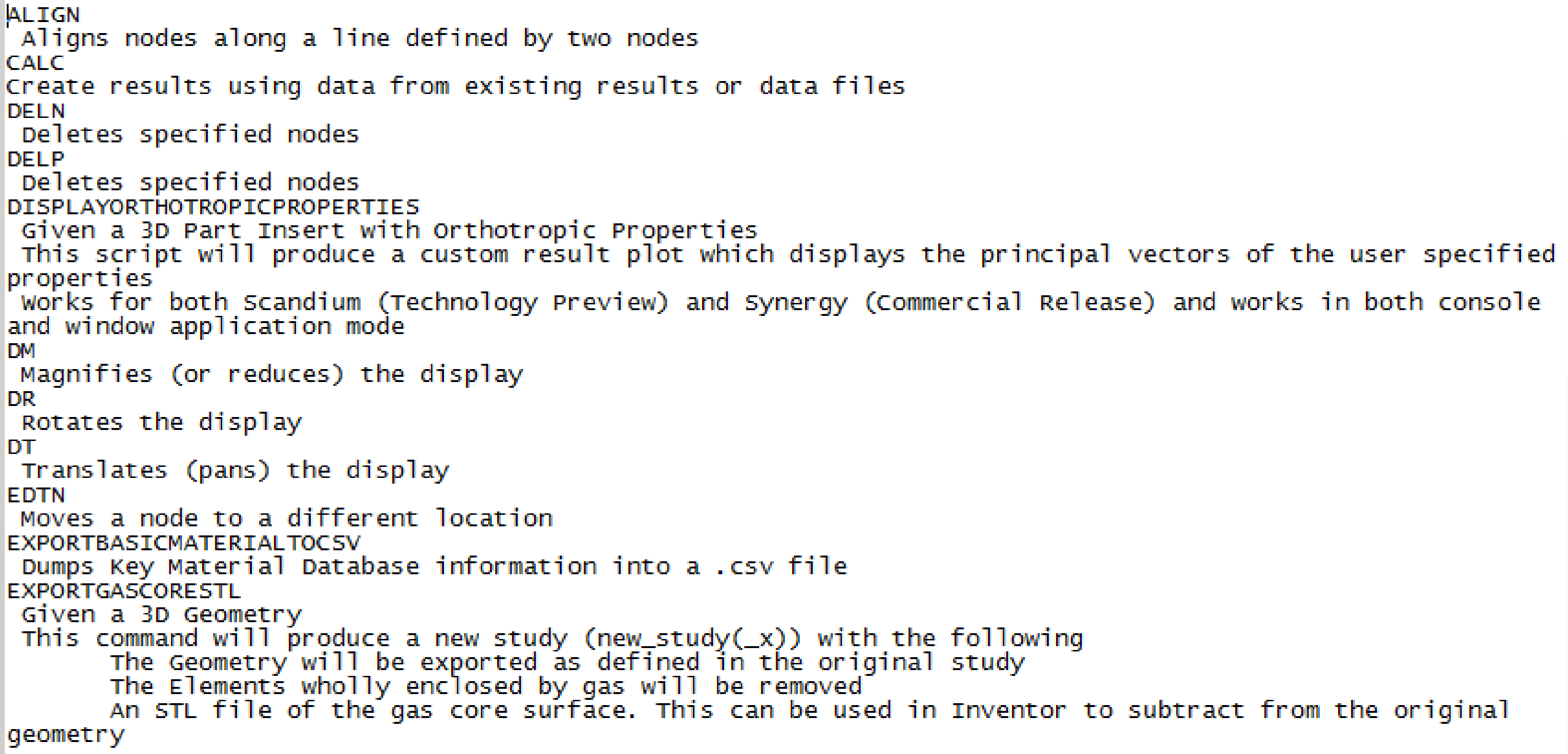
# Command Scripts Locations

- Your default project folder specified during installation:
  - My Documents\My AMI 20xx Projects\commands
    (where xx is the software release)

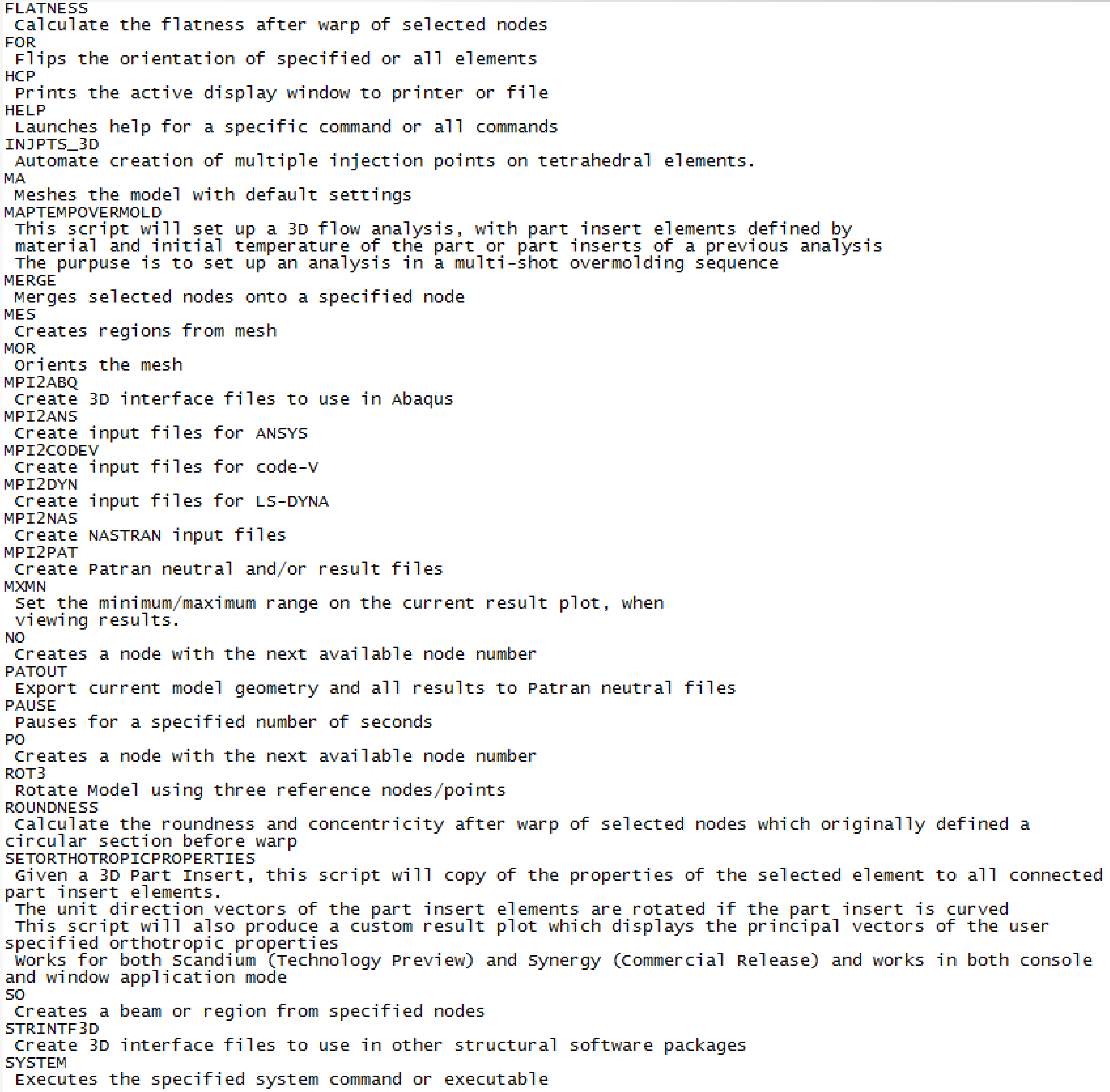
- If AMI cannot find the script in the default project folder, it searches the AMI commands folder
  - Typically, Program Files\Autodesk\Moldflow Insight 20xx\data\commands
    (where xx is the software release)

# Command Line

- Preloaded scripts available
  - Type **`help`** in command line
  - MFView/Modeler commands



```
ALIGN
 Aligns nodes along a line defined by two nodes
CALC
Create results using data from existing results or data files
DELN
 Deletes specified nodes
DELP
 Deletes specified nodes
DISPLAYORTHOTROPICPROPERTIES
 Given a 3D Part Insert with Orthotropic Properties
 This script will produce a custom result plot which displays the principal vectors of the user specified
properties
 Works for both Scandium (Technology Preview) and Synergy (Commercial Release) and works in both console
and window application mode
DM
 Magnifies (or reduces) the display
DR
 Rotates the display
DT
 Translates (pans) the display
EDTN
 Moves a node to a different location
EXPORTBASICMATERIALTOCSV
 Dumps Key Material Database information into a .csv file
EXPORTGASCORESTL
 Given a 3D Geometry
 This command will produce a new study (new_study(_x)) with the following
       The Geometry will be exported as defined in the original study
       The Elements wholly enclosed by gas will be removed
       An STL file of the gas core surface. This can be used in Inventor to subtract from the original
geometry
```



