CP12451-L: The Autodesk Moldflow Synergy API Part 2: Building real world applications

David Astbury **Senior Manager, Solver Development**

Matthew Jaworski **Senior Tech. Sales Specialist, WWSS-AMER Territory Tech** Spec

Version: 3.1

Class summary

In this class we will:

- Show how to build practical API scripts using the API in the Moldflow Insight
- Introduce the API functionality and conventions
- Explain the data formats stored with the Study (.sdy) file
- Present a mixture of lectures and hands-on exercises to ensure that we cover the theoretical and practical aspects of building API scripts.

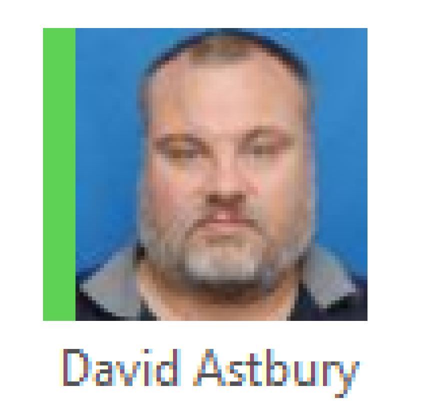
Some programming knowledge is preferred, but not required.

Key learning objectives

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

- Build real-world applications with the Autodesk Moldflow Synergy API
- Export Solver Output, Results and mesh information using the API
- Understand how data is stored in the Study (.sdy) file

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

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

House Keeping

Timetable

Time	Activity
1:00 - 1:15	Introduction
1:15 - 1:50	Result Data
1:50 - 2:00	Break
2:00 - 2:50	Screen Output Data
2:50 - 3:00	Break
3:00 - 3 :50	Understanding the Study File
3:50 - 4:00	Break
4:00 - 4:30	Understanding the Study File
4:30 - 5:00	Discussions / Question

Demographics

- Are you a Moldflow User ?
- Programmed with the Synergy API before ?

Standing Orders

- Ask questions at any time
  - We may refer it to a break if it’s complex/lengthy
- This is new course
  - It’s not perfect. Your feedback is appreciated
  - There is a lot of technical content in this course
    - You will not became and API expert in 4 hours
      - We aim to show you what’s possible and how to get started
      - Use these Notes with the sample scripts provided.
- About 30% of this course refers to features only available in the Moldflow 2017 release
  - I will point out this content

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!

Standing Orders

Have Fun!

After all, it is Las Vegas

Agenda

- Module: Result Data
  - Instruction plus 1 exercise
- Module: Screen Output Data
  - Instruction plus 1 exercise
- Module: Understanding the Study File
  - Instruction plus 1 exercise
- FAQ
- Sample Scripts/Questions/Discussion

Module: Result Data and the API

Module: Result Data and the API

- Scope
  - Understand the different result types
  - Show how to read results with the API
  - Show how to manipulate results with the API
  - Show how to create custom plots with the API
  - Show how to export results to PowerPoint

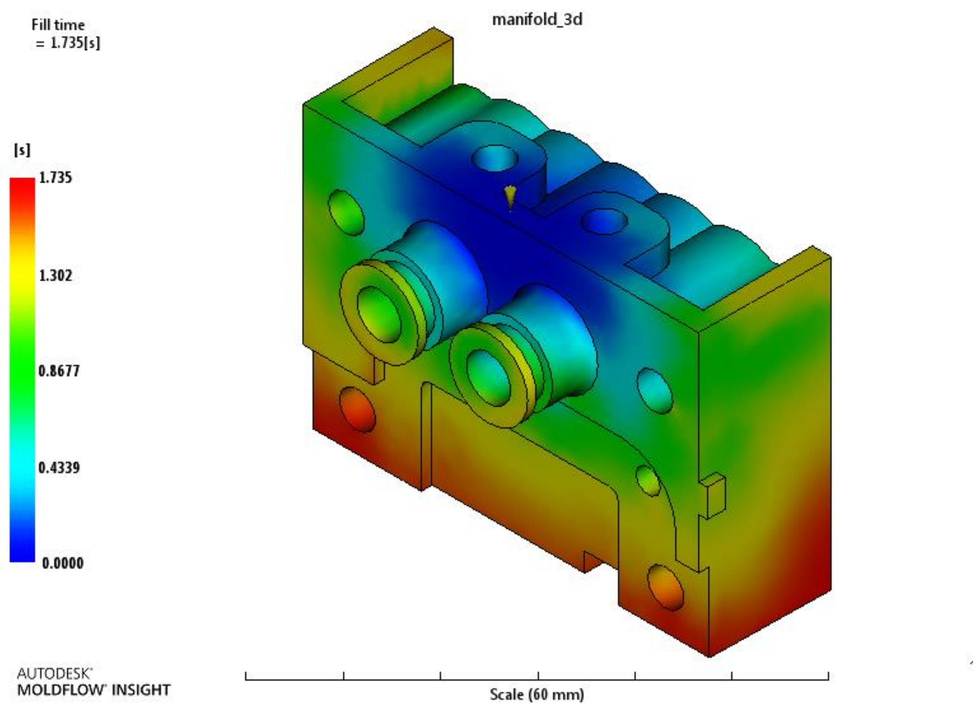
Result Data and the API

- When writing more advanced scripts you need a basic knowledge and understanding of the content of the following data files:
  - results.dat
  - units97.dat
  - tcodes.dat
  - tcodeset.dat
  - cmmesage.dat
  - process.dat

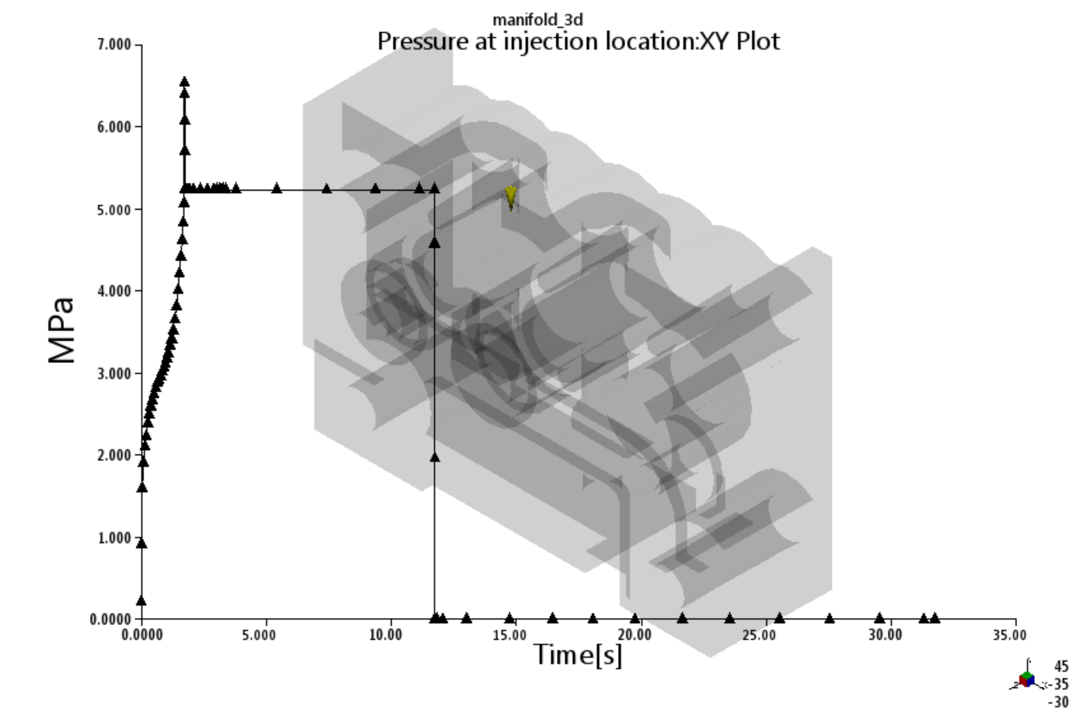
Understanding Key data files

- Location of these files
  - Installation directory
    - ….\Autodesk\Moldflow Synergy 20xx\data\dat
- WARNING
  - Editing these files changes the behaviour of the software
  - Editing these files may cause the software to crash

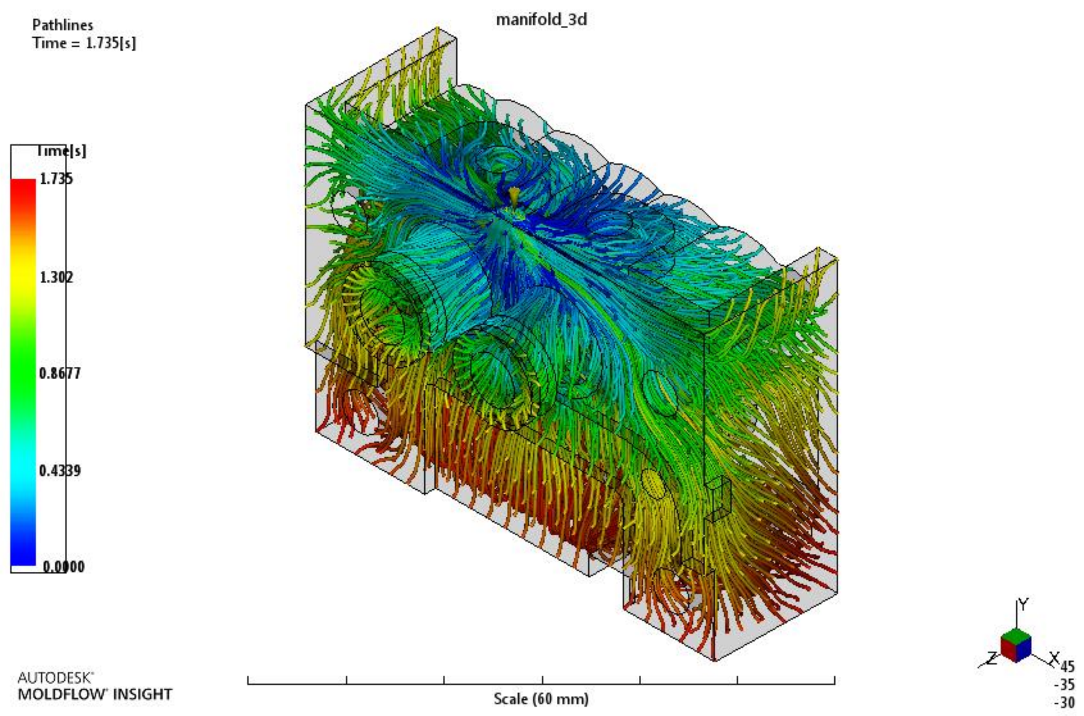
Results



Contour Plot



Time Series:XY Plot



Pathline Plot

Manipulating result data with the API

- Identifying a result

Result Name	“Fill time”	PlotManager.FindPlotByName(“Fill time”)
Result ID	1610	
Active plot		Viewer.GetActive()