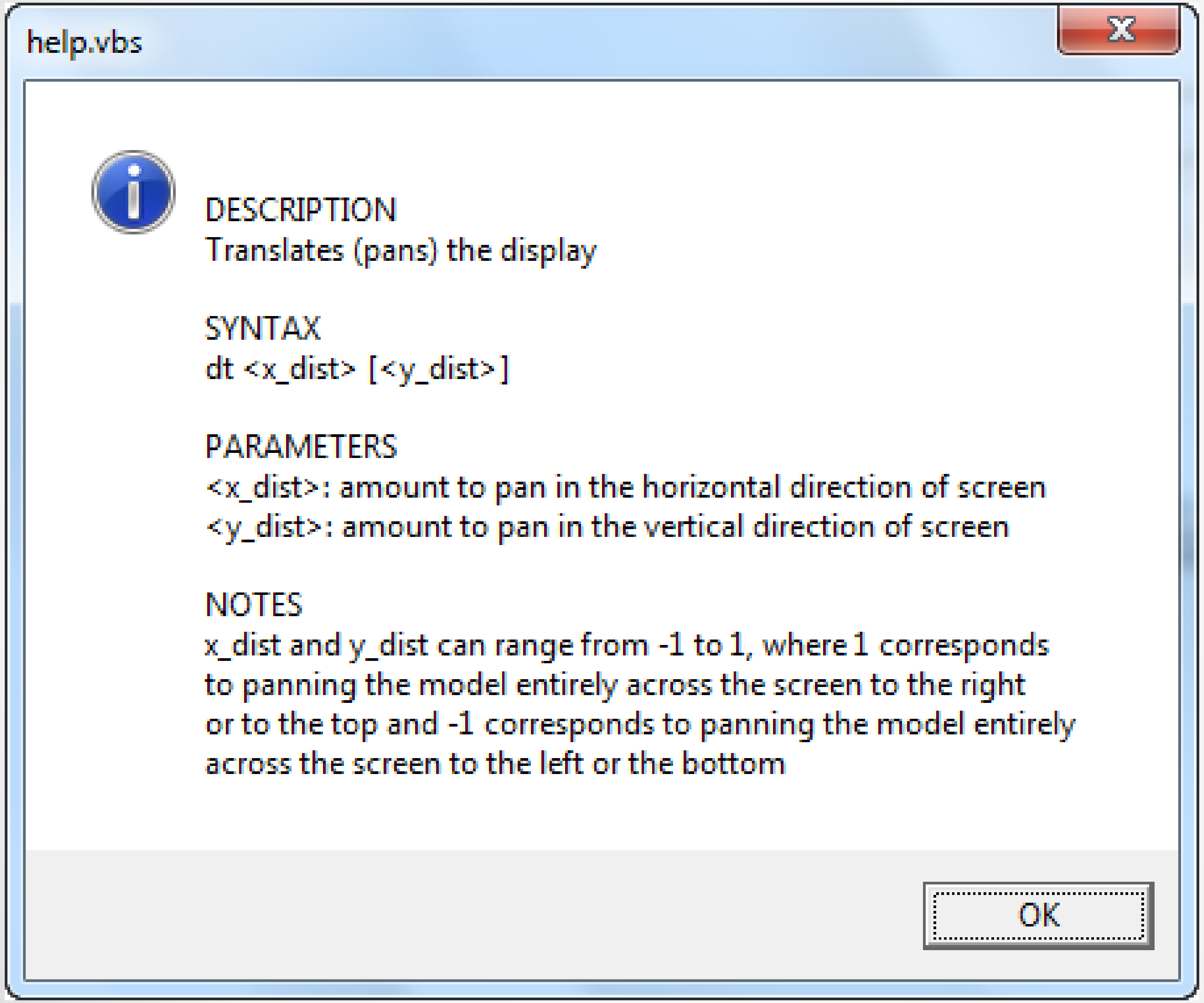
```
FLATNESS
 Calculate the flatness after warp of selected nodes
FOR
 Flips the orientation of specified or all elements
HCP
 Prints the active display window to printer or file
HELP
 Launches help for a specific command or all commands
INJPTS_3D
 Automate creation of multiple injection points on tetrahedral elements.
MA
 Meshes the model with default settings
MAPTEMPOVERMOLD
 This script will set up a 3D flow analysis, with part insert elements defined by
 material and initial temperature of the part or part inserts of a previous analysis
 The purpose is to set up an analysis in a multi-shot overmolding sequence
MERGE
 Merges selected nodes onto a specified node
MES
 Creates regions from mesh
MOR
 Orients the mesh
MPI2ABQ
 Create 3D interface files to use in Abaqus
MPI2ANS
 Create input files for ANSYS
MPI2CODEV
 Create input files for code-V
MPI2DYN
 Create input files for LS-DYNA
MPI2NAS
 Create NASTRAN input files
MPI2PAT
 Create Patran neutral and/or result files
MXMN
 Set the minimum/maximum range on the current result plot, when
 viewing results.
NO
 Creates a node with the next available node number
PATOUT
 Export current model geometry and all results to Patran neutral files
PAUSE
 Pauses for a specified number of seconds
PO
 Creates a node with the next available node number
ROT3
 Rotate Model using three reference nodes/points
ROUNDNESS
 Calculate the roundness and concentricity after warp of selected nodes which originally defined a
circular section before warp
SETORTHOTROPICPROPERTIES
 Given a 3D Part Insert, this script will copy of the properties of the selected element to all connected
part insert elements.
 The unit direction vectors of the part insert elements are rotated if the part insert is curved
 This script will also produce a custom result plot which displays the principal vectors of the user
specified orthotropic properties
 Works for both Scandium (Technology Preview) and Synergy (Commercial Release) and works in both console
and window application mode
SO
 Creates a beam or region from specified nodes
STRINTF3D
 Create 3D interface files to use in other structural software packages
SYSTEM
 Executes the specified system command or executable
```

# Command Line Help for a Script

- If available, type **`help script_name`** (where script_name is the name of the script)

- Example: Type **`help DT`** into the command line text field and click Go
  - Information about what function the script performs (DESCRIPTION)
  - Information about what you must type in on the command line to launch the script (SYNTAX)
  - Information about any parameter values that you need to type in on the command line when you run the script (PARAMETERS)
  - Any further information about the script (NOTES)

# Command Line Help DT Example



DESCRIPTION
Translates (pans) the display

SYNTAX
dt <x_dist> [<y_dist>]

PARAMETERS
<x_dist>: amount to pan in the horizontal direction of screen
<y_dist>: amount to pan in the vertical direction of screen

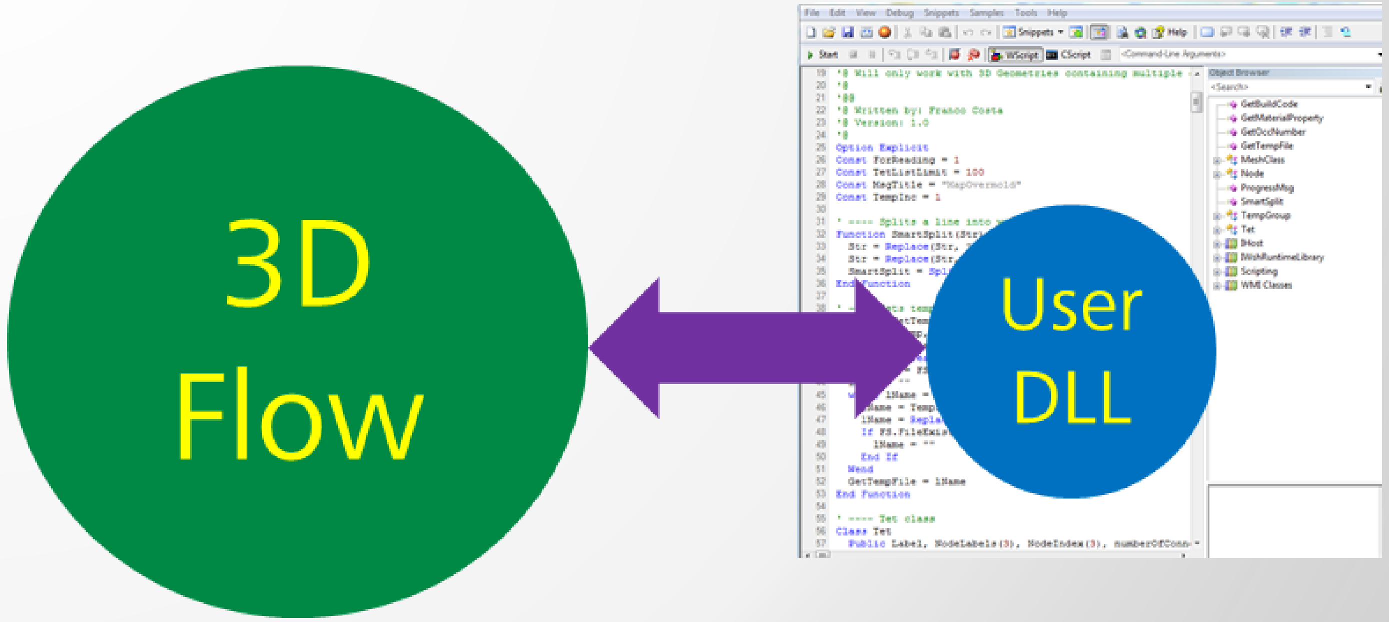
NOTES
x_dist and y_dist can range from -1 to 1, where 1 corresponds
to panning the model entirely across the screen to the right
or to the top and -1 corresponds to panning the model entirely
across the screen to the left or the bottom