Results.dat Edit

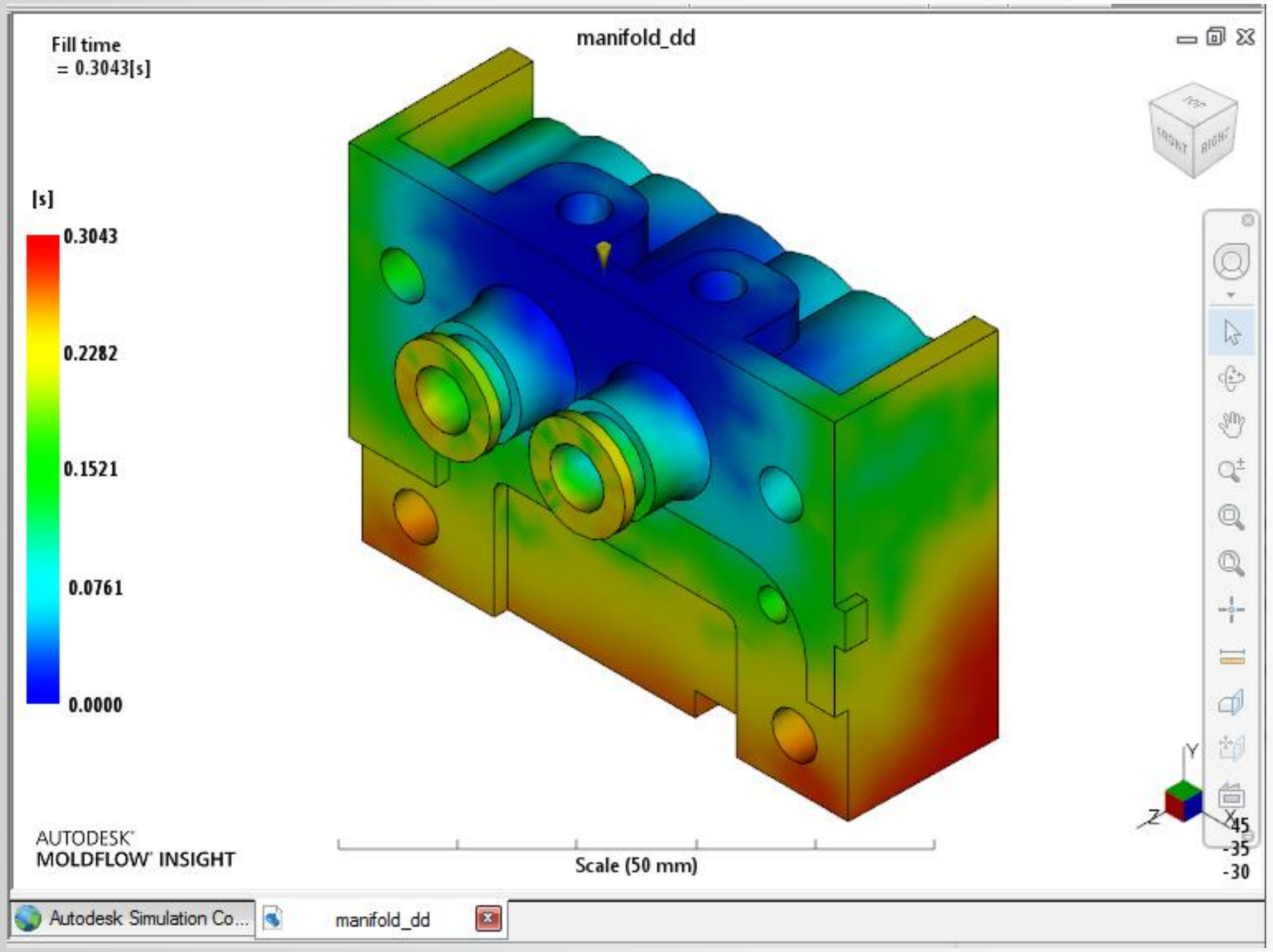
Manipulating result data with the API

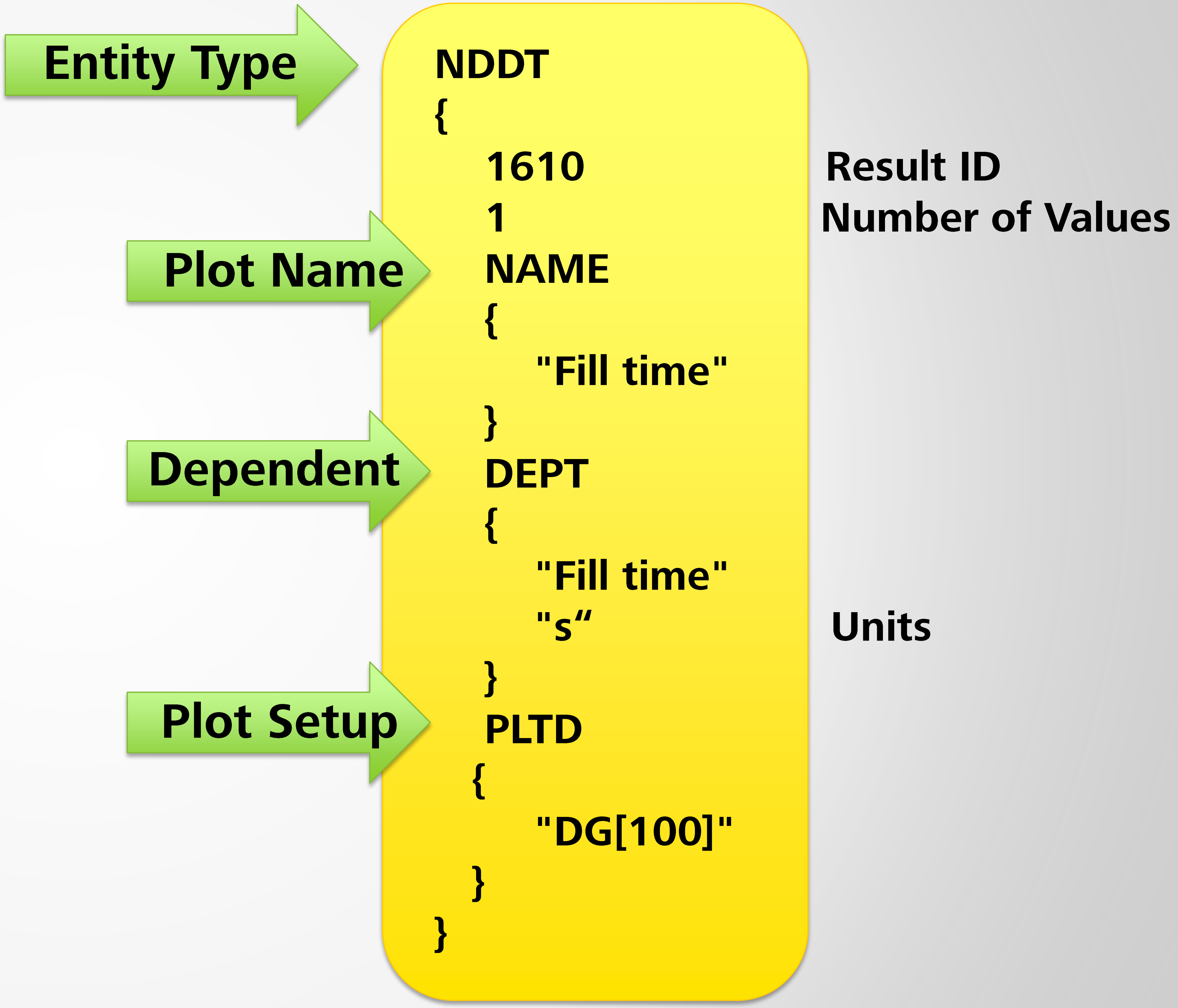
- Key Types of Results
  - NDDT = Nodal Data
  - ELDT = Elemental Data
  - TXDT = Text Data
  - NMDT = Non Mesh Data
  - Note: There are other types …. Not covered here

Results.dat Edit

Manipulating result data with the API

- NDDT
  - Fill time (Result ID = 1610)



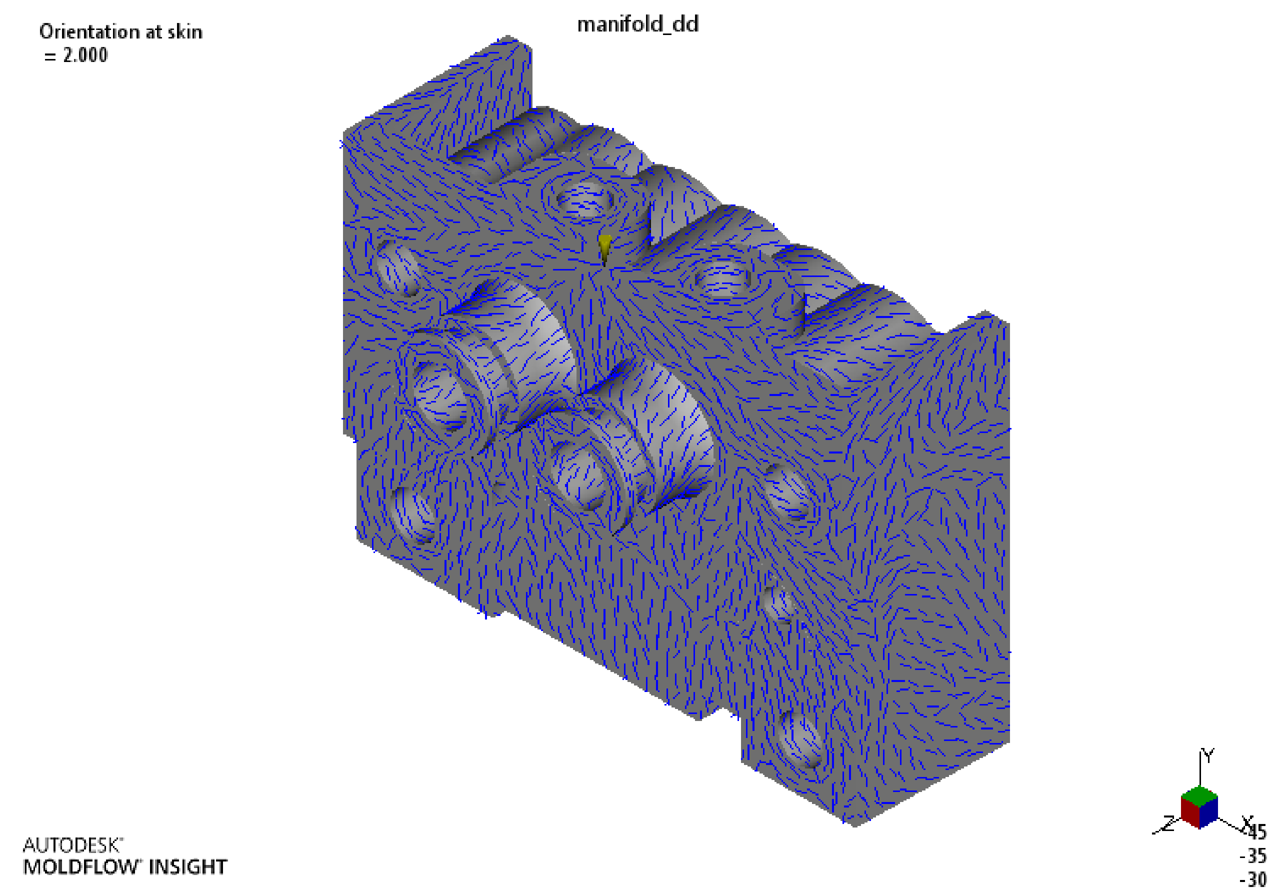


```
NDDT
{
   1610
   1
   NAME
   {
      "Fill time"
   }
   DEPT
   {
      "Fill time"
      "s“
   }
   PLTD
   {
      "DG[100]"
   }
}
```

Results.dat   Edit

Manipulating result data with the API

- ELDT
  - Orientation at Skin (Result ID = 1040)
    - Vector (3 Values X,Y,Z)
    - Dimensionless



```
ELDT
{
   1040
   3
   0
  NAME
  {
    "Orientation at skin"
  }
  DEPT
  {
     "Orientation at skin"
     ""
  }
  PLTD
 {
   "LDR[1,2,2]G[100]"
 }
}
```

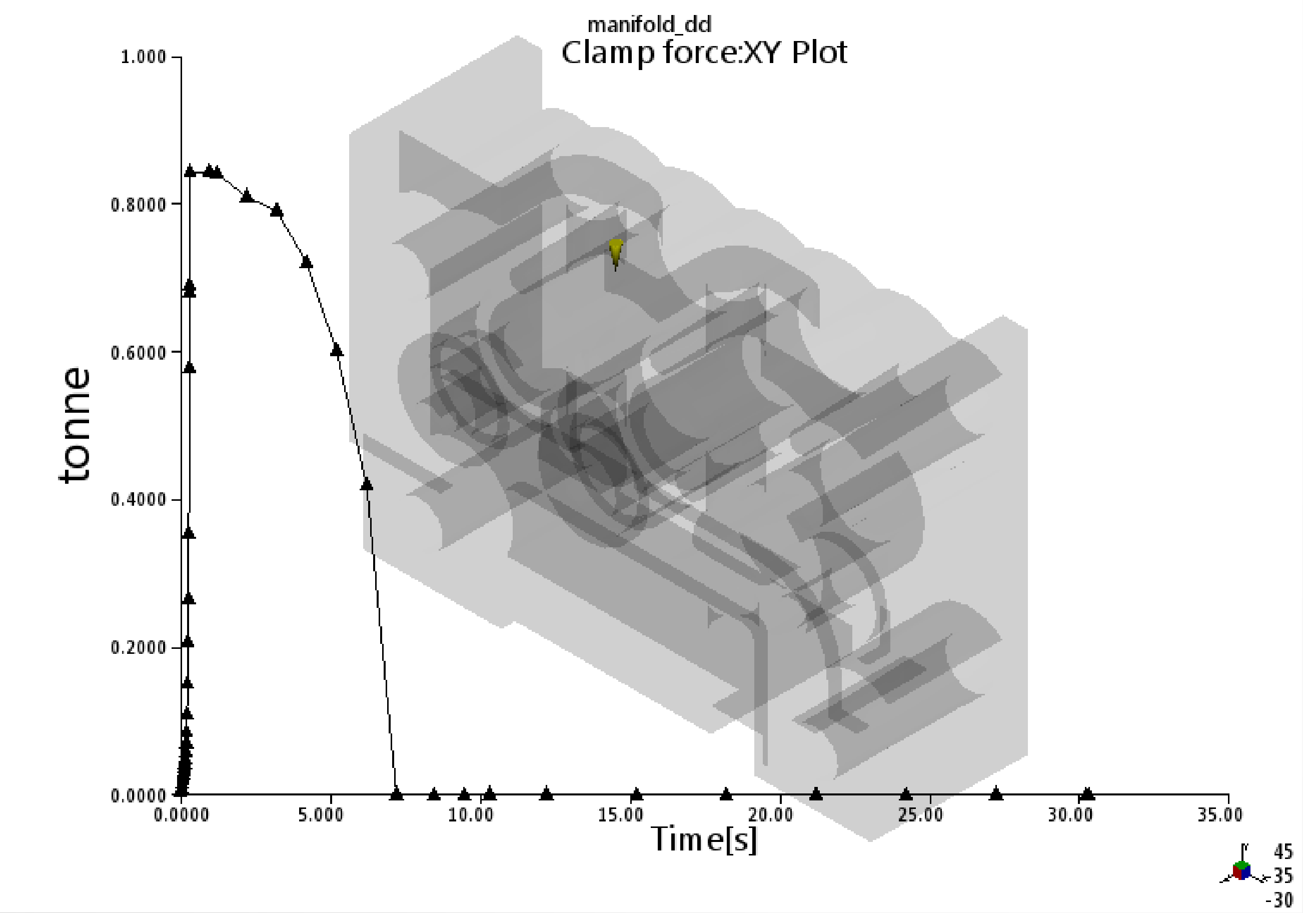
Vector

Dimensionless

Results.dat Edit

Manipulating result data with the API

- NMDT
  - Clamp Force (Result ID = 1150)
    - Time Based



Independent

```
NMDT
{
  1150
  1
  NAME
  {
      "Clamp force"
  }
  DEPT
  {
      "Clamp force"
      "N"
  }
  INDP
  {
      "Time"
      "s"
  }
  PLTD
  {
          "XG[100]"
  }
}
```

Results.dat Edit

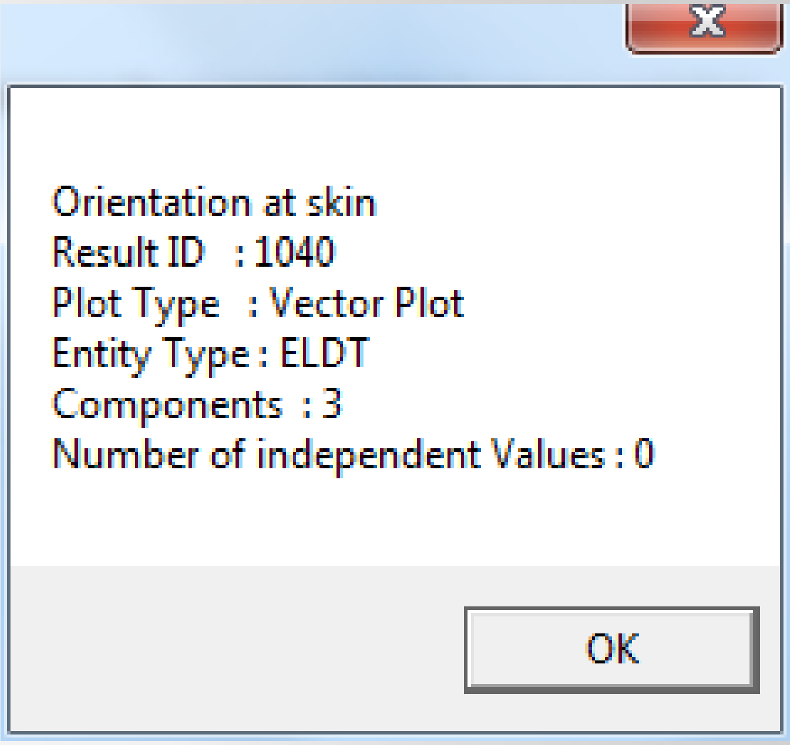
Manipulating result data with the API

Class	Purpose	Additional Comments
PlotManager	Find, Create/Delete, Enumerate Plots	Approximately 45 methods
Plot	Set and manipulate Plot attributes	Approximately 180 methods
UserPlot	Create user-defined Plots	Approximately 40 methods