# Additional Information about Scripts

- Scripts that do not have command line arguments (input on the AMI command line) can be run as macros
- If you need to create a script that uses parameters, you should write a script that takes command line arguments
- There are limitations to the API. For the latest list, look under Help > Application Programming Interface (API) > Limitations in API functionality

# Solver API – 3D Flow (mhb3d)

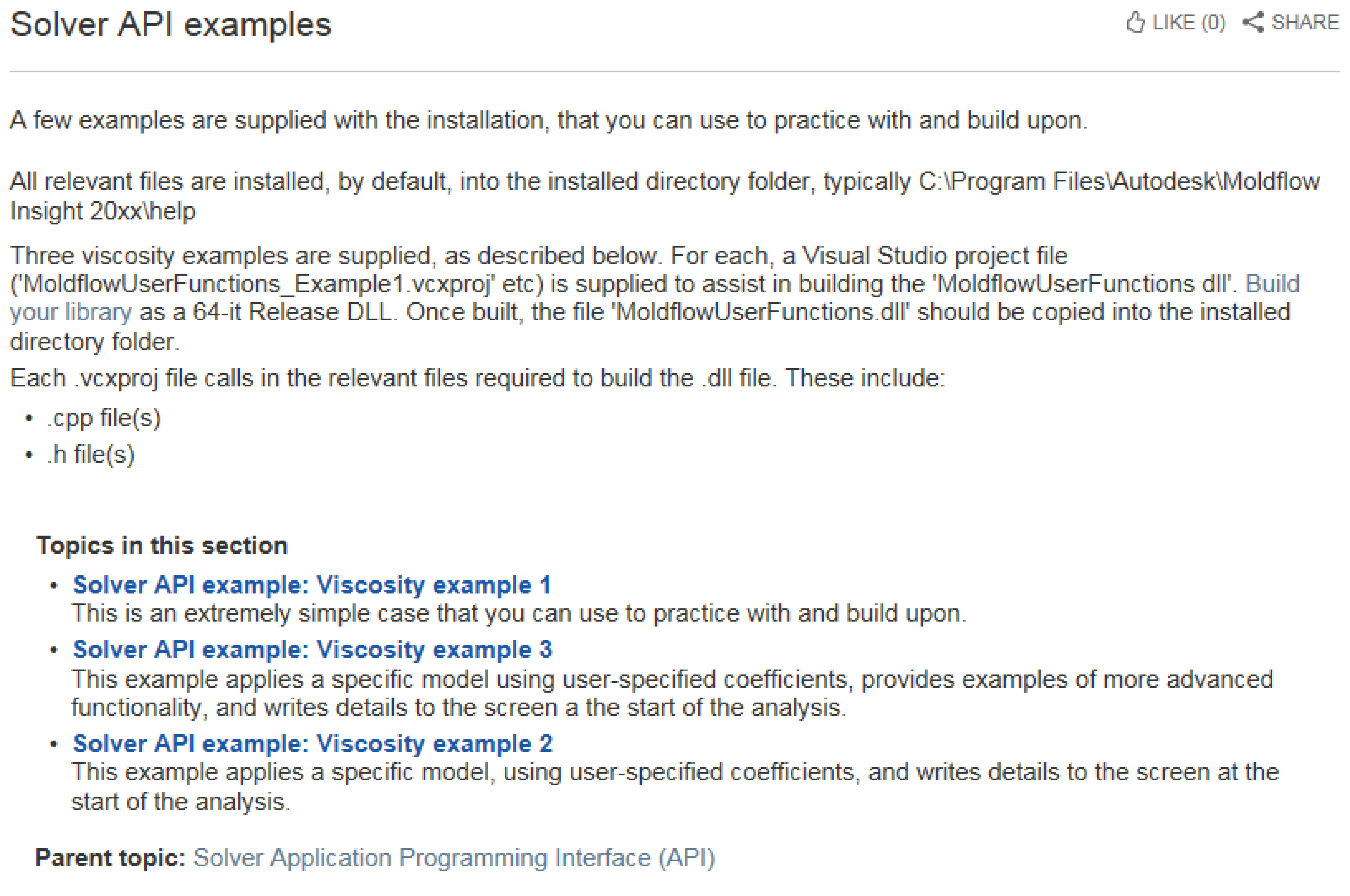
- Allow user routines for selected properties and calculations
  - Easier research collaborations
  - User-coded routines in a DLL
    - Example template provided
  - First implementation users can supply their own function to calculate thermoplastic viscosity in 3D Flow analyses
  - Others to follow:
    - PVT ?
    - Fiber Orientation ?
    - Curing ?



```
'@ Will only work with 3D Geometries containing multiple
'@
'@@
'@ Written by: Franco Costa
'@ Version: 1.0
'@
Option Explicit
Const ForReading = 1
Const TetListLimit = 100
Const MsgTitle = "MapOvermold"
Const TempInc = 1

' ==== Splits a line into
Function SmartSplit(Str)
  Str = Replace(Str,
  Str = Replace(Str,
  SmartSplit = Spl
End Function

    iName =
    iName = Temp
    iName = Repla
    If FS.FileExis
      iName = ""
    End If
  Wend
  GetTempFile = iName
End Function

' ---- Tet class
Class Tet
  Public Label, NodeLabels(3), NodeIndex(3), numberOfConn
```

# Solver API – 3D Flow

- This special API functionality is not covered in this class
- See examples in Help (Topic “Solver API Examples”)



## Solver API examples

A few examples are supplied with the installation, that you can use to practice with and build upon.

All relevant files are installed, by default, into the installed directory folder, typically C:\Program Files\Autodesk\Moldflow Insight 20xx\help

Three viscosity examples are supplied, as described below. For each, a Visual Studio project file ('MoldflowUserFunctions_Example1.vcxproj' etc) is supplied to assist in building the 'MoldflowUserFunctions dll'. Build your library as a 64-it Release DLL. Once built, the file 'MoldflowUserFunctions.dll' should be copied into the installed directory folder.

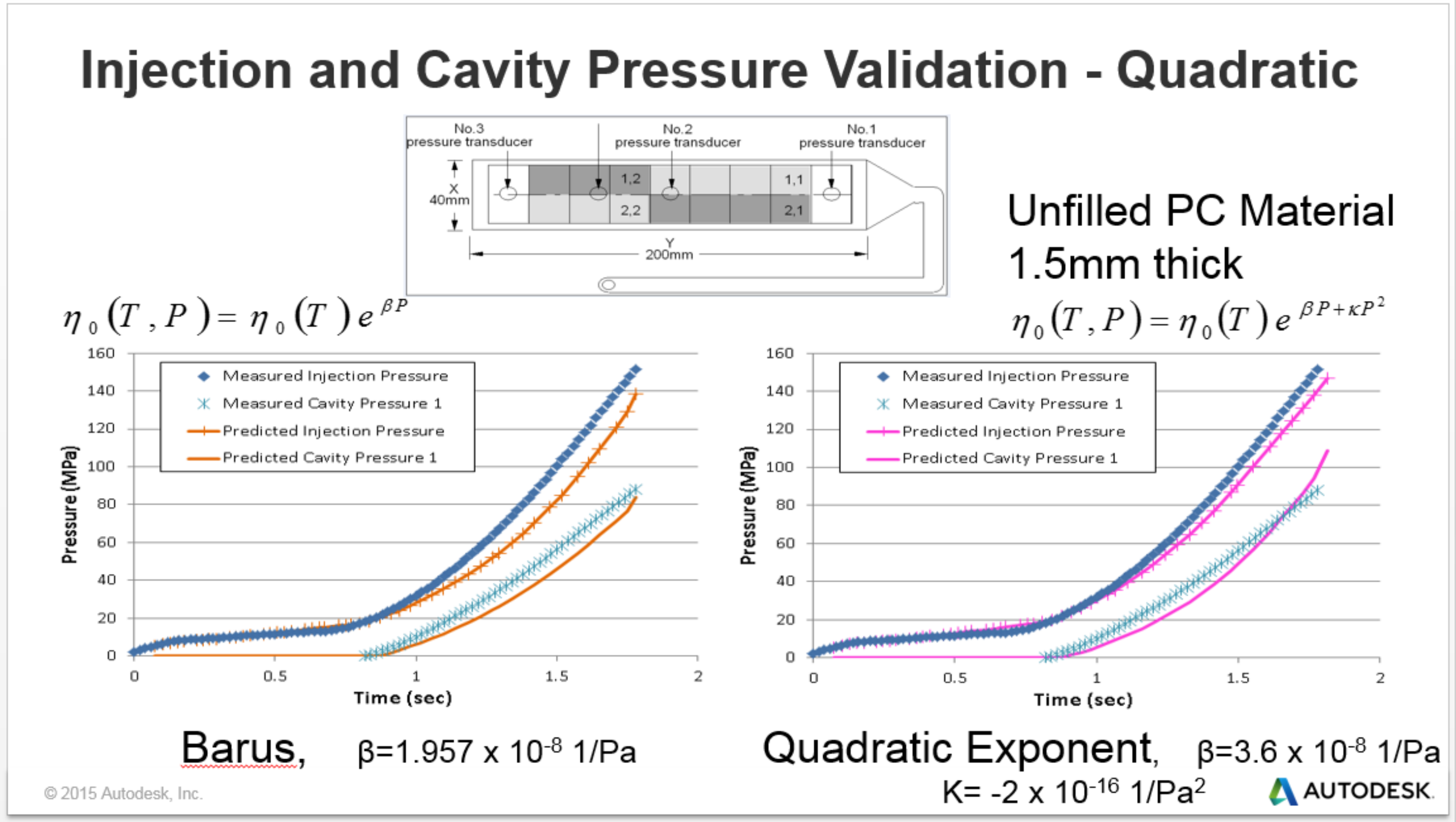
Each .vcxproj file calls in the relevant files required to build the .dll file. These include:

- .cpp file(s)
- .h file(s)

**Topics in this section**

- **Solver API example: Viscosity example 1**
  This is an extremely simple case that you can use to practice with and build upon.
- **Solver API example: Viscosity example 3**
  This example applies a specific model using user-specified coefficients, provides examples of more advanced functionality, and writes details to the screen a the start of the analysis.
- **Solver API example: Viscosity example 2**
  This example applies a specific model, using user-specified coefficients, and writes details to the screen at the start of the analysis.

**Parent topic:** Solver Application Programming Interface (API)



## Injection and Cavity Pressure Validation - Quadratic

Unfilled PC Material
1.5mm thick

$$\eta_0(T,P) = \eta_0(T)\, e^{\beta P}$$

$$\eta_0(T,P) = \eta_0(T)\, e^{\beta P + \kappa P^2}$$

Barus, β=1.957 x $10^{-8}$ 1/Pa

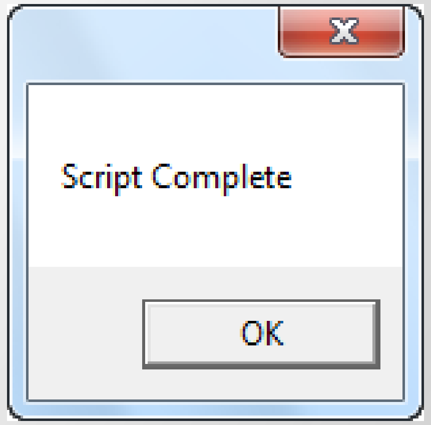
Quadratic Exponent, β=3.6 x $10^{-8}$ 1/Pa
K= -2 x $10^{-16}$ 1/$Pa^2$

# Programming Workflow

- Determine what you want the script to do
- Take advantage of macro recording to get the general guidelines for the script
- Edit the macro using an editor (e.g. notepad, notepad ++)
- Create additional objects and code as needed
    - Use references such as API Reference as a guide
    - Use code from other scripts, don’t reinvent the wheel
        - Comments help others so add comments to your scripts
    - Review examples in Help
- Test the script to remove any bugs

# Example API Script (First Lines)

```
'%RunPerInstance
'@
'@ DESCRIPTION
'@
'@
'@ SYNTAX
'@ TheFirstLines
'@
'@ PARAMETERS
'@ none
'@
'@ DEPENDENCIES/LIMITATIONS
'@ none
'@
'@ History
'@ Created MJJ 11/4/2015
'@@
Option Explicit
SetLocale("en-us")
Dim SynergyGetter
On Error Resume Next
Set SynergyGetter = GetObject(CreateObject("WScript.Shell").ExpandEnvironmentStrings("%SAInstance%"))
On Error GoTo 0
If (Not IsEmpty(SynergyGetter)) Then
Set Syn = SynergyGetter.GetSASynergy
Else
Set Syn = CreateObject("synergy.Synergy")
End If
'
' Put remainder of code here.
'
MsgBox "Script Complete"
Wscript.Quit
```