Result Data and the API

- Script to Find Key Result Data

```
Set Viewer = Synergy.Viewer()
Set PlotManager = Synergy.PlotManager()
If Not Viewer Is Nothing then
        Set Plot = Viewer.ActivePlot()
        ID = Plot.GetDataID()
        MsgBox    Plot.GetName() & vbCrlf & _
                "Result ID   : " & ID & vbCrlf & _
                "Plot Type   : " & Plot.GetPlotType() & vbCrlf & _
                "Entity Type : " & Plot.GetDataType() & vbCrlf & _
                "Components : " & PlotManager.GetDataNbComponents(ID) & vbCrlf & _
                "Number of independent Values : " & Plot.GetNumberOfIndpVars()
    End If
End If
```

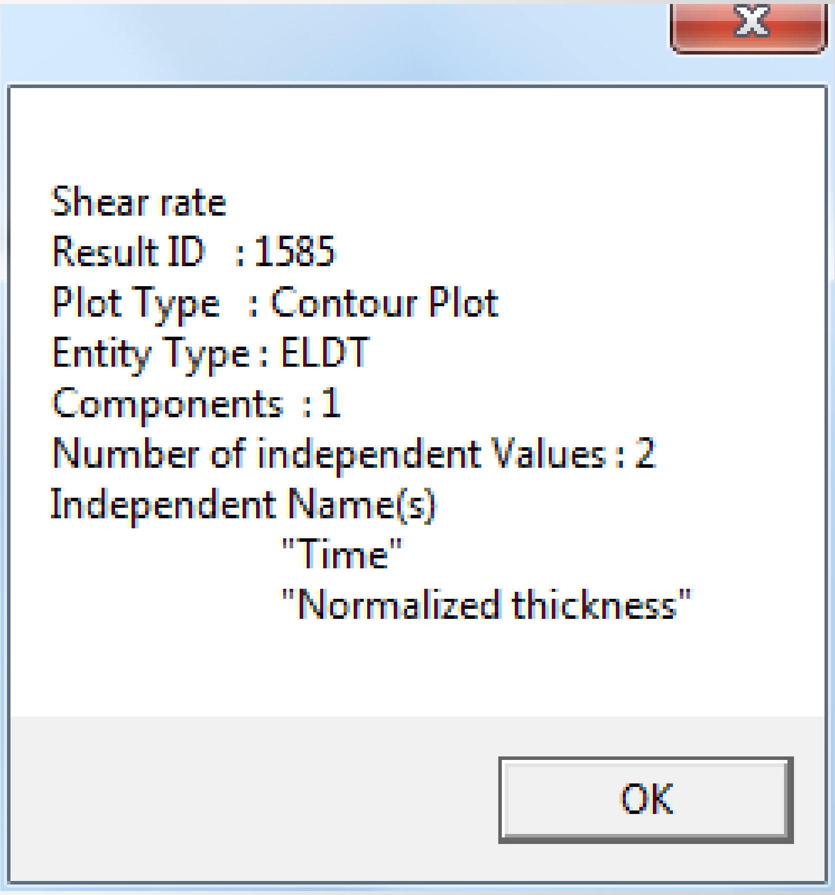


```
Orientation at skin
Result ID   : 1040
Plot Type   : Vector Plot
Entity Type : ELDT
Components  : 3
Number of independent Values : 0
```

ResultID.vbs Edit Execute

Result Data and the API

- Script to find key result data (a better version)
  - No Current API call to get the names of the Independent variable
    - If we know the format of the results.dat file
      - We can write vbscript to extract this information.



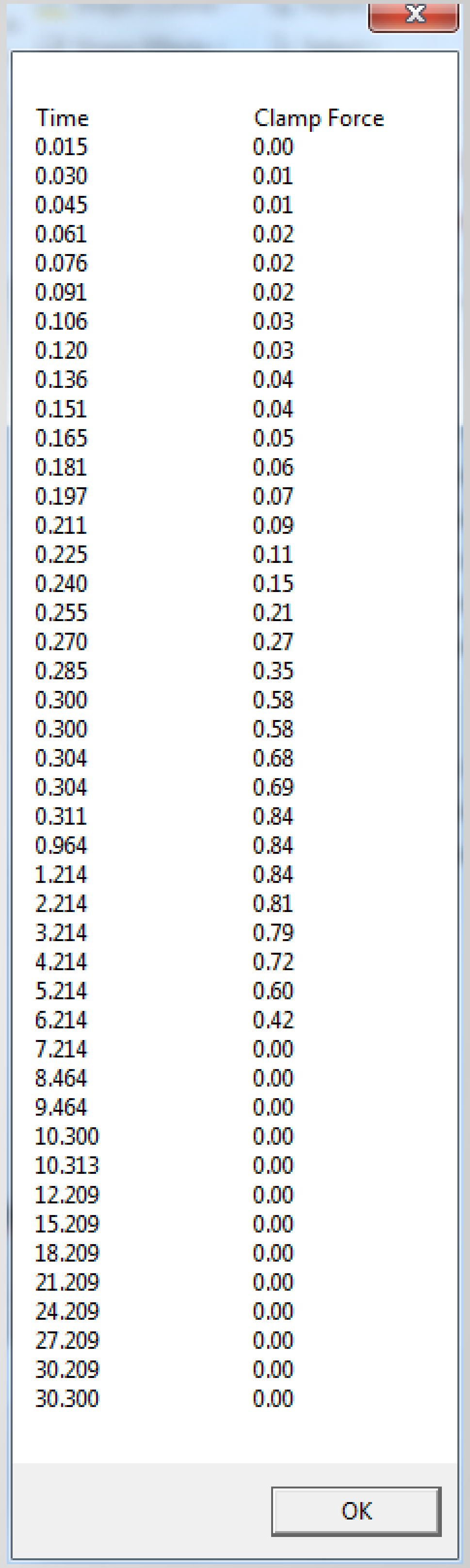
```
Shear rate
Result ID   : 1585
Plot Type   : Contour Plot
Entity Type : ELDT
Components  : 1
Number of independent Values : 2
Independent Name(s)
                "Time"
                "Normalized thickness"
```

ResultID2.vbs Edit Execute

Result Data and the API

- Read Clamp Force Data

```
Set Plot = PlotManager.FindPlotByName("Clamp force:XY Plot")
ID = Plot.GetDataID()
Set Independent = Synergy.CreateDoubleArray()
PlotManager.GetIndpValues ID, Independent
lMessage = "Time" & vbTab & "Clamp Force" &vbCrlf
For I = 0 To Independent.Size()-1
        ltime = Independent.Val(I)
        Set Indp = Synergy.CreateDoubleArray()
        Indp.AddDouble(ltime)
        Set values = Synergy.CreateDoubleArray()
        PlotManager.GetNonmeshData ID, Indp, values
        lMessage = lMessage & FormatNumber(ltime,3) vbTab & Formatnumber(values.val(0),2) & vbCrlf
Next
Msgbox lMessage
```

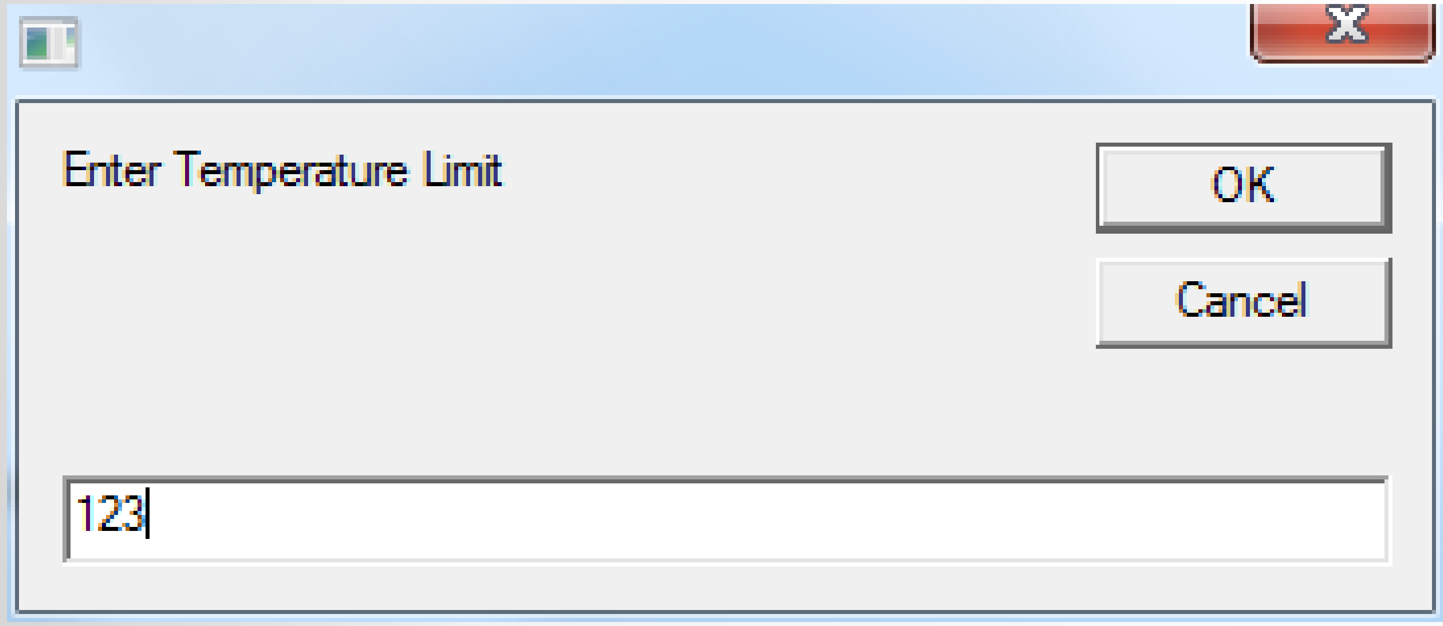


```
Time        Clamp Force
0.015       0.00
0.030       0.01
0.045       0.01
0.061       0.02
0.076       0.02
0.091       0.02
0.106       0.03
0.120       0.03
0.136       0.04
0.151       0.04
0.165       0.05
0.181       0.06
0.197       0.07
0.211       0.09
0.225       0.11
0.240       0.15
0.255       0.21
0.270       0.27
0.285       0.35
0.300       0.58
0.300       0.58
0.304       0.68
0.304       0.69
0.311       0.84
0.964       0.84
1.214       0.84
2.214       0.81
3.214       0.79
4.214       0.72
5.214       0.60
6.214       0.42
7.214       0.00
8.464       0.00
9.464       0.00
10.300      0.00
10.313      0.00
12.209      0.00
15.209      0.00
18.209      0.00
21.209      0.00
24.209      0.00
27.209      0.00
30.209      0.00
30.300      0.00
```

ReadClampForceData.vbs Edit Execute

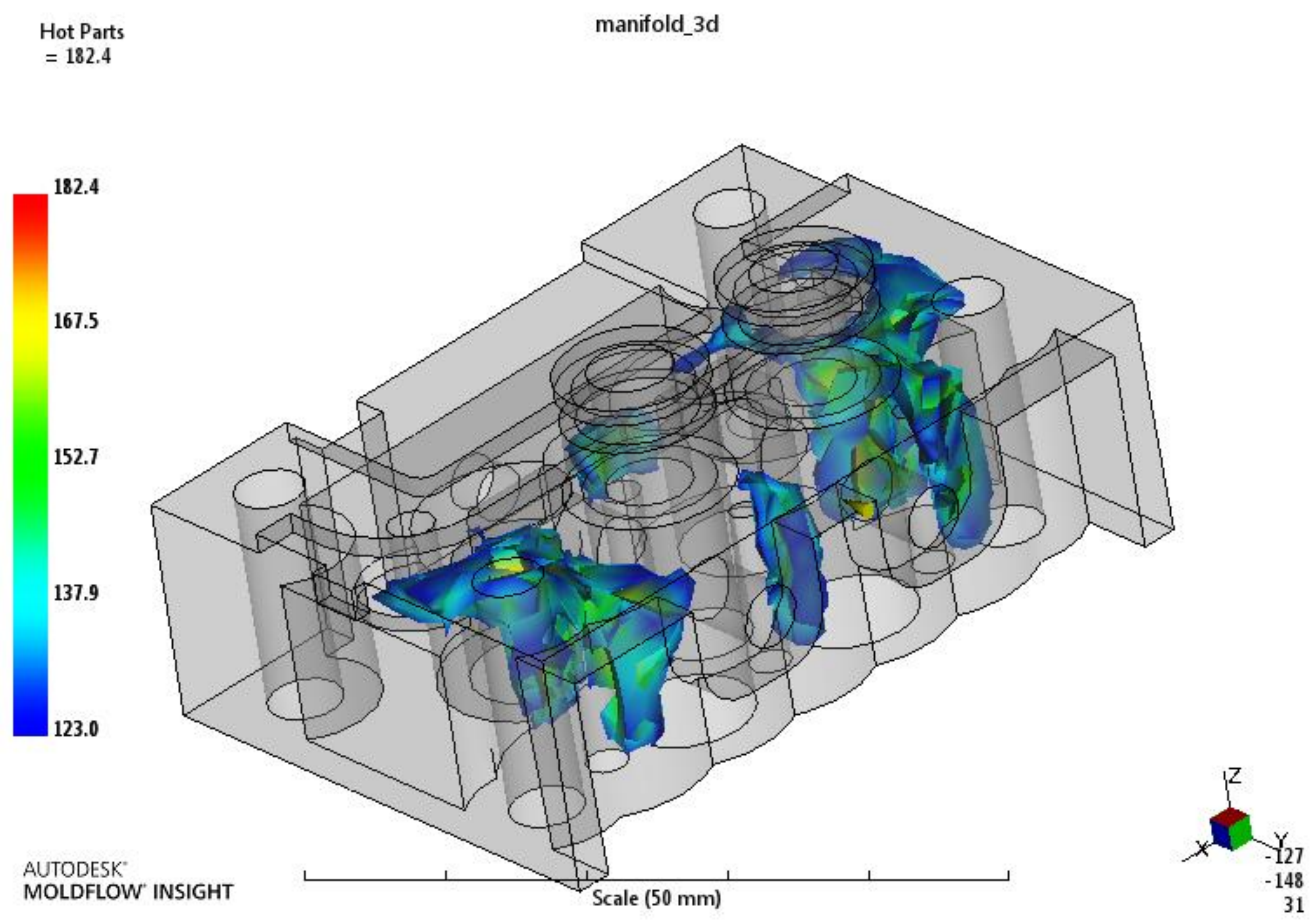
Result Data and the API

- Hot Area
  - Show all the temperatures above a define threshold



Enter Temperature Limit

123

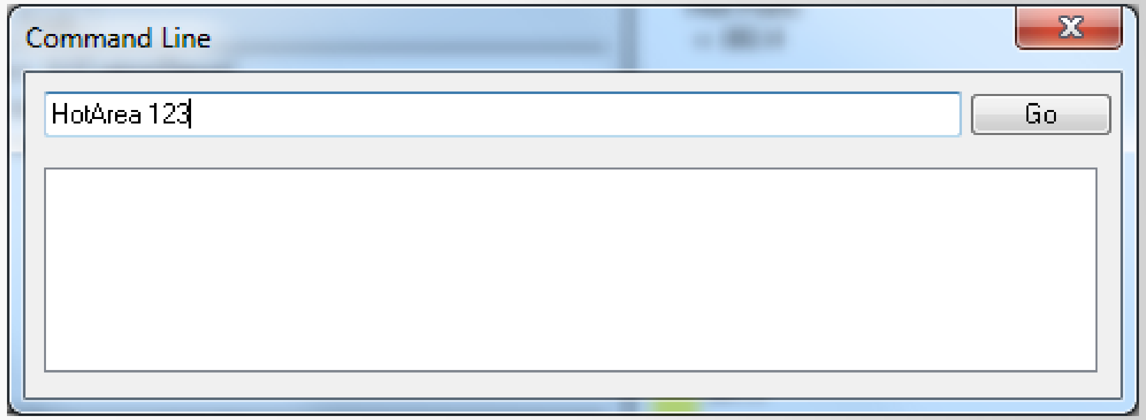


HotArea.vbs Edit Execute

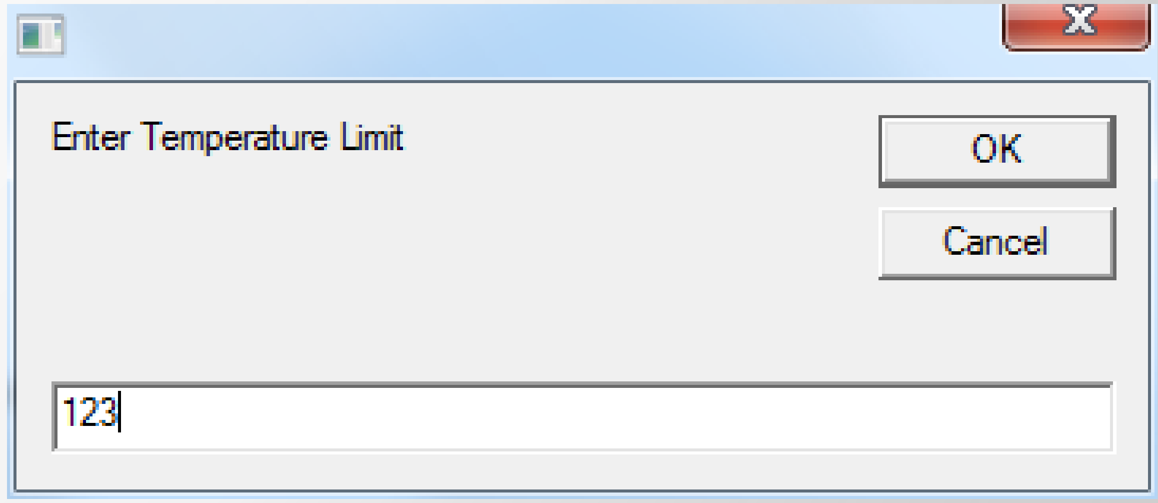
Result Data and the API

- Show all the temperatures above a threshold
  - Read inputs

```
Dim Args
Set Args = Wscript.Arguments
Dim MinTemperature
If Args.Count = 1 Then
        MinTemperature = Args(0)
Else
         MinTemperature = InputBox("Enter Temperature Limit ")
End If
```


Command Line

HotArea 123


Enter Temperature Limit

123

HotArea.vbs Edit Execute

Result Data and the API

- Show all the temperatures above a threshold
  - Read last set of Temperature Data

```
Set PlotMgr = Synergy.PlotManager
Set IndpValues = Synergy.CreateDoubleArray()
PlotMgr.GetIndpValues 1540, IndpValues

Set Indp = Synergy.CreateDoubleArray()
Indp.AddDouble(IndpValues.Val(IndpValues.Size()-1))
Set Arr = Synergy.CreateIntegerArray()
Set ArrResult = Synergy.CreateDoubleArray()
PlotMgr.GetScalarData 1540, Indp, Arr, ArrResult
```

HotArea.vbs Edit Execute

Result Data and the API

- Show all the temperatures above a threshold
  - Determine which nodes are above threshold