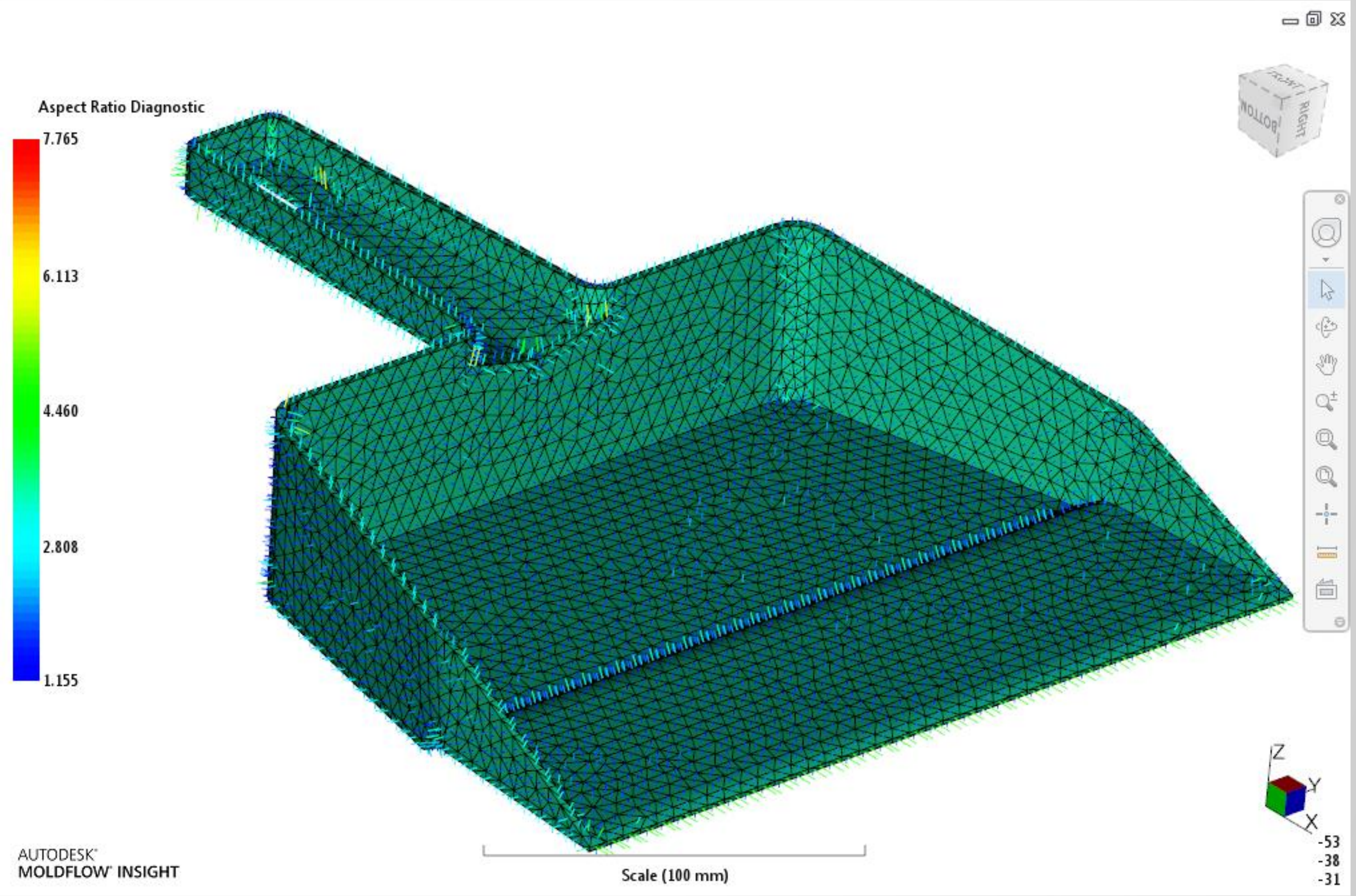


Script Complete

# Exercise 2: Custom Aspect Ratio

# Aspect Ratio

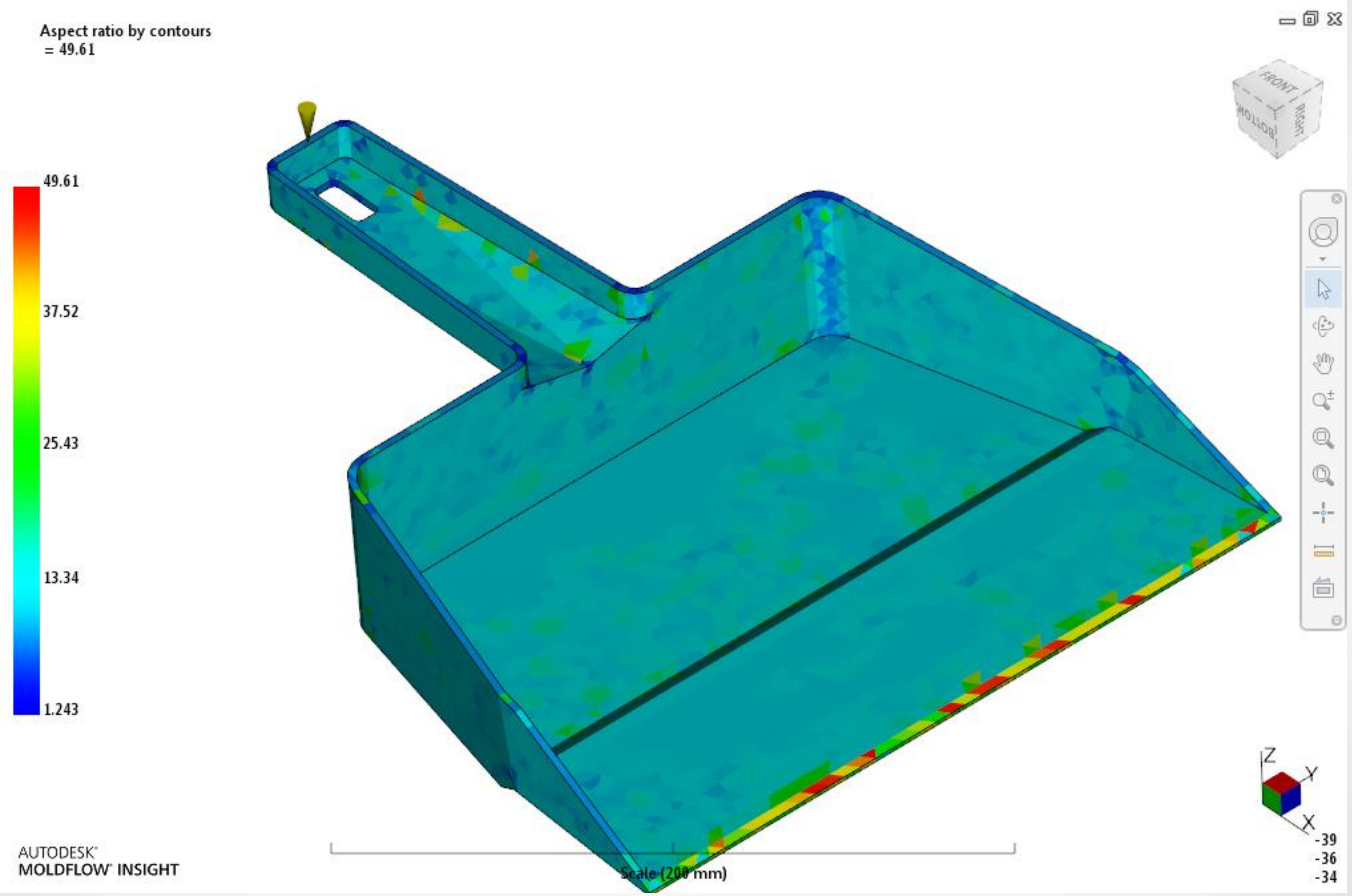
- Normally for MP or DD mesh
- Spikes or tensors
- What if we could display this result differently?
  - As contour plot of elements
  - On 3D mesh
- Let’s try on this exercise



# Workflow

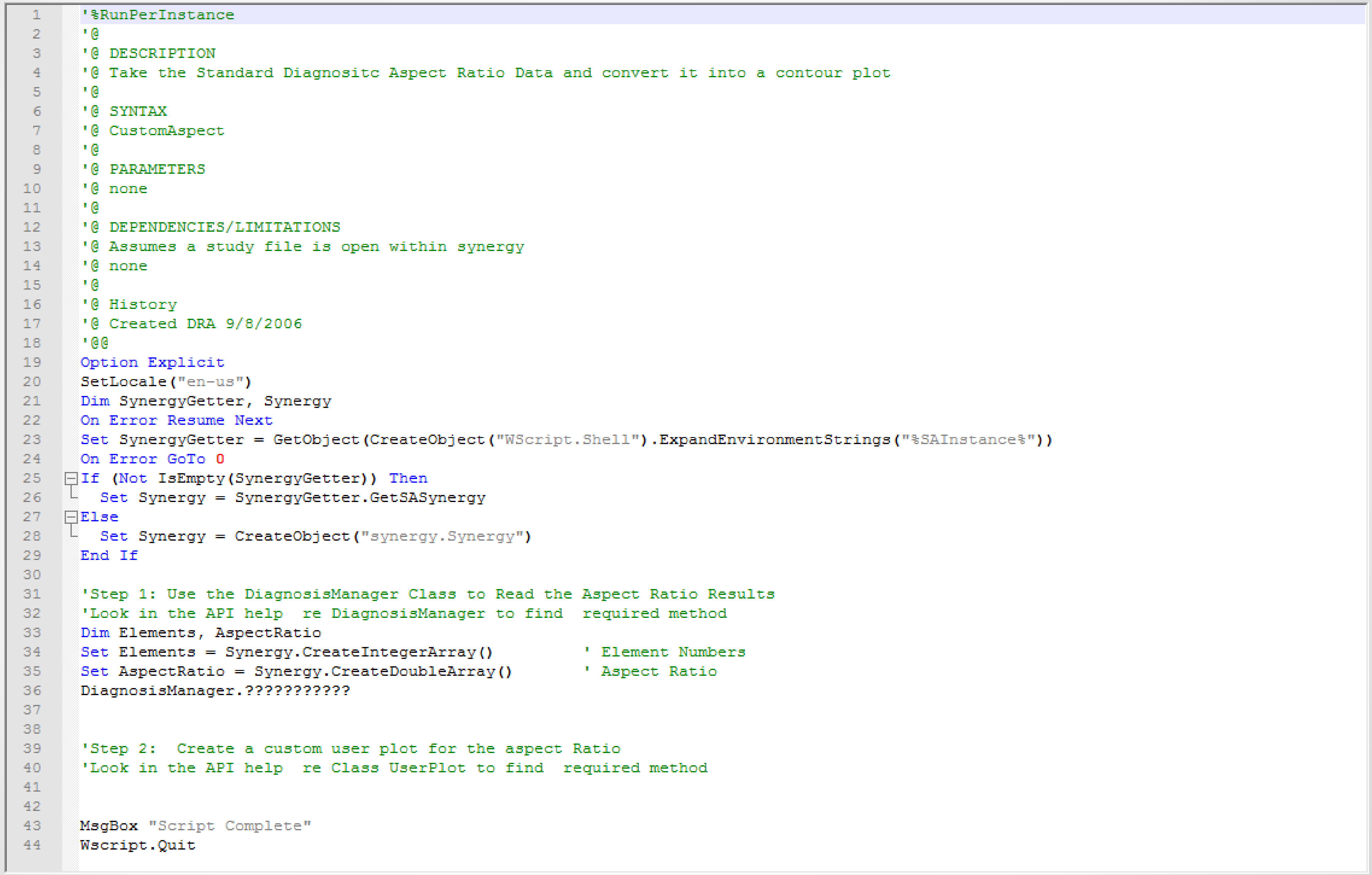
- Use EX2 files for this exercise
  - Located in C:\AU2015\API Training 1\Exercises\Exercise 2 Custom Results Aspect Ratio\EX2.mpi
- Open CustomAspect.vbs as a starting point in Notepad ++
- Step 1: Use API help to find DiagnosisManager Class nomenclature and finish script
- Step 2: Use API help UserPlot Class to finish script

# Custom Aspect Ratio Contour Plot



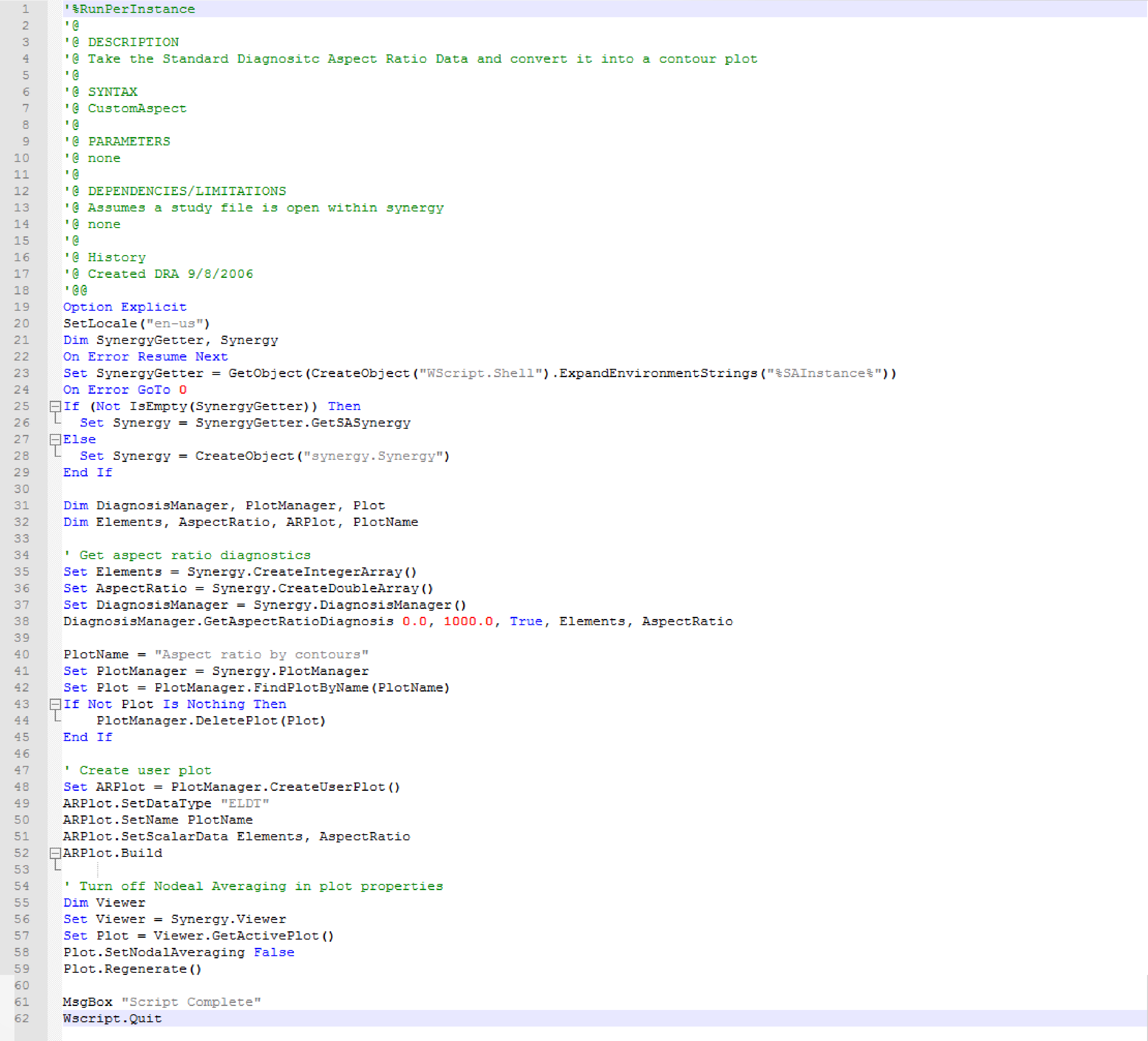
Aspect ratio by contours
= 49.61

# Incomplete Script – Research to Finish Script



```
'%RunPerInstance
'@
'@ DESCRIPTION
'@ Take the Standard Diagnositc Aspect Ratio Data and convert it into a contour plot
'@
'@ SYNTAX
'@ CustomAspect
'@
'@ PARAMETERS
'@ none
'@
'@ DEPENDENCIES/LIMITATIONS
'@ Assumes a study file is open within synergy
'@ none
'@
'@ History
'@ Created DRA 9/8/2006
'@@
Option Explicit
SetLocale("en-us")
Dim SynergyGetter, Synergy
On Error Resume Next
Set SynergyGetter = GetObject(CreateObject("WScript.Shell").ExpandEnvironmentStrings("%SAInstance%"))
On Error GoTo 0
If (Not IsEmpty(SynergyGetter)) Then
  Set Synergy = SynergyGetter.GetSASynergy
Else
  Set Synergy = CreateObject("synergy.Synergy")
End If

'Step 1: Use the DiagnosisManager Class to Read the Aspect Ratio Results
'Look in the API help  re DiagnosisManager to find  required method
Dim Elements, AspectRatio
Set Elements = Synergy.CreateIntegerArray()         ' Element Numbers
Set AspectRatio = Synergy.CreateDoubleArray()       ' Aspect Ratio
DiagnosisManager.???????????


'Step 2:  Create a custom user plot for the aspect Ratio
'Look in the API help  re Class UserPlot to find  required method


MsgBox "Script Complete"
Wscript.Quit
```

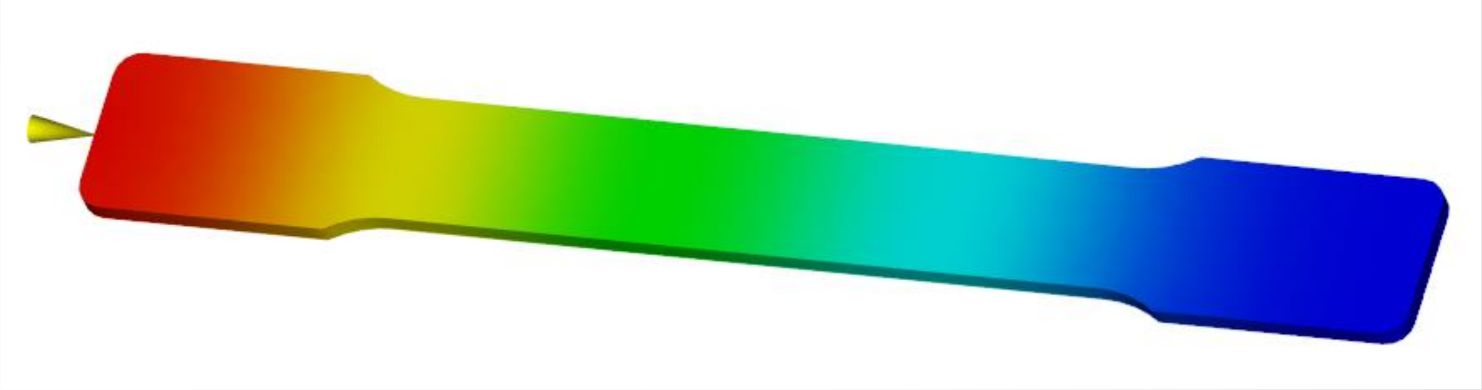
# One Solution…



```
'%RunPerInstance
'@
'@ DESCRIPTION
'@ Take the Standard Diagnositc Aspect Ratio Data and convert it into a contour plot
'@
'@ SYNTAX
'@ CustomAspect
'@
'@ PARAMETERS
'@ none
'@
'@ DEPENDENCIES/LIMITATIONS
'@ Assumes a study file is open within synergy
'@ none
'@
'@ History
'@ Created DRA 9/8/2006
'@@
Option Explicit
SetLocale("en-us")
Dim SynergyGetter, Synergy
On Error Resume Next
Set SynergyGetter = GetObject(CreateObject("WScript.Shell").ExpandEnvironmentStrings("%SAInstance%"))
On Error GoTo 0
If (Not IsEmpty(SynergyGetter)) Then
  Set Synergy = SynergyGetter.GetSASynergy
Else
  Set Synergy = CreateObject("synergy.Synergy")
End If

Dim DiagnosisManager, PlotManager, Plot
Dim Elements, AspectRatio, ARPlot, PlotName

' Get aspect ratio diagnostics
Set Elements = Synergy.CreateIntegerArray()
Set AspectRatio = Synergy.CreateDoubleArray()
Set DiagnosisManager = Synergy.DiagnosisManager()
DiagnosisManager.GetAspectRatioDiagnosis 0.0, 1000.0, True, Elements, AspectRatio

PlotName = "Aspect ratio by contours"
Set PlotManager = Synergy.PlotManager
Set Plot = PlotManager.FindPlotByName(PlotName)
If Not Plot Is Nothing Then
    PlotManager.DeletePlot(Plot)
End If

' Create user plot
Set ARPlot = PlotManager.CreateUserPlot()
ARPlot.SetDataType "ELDT"
ARPlot.SetName PlotName
ARPlot.SetScalarData Elements, AspectRatio
ARPlot.Build

' Turn off Nodeal Averaging in plot properties
Dim Viewer
Set Viewer = Synergy.Viewer
Set Plot = Viewer.GetActivePlot()
Plot.SetNodalAveraging False
Plot.Regenerate()

MsgBox "Script Complete"
Wscript.Quit
```