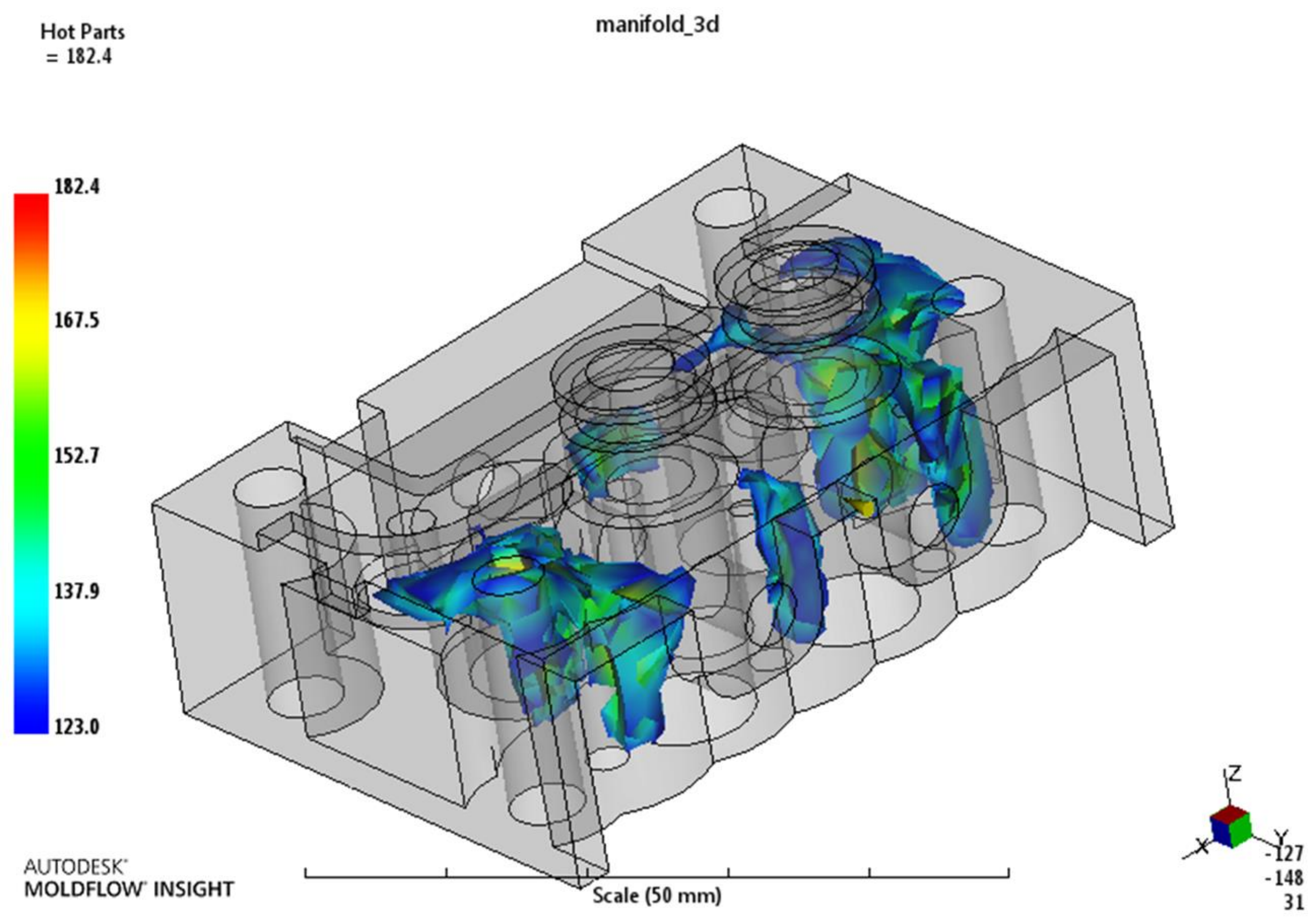
```
Set ResultID = Synergy.CreateIntegerArray()
Set ResultValue = Synergy.CreateDoubleArray()

For i = 0 To Arr.Size()-1
        lValue = ArrResult.Val(i)
        If (CDbl(lValue) > CDbl(MinTemperature)) Then
                ResultID.AddInteger(Arr.Val(I))
                ResultValue.AddDouble(lValue)
        End If
Next
```

HotArea.vbs Edit Execute

Result Data and the API

- Show all the temperatures above a threshold
  - Create Custom Plot “Hot Parts”

```
Set Plot = PlotMgr.FindPlotByName("Hot Parts")
If Not Plot Is Nothing Then
        PlotMgr.DeletePlot(Plot)
End If
If ResultID.Size() > 0 Then
        Set UserPlot = PlotMgr.CreateUserPlot()
        UserPlot.SetName("Hot Parts")
        UserPlot.SetDataType("NDDT")
        UserPlot.SetScalarData ResultID, ResultValue
        UserPlot.Build()
End if
```



HotArea.vbs Edit Execute

Exercise: Result Data and the API

End of Module: Result Data and the API

Module: Screen Output Data

Module: Result Data and the API

- Scope
  - Understand how the solver writes output data
  - Show how to read screen output data with the API
  - Show how to export screen output to PowerPoint

Module 2: Screen Output

- To work with Screen output in the API
  - A basic knowledge of the following data files is required:
    - cmmesage.dat
    - units97.dat

Understanding Units

- List of available unit systems
  - “Metric” or “Imperial”
  - List of unit descriptors
    - Will be in the API help
  - Unit conversion factors

- All data stored in the study and results and solver output is physically stored in SI units

Units97.dat Edit

Understanding Units

- In the Plot, PlotManager and UserPlot Classes
  - Numerical Data Input/Output is in the current Active Unit System
  - Synergy.GetUnits

    Returns “Metric” or “Imperial”

  - Synergy.SetUnits

    Synergy.SetUnits “Imperial” - Set units to Imperial

    Synergy.SetUnits “Metric” - Set units to Metric

  - Read Temperature result explicitly in Imperial Units

```
Synergy.SetUnits “Imperial”
PlotMgr.GetScalarData 1540, Indp, Arr, ArrResult
Synergy.SetUnits “Metric”
```

Units97.dat Edit

Understanding Units: units97.dat

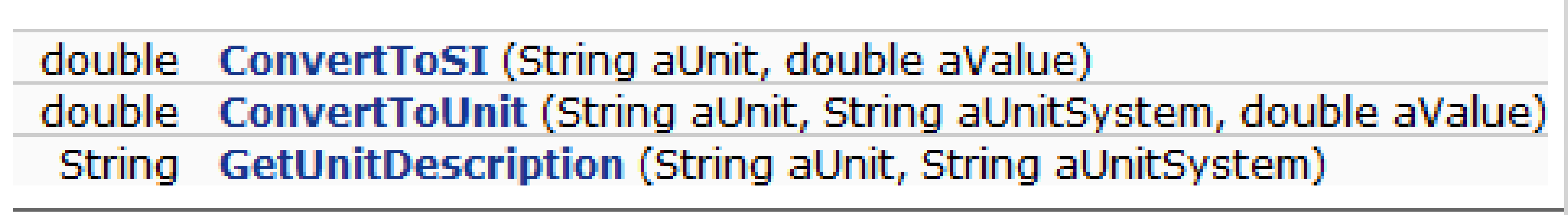
- Units97.dat contains
  - Available unit systems (USYS)
    - “Metric” or “Imperial”
  - List of unit descriptors (UNIT)
    - Visible Unit
    - SI unit
    - Unit conversion factors

```
USYS
{ "Metric"
   UNIT{ "mm" "m" 1.0e+03 0.0 }
   UNIT{ "MPa" "Pa" 1.0e-06 0.0 }
   UNIT{ "C" "K" 1.0 -273.15 }
   .....
 }
USYS
{ "English"
    UNIT{ "in" "m" 3.9370e+01 0.0 }
    UNIT{ "psi" "Pa" 1.4504e-04 0.0 }
    UNIT{ "F" "K" 1.8 -459.67 }
    .....
 }
```

Units97.dat Edit

Understanding Key data files: units97.dat

- Class: UnitConversion
  - Convert data between unit systems
  - Display unit system mnemonics

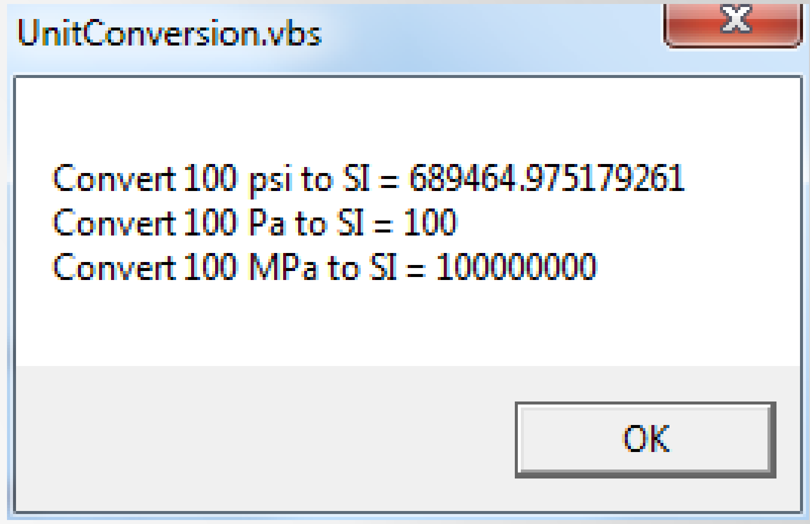


double	**ConvertToSI** (String aUnit, double aValue)
double	**ConvertToUnit** (String aUnit, String aUnitSystem, double aValue)
String	**GetUnitDescription** (String aUnit, String aUnitSystem)