# Exercise 3: Exporting to Excel

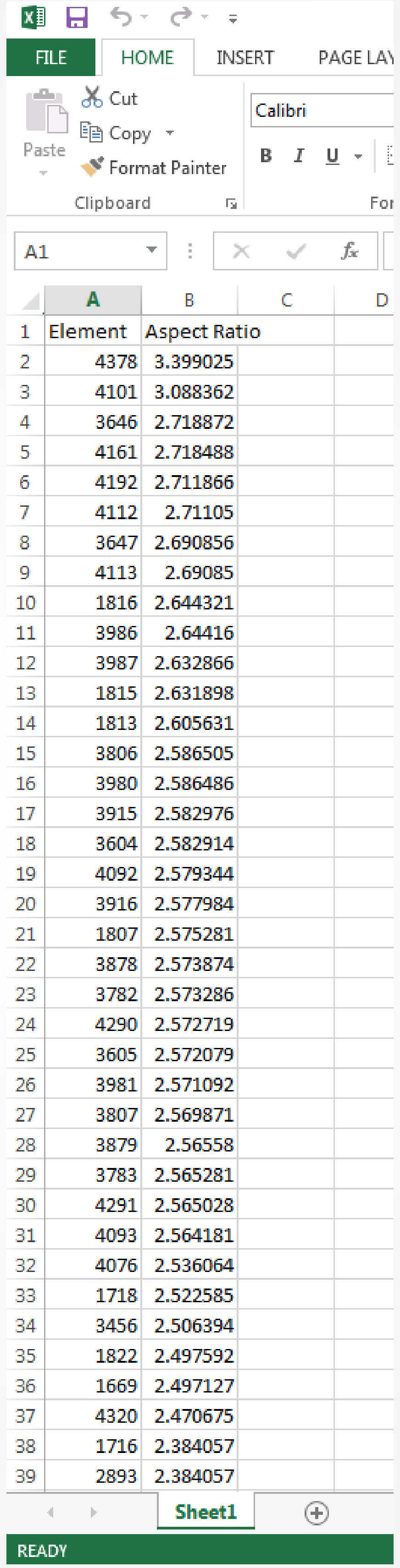
# Workflow



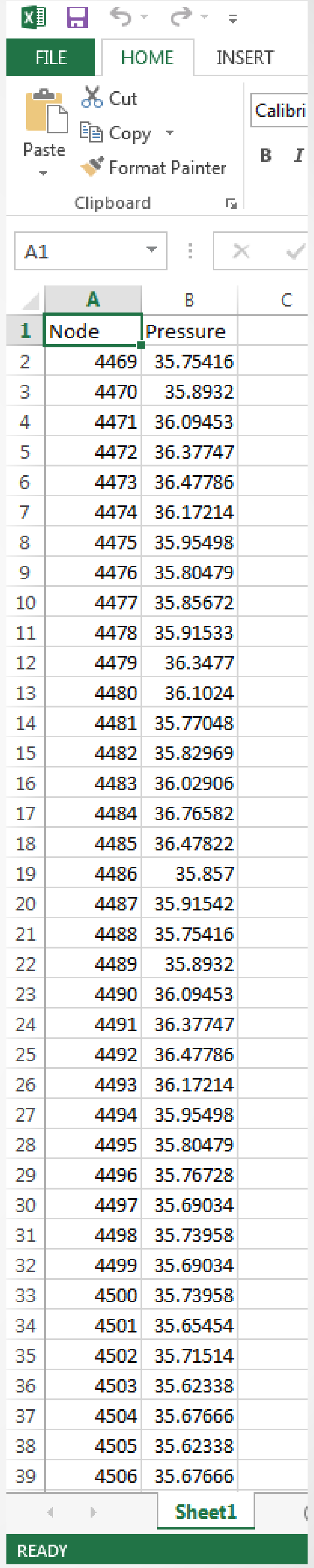
- Use EX3 files for this exercise
  - Located in C:\AU2015\API Training 1\Exercises\Exercise 3 Exporting to Excel\EX3.mpi
- Open ExportAspectToExcel.vbs as a starting point in Notepad ++
- Step 1: Use API help to find DiagnosisManager Class nomenclature and finish script. You can reuse your EX2 code
- Step 2: Review logic in how to export to Excel
- Run script to export results to Excel spreadsheet
  - Try to export aspect ratio (elements), pressure (nodes), temperature at flow front (nodes), etc

# Example Excel Export

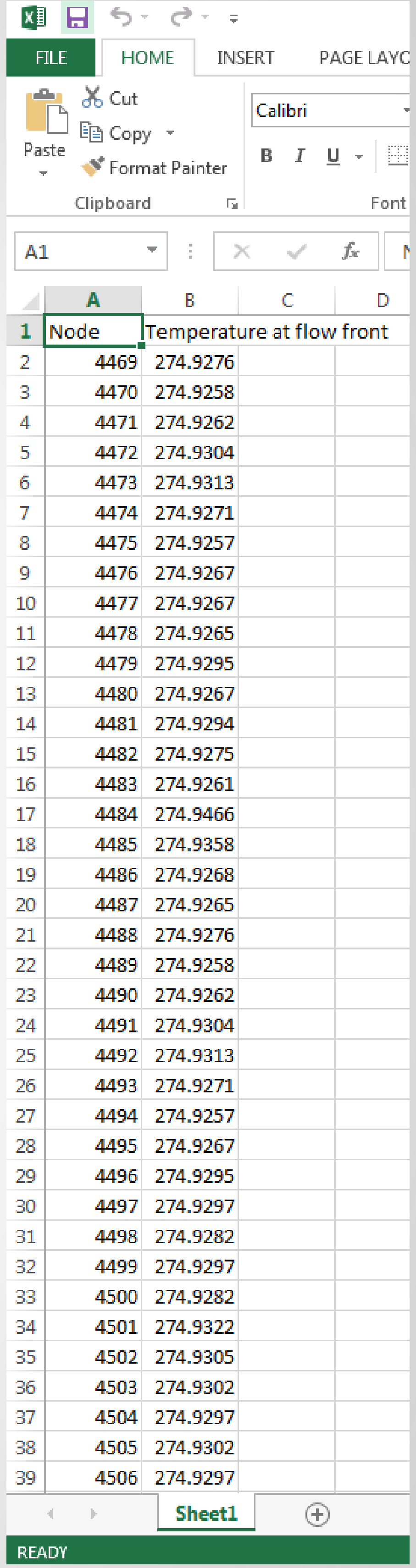
- Aspect ratio
  - Element list
- Pressure
  - Node list
- Temperature at flow front
  - Node list
- See Solutions folder if stuck



| | A | B | C | D |
|---|---|---|---|---|
| 1 | Element | Aspect Ratio | | |
| 2 | 4378 | 3.399025 | | |
| 3 | 4101 | 3.088362 | | |
| 4 | 3646 | 2.718872 | | |
| 5 | 4161 | 2.718488 | | |
| 6 | 4192 | 2.711866 | | |
| 7 | 4112 | 2.71105 | | |
| 8 | 3647 | 2.690856 | | |
| 9 | 4113 | 2.69085 | | |
| 10 | 1816 | 2.644321 | | |
| 11 | 3986 | 2.64416 | | |
| 12 | 3987 | 2.632866 | | |
| 13 | 1815 | 2.631898 | | |
| 14 | 1813 | 2.605631 | | |
| 15 | 3806 | 2.586505 | | |
| 16 | 3980 | 2.586486 | | |
| 17 | 3915 | 2.582976 | | |
| 18 | 3604 | 2.582914 | | |
| 19 | 4092 | 2.579344 | | |
| 20 | 3916 | 2.577984 | | |
| 21 | 1807 | 2.575281 | | |
| 22 | 3878 | 2.573874 | | |
| 23 | 3782 | 2.573286 | | |
| 24 | 4290 | 2.572719 | | |
| 25 | 3605 | 2.572079 | | |
| 26 | 3981 | 2.571092 | | |
| 27 | 3807 | 2.569871 | | |
| 28 | 3879 | 2.56558 | | |
| 29 | 3783 | 2.565281 | | |
| 30 | 4291 | 2.565028 | | |
| 31 | 4093 | 2.564181 | | |
| 32 | 4076 | 2.536064 | | |
| 33 | 1718 | 2.522585 | | |
| 34 | 3456 | 2.506394 | | |
| 35 | 1822 | 2.497592 | | |
| 36 | 1669 | 2.497127 | | |
| 37 | 4320 | 2.470675 | | |
| 38 | 1716 | 2.384057 | | |
| 39 | 2893 | 2.384057 | | |