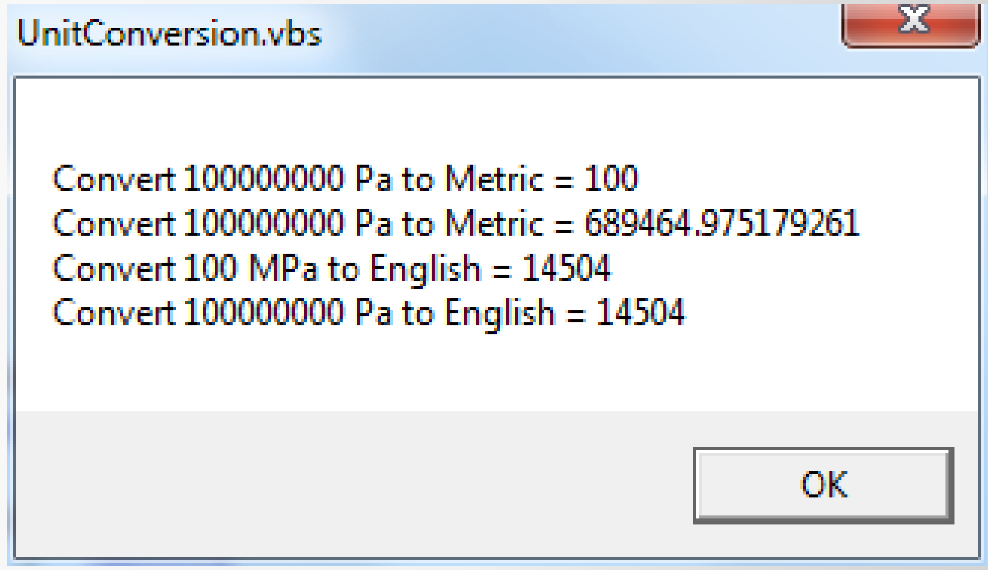
Understanding Key data files: units97.dat

- UnitConversion Example

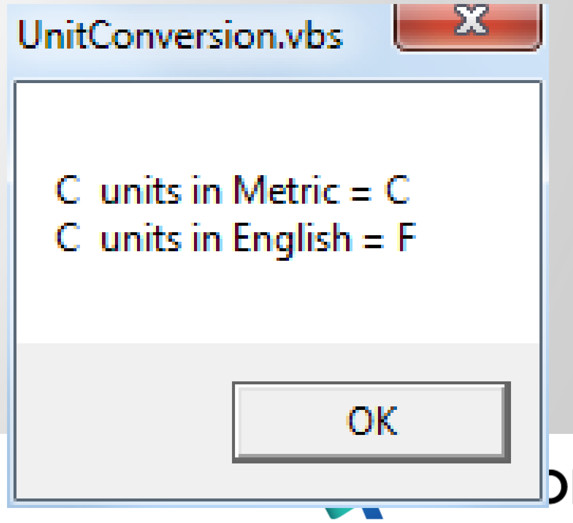
```
conv1 = UnitConversion.ConvertToSI("psi", 100)
lStr = "Convert 100 psi to SI = " & conv1 &  vbCrlf
conv1 = UnitConversion.ConvertToSI("Pa", 100)
lStr = lStr + "Convert 100 Pa to SI = " & conv1 &  vbCrlf
conv1 = UnitConversion.ConvertToSI("MPa", 100)
lStr = lStr + "Convert 100 MPa to SI = " & conv1
MsgBox lStr,,WScript.ScriptName

conv1 = UnitConversion.ConvertToUnit("Pa","Metric", 100000000)
lStr = "Convert 100000000 Pa to Metric = " & conv1 &  vbCrlf
conv1 = UnitConversion.ConvertToUnit("psi","Metric", 100000000)
lStr = lStr + "Convert 100000000 Pa to Metric = " & conv1 &  vbCrlf
conv1 = UnitConversion.ConvertToUnit("psi","English", 100000000)
lStr = lStr + "Convert 100 MPa to English = " & conv1 &  vbCrlf
conv1 = UnitConversion.ConvertToUnit("Pa","English", 100000000)
lStr = lStr + "Convert 100000000 Pa to English = " & conv1
MsgBox lStr,,WScript.ScriptName

units = UnitConversion.GetUnitDescription("C","Metric")
lStr = "C  units in Metric = " & units  & vbCrlf
units = UnitConversion.GetUnitDescription("C","English")
lStr = lStr + "C  units in English = " & units
MsgBox lStr,,WScript.ScriptName
```



```
Convert 100 psi to SI = 689464.975179261
Convert 100 Pa to SI = 100
Convert 100 MPa to SI = 100000000
```



```
Convert 100000000 Pa to Metric = 100
Convert 100000000 Pa to Metric = 689464.975179261
Convert 100 MPa to English = 14504
Convert 100000000 Pa to English = 14504
```



```
C  units in Metric = C
C  units in English = F
```

UnitConversion.vbs   Edit   Execute

Solver Output Data

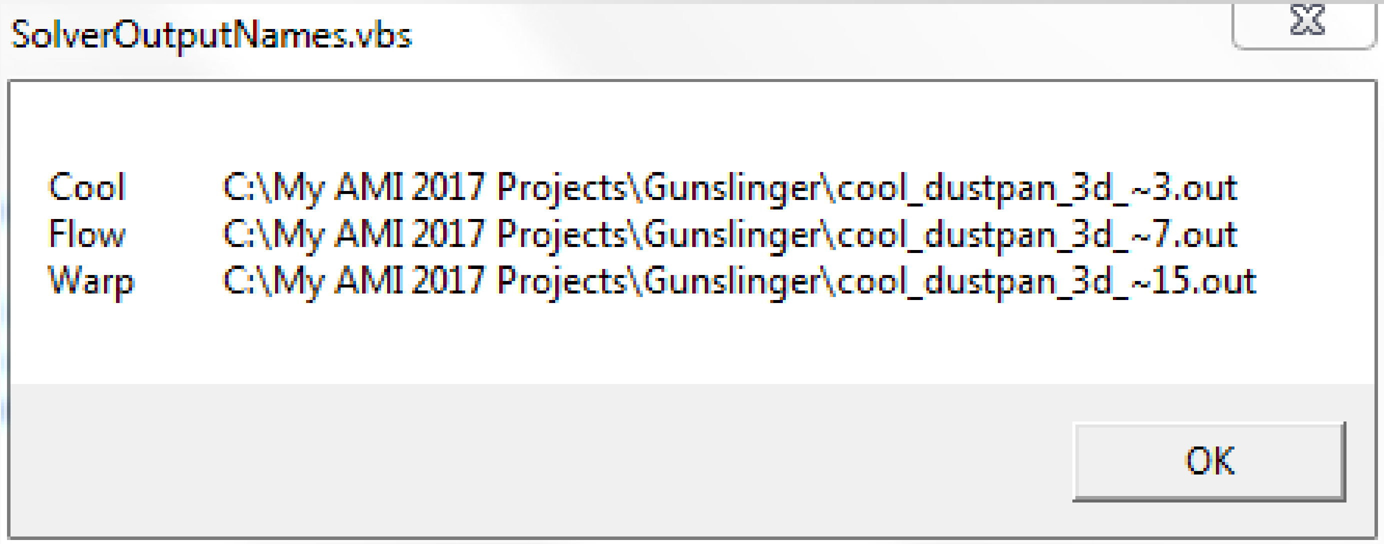
- How solvers write output data
  - Each solver writes screen and summary output to a ***.out*** file

    model~1.out

  - If there is a sequence, one output file is written for each stage in the sequence

  - The ***.err*** file which contains warning and error messages uses the same format as the ***.out*** file

Solver Output Data



SolverOutputNames.vbs

```
Cool	C:\My AMI 2017 Projects\Gunslinger\cool_dustpan_3d_~3.out
Flow	C:\My AMI 2017 Projects\Gunslinger\cool_dustpan_3d_~7.out
Warp	C:\My AMI 2017 Projects\Gunslinger\cool_dustpan_3d_~15.out
```

- How solvers write output data

```
lStr = ""
For I = 0 To StudyDoc.NumberOfAnalyses() - 1
	AnalysisName = StudyDoc.AnalysisName(I)
	lResultPrefix = StudyDoc.GetResultPrefix(AnalysisName)
	lOutFileName = Project.Path & lResultPrefix & ".out"
	lStr = lStr + AnalysisName & vbTab & lOutFileName & vbCrlf
Next
MsgBox lStr,,WScript.ScriptName
```

SolverOutputNames.vbs Edit Execute

Solver Output Data

- How solvers write output data
  - Message Block

    MSCD

    Number of Strings

    Number of Numbers

    <number>

    …

    < number>

```
….
1313
0
7
0.3511
55737.105
1.41469e+013
233.14999
0
27.202999
51.599998
….
```

Model~1.out Edit

Solver Output Data

- Processing Screen output
  - MSCD 1313 (Viscosity Model)

solver~1.out

```
..
1313
0
7
0.3511
55737.105
1.41469e+013
233.14999
0
27.202999
51.599998
..
```