| | A | B | C |
|---|---|---|---|
| 1 | Node | Pressure | |
| 2 | 4469 | 35.75416 | |
| 3 | 4470 | 35.8932 | |
| 4 | 4471 | 36.09453 | |
| 5 | 4472 | 36.37747 | |
| 6 | 4473 | 36.47786 | |
| 7 | 4474 | 36.17214 | |
| 8 | 4475 | 35.95498 | |
| 9 | 4476 | 35.80479 | |
| 10 | 4477 | 35.85672 | |
| 11 | 4478 | 35.91533 | |
| 12 | 4479 | 36.3477 | |
| 13 | 4480 | 36.1024 | |
| 14 | 4481 | 35.77048 | |
| 15 | 4482 | 35.82969 | |
| 16 | 4483 | 36.02906 | |
| 17 | 4484 | 36.76582 | |
| 18 | 4485 | 36.47822 | |
| 19 | 4486 | 35.857 | |
| 20 | 4487 | 35.91542 | |
| 21 | 4488 | 35.75416 | |
| 22 | 4489 | 35.8932 | |
| 23 | 4490 | 36.09453 | |
| 24 | 4491 | 36.37747 | |
| 25 | 4492 | 36.47786 | |
| 26 | 4493 | 36.17214 | |
| 27 | 4494 | 35.95498 | |
| 28 | 4495 | 35.80479 | |
| 29 | 4496 | 35.76728 | |
| 30 | 4497 | 35.69034 | |
| 31 | 4498 | 35.73958 | |
| 32 | 4499 | 35.69034 | |
| 33 | 4500 | 35.73958 | |
| 34 | 4501 | 35.65454 | |
| 35 | 4502 | 35.71514 | |
| 36 | 4503 | 35.62338 | |
| 37 | 4504 | 35.67666 | |
| 38 | 4505 | 35.62338 | |
| 39 | 4506 | 35.67666 | |



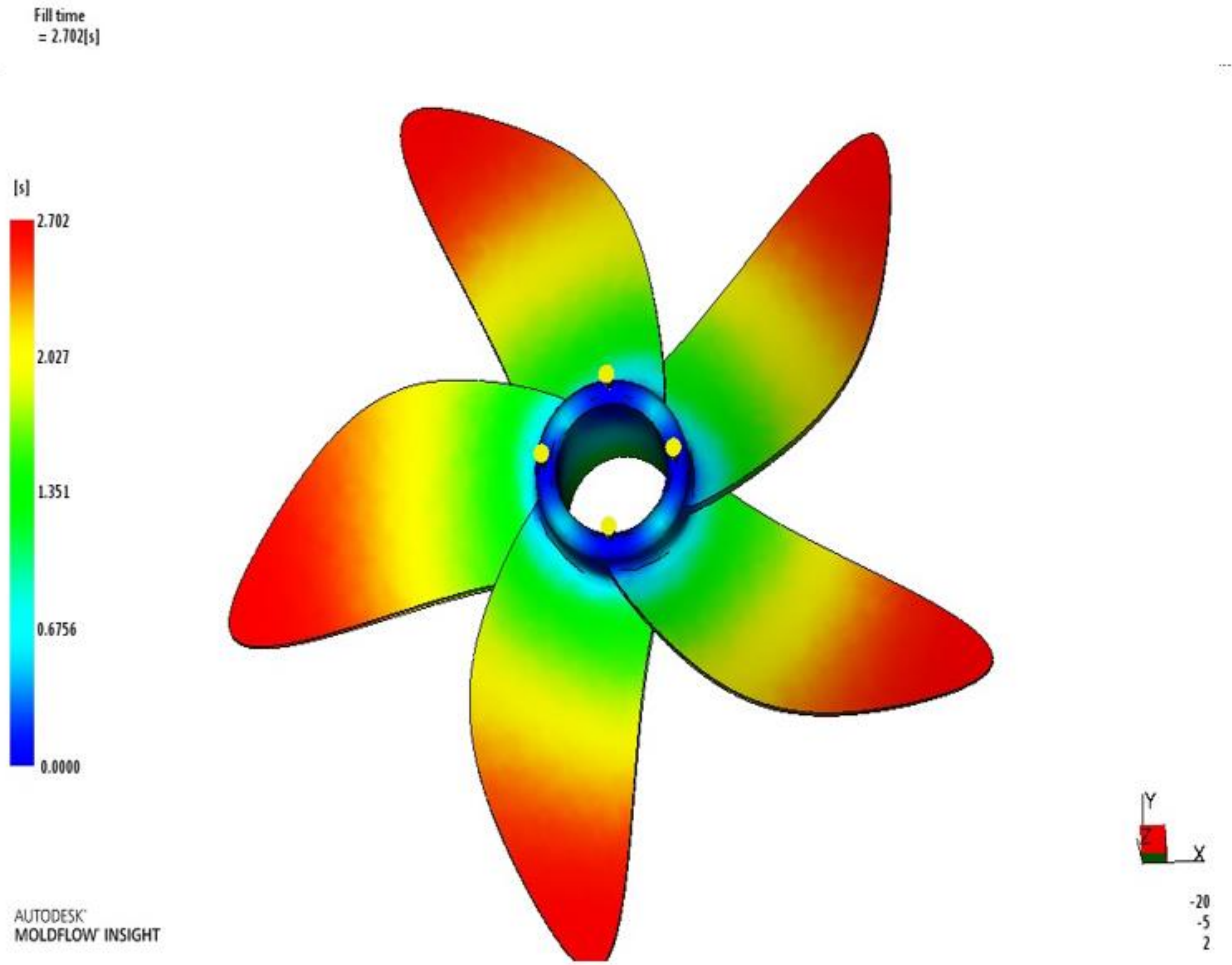
| | A | B | C | D |
|---|---|---|---|---|
| 1 | Node | Temperature at flow front | | |
| 2 | 4469 | 274.9276 | | |
| 3 | 4470 | 274.9258 | | |
| 4 | 4471 | 274.9262 | | |
| 5 | 4472 | 274.9304 | | |
| 6 | 4473 | 274.9313 | | |
| 7 | 4474 | 274.9271 | | |
| 8 | 4475 | 274.9257 | | |
| 9 | 4476 | 274.9267 | | |
| 10 | 4477 | 274.9267 | | |
| 11 | 4478 | 274.9265 | | |
| 12 | 4479 | 274.9295 | | |
| 13 | 4480 | 274.9267 | | |
| 14 | 4481 | 274.9294 | | |
| 15 | 4482 | 274.9275 | | |
| 16 | 4483 | 274.9261 | | |
| 17 | 4484 | 274.9466 | | |
| 18 | 4485 | 274.9358 | | |
| 19 | 4486 | 274.9268 | | |
| 20 | 4487 | 274.9265 | | |
| 21 | 4488 | 274.9276 | | |
| 22 | 4489 | 274.9258 | | |
| 23 | 4490 | 274.9262 | | |
| 24 | 4491 | 274.9304 | | |
| 25 | 4492 | 274.9313 | | |
| 26 | 4493 | 274.9271 | | |
| 27 | 4494 | 274.9257 | | |
| 28 | 4495 | 274.9267 | | |
| 29 | 4496 | 274.9295 | | |
| 30 | 4497 | 274.9297 | | |
| 31 | 4498 | 274.9282 | | |
| 32 | 4499 | 274.9297 | | |
| 33 | 4500 | 274.9282 | | |
| 34 | 4501 | 274.9322 | | |
| 35 | 4502 | 274.9305 | | |
| 36 | 4503 | 274.9302 | | |
| 37 | 4504 | 274.9297 | | |
| 38 | 4505 | 274.9302 | | |
| 39 | 4506 | 274.9297 | | |

# Exercise 4: Exporting to PPT

# Workflow

- Use EX4 files for this exercise
  - Located in C:\AU2015\API Training 1\Exercises\Exercise 4 Exporting to PPT\EX4.mpi
- Open ExportResultsToPPT.vbs in Notepad ++ and review script
  - Introduce Sub & Call, Function for temp folder path
- Run script to export results to PPT
  - Fill time and Pressure
  - Try to export other plots
  - See Solutions folder for other scripts for Histograms, multiple results on one page, etc

# Click to add title



# Useful Sample Scripts Folder Review

- CountPPMaterialData.vbs
- DeleteAllResults.vbs
- ExportSpecifiedToMFR.vbs
- MaterialDataHistogram.vbs
- ReadShearRateProfileData.vbs
- ReadTemperatureProfileData.vbs
- ReadViscosityProfileData.vbs
- RunAllResults.vbs
- ShowMaterialID.vbs
- ShowMaterialTsetAndSubID.vbs

ANY
QUESTIONS
?

# AU Answer Bar

AUTODESK UNIVERSITY 2015

Learn. Connect. Explore.

- Seek answers to all of your technical product questions by visiting the Answer Bar.
- Open daily 8am-6pm Tues-Wed and 8am-4pm Thurs
- Located just outside of Hall C on level 2.
- Staffed by Autodesk developers, QA, & support engineers ready to help you through your most challenging technical questions.

AUTODESK®