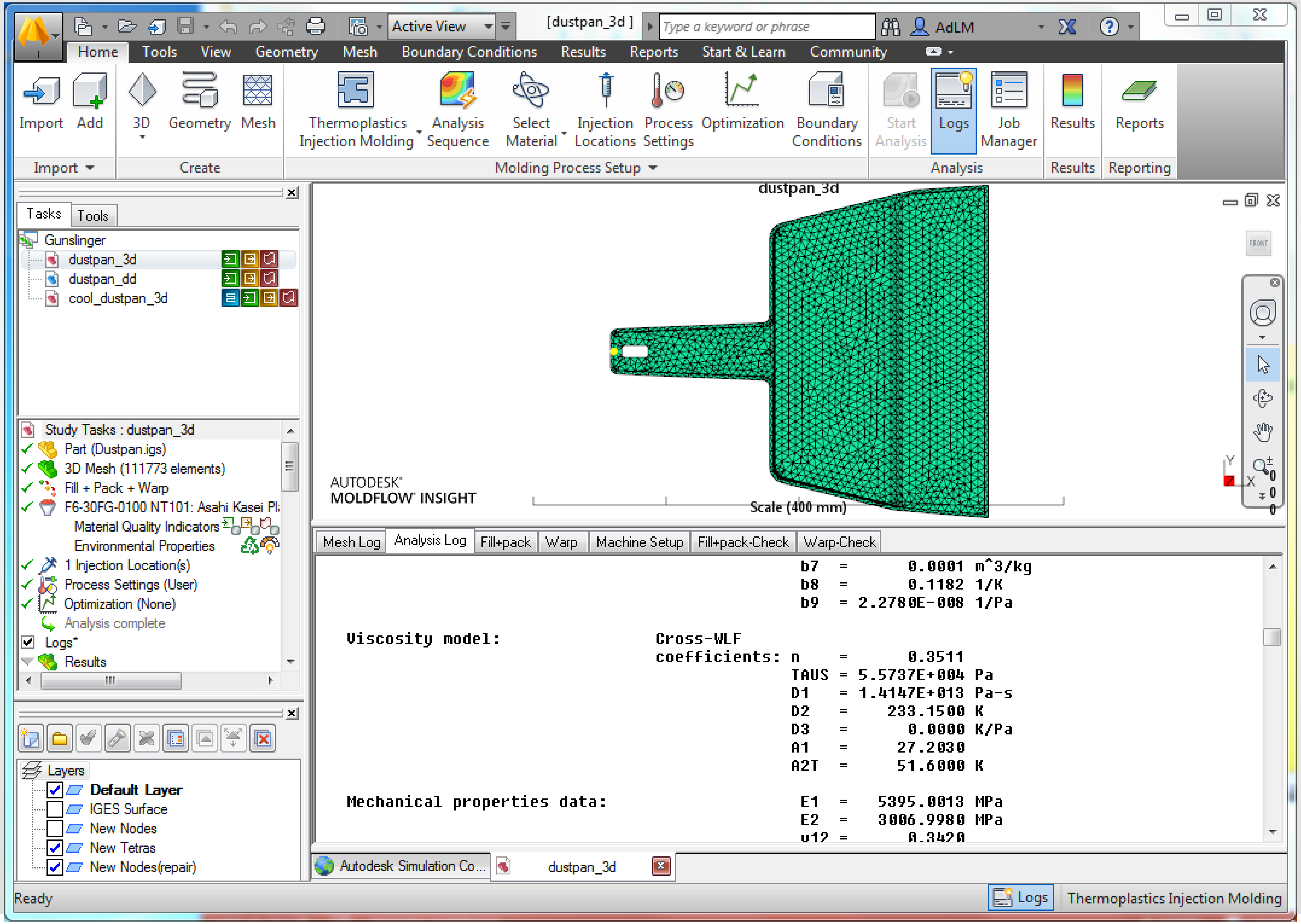
+

cmmesage.dat

```
..
MSCD 1313 9 0 0 0 0 0 5

     Viscosity model:              Cross-WLF
                             coefficients: n    = %11.4G
                                        TAUS = %11.4G
                                        D1   = %11.4G
                                        D2   = %11.4G
                                        D3   = %11.4G
                                        A1   = %11.4G
                                        A2T  = %11.4G
   ,0,1
   Pa,0,2
   Pa-s,0,3
   K,0,4
   K/Pa,0,5
   ,0,6
   K,0,7
..
```

solver~1.out Edit

Solver Output Data



```
                                            b7   =       0.0001 m^3/kg
                                            b8   =       0.1182 1/K
                                            b9   = 2.2780E-008 1/Pa

   Viscosity model:                Cross-WLF
                                   coefficients: n    =       0.3511
                                            TAUS = 5.5737E+004 Pa
                                            D1   = 1.4147E+013 Pa-s
                                            D2   =     233.1500 K
                                            D3   =       0.0000 K/Pa
                                            A1   =      27.2030
                                            A2T  =      51.6000 K

   Mechanical properties data:              E1   =   5395.0013 MPa
                                            E2   =   3006.9980 MPa
                                            υ12  =       0.3420
```

Understanding Key data files: cmmesage.dat

- cmmesage.dat contains:
  - The definition and formatting for all solver output
  - The unit definition and descriptions

Understanding Key data files: cmmesage.dat

- MSCD (Message Code) Definition

```
----   MSCD  Icode Hlin Rlin Tlin Idec Idiv Istr mFlag
----       Text message with variable formats %11.4G
----       Unit,C_pass,A_pos,Incr(for repeating lines only),DataType
----
```

Cmmesage.dat

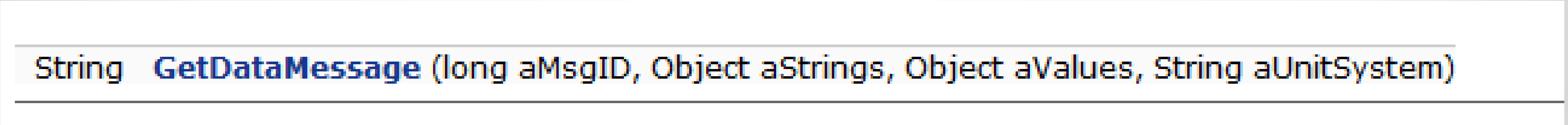
Understanding Key data files: cmmesage.dat

```
MSCD -  (appears as is) stands for Message code
----   Icode    - message number
----   Hlin     -  number of head lines (lines in message before starting repeating lines), must be >0
----   Rlin     -  number of lines in message data file that are to be printed again and again.
----   Tlin     -  number of tail lines (lines of text after repeating lines)
----   Idec     -  position in the array to start the values for the repeating lines
----   Idiv     -  the number of values in each repeating block
----   Istr     -  the number of strings in each block
----  Unit      -  Pa,s,K,% unit to be printed after value
----   C_pass - flag: 1 if the units and number should be passed on screen. UI will convert units if possible
----              flag: 2 convert into the units but not to display the units
----   A_pos  - position in the value array for the variable
----   Incr     - (repeating lines only, otherwise blank) number of array positions between values of the ----
----               same field on adjacent repeating lines
----   DataType       - flag: 0- default, 1 - node , 2 element(beam/tri/tetra/wedge)
----                     this flag is used to filter elements in Analysis Log
```

Cmmesage.dat   Edit

Understanding Key data files: cmmesage.dat

- Class: SystemMessage
  - New in 2017 Release
  - 1 Method



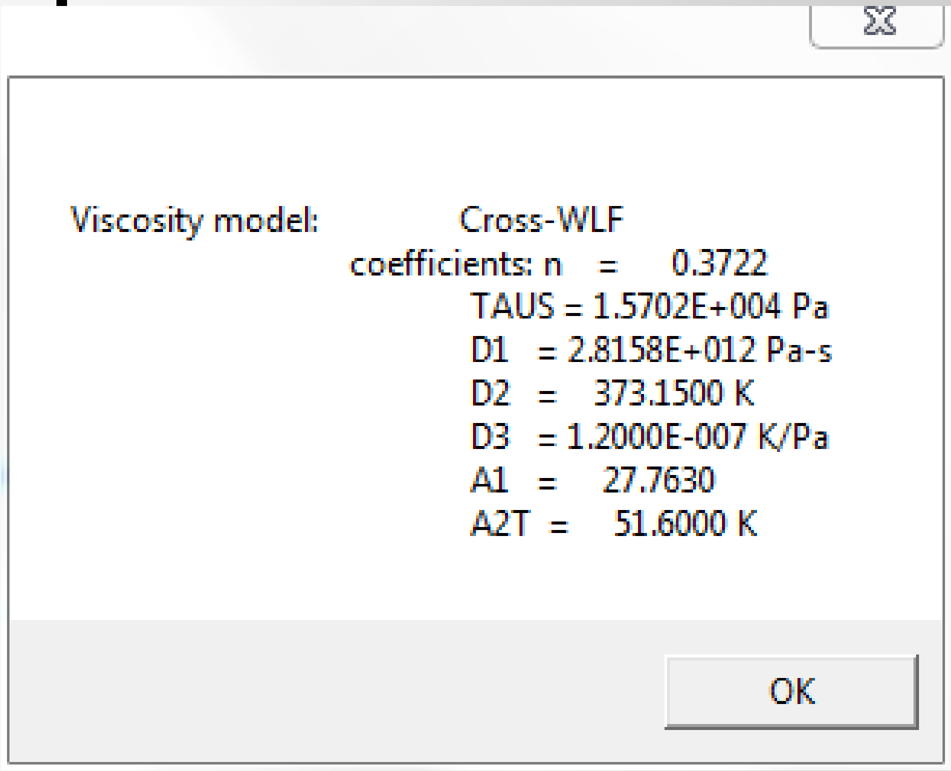
```
String   GetDataMessage (long aMsgID, Object aStrings, Object aValues, String aUnitSystem)
```

- No Help file containing Message Data (Yet)
  - Need to read …Moldflow Synergy 20xx/data/dat/cmmesage.dat

Understanding Key data files: cmmesage.dat

- Read viscosity coefficient from screen output

```
' Find the Viscosity Coefficient data
MessageID = 1313        ' from cmmesage.dat
Instance = -1           ' -1 latest
Set lStrings = Synergy.CreateStringArray()
Set lValues = Synergy.CreateDoubleArray()

OK = GetScreenMessageData(MessageID, Instance, lStrings, lValues)
If OK Then
        Dim SystemMessage
        Set SystemMessage = Synergy.SystemMessage
        Dim lStr
        lStr  = SystemMessage.GetDataMessage( MessageID, lStrings, lValues, Synergy.GetUnits())
        MsgBox lStr
End If
```



```
Viscosity model:          Cross-WLF
               coefficients: n   =     0.3722
                      TAUS = 1.5702E+004 Pa
                      D1   = 2.8158E+012 Pa-s
                      D2   =    373.1500 K
                      D3   = 1.2000E-007 K/Pa
                      A1   =     27.7630
                      A2T  =     51.6000 K
```

ReadSolverOutput.vbs   Edit   Execute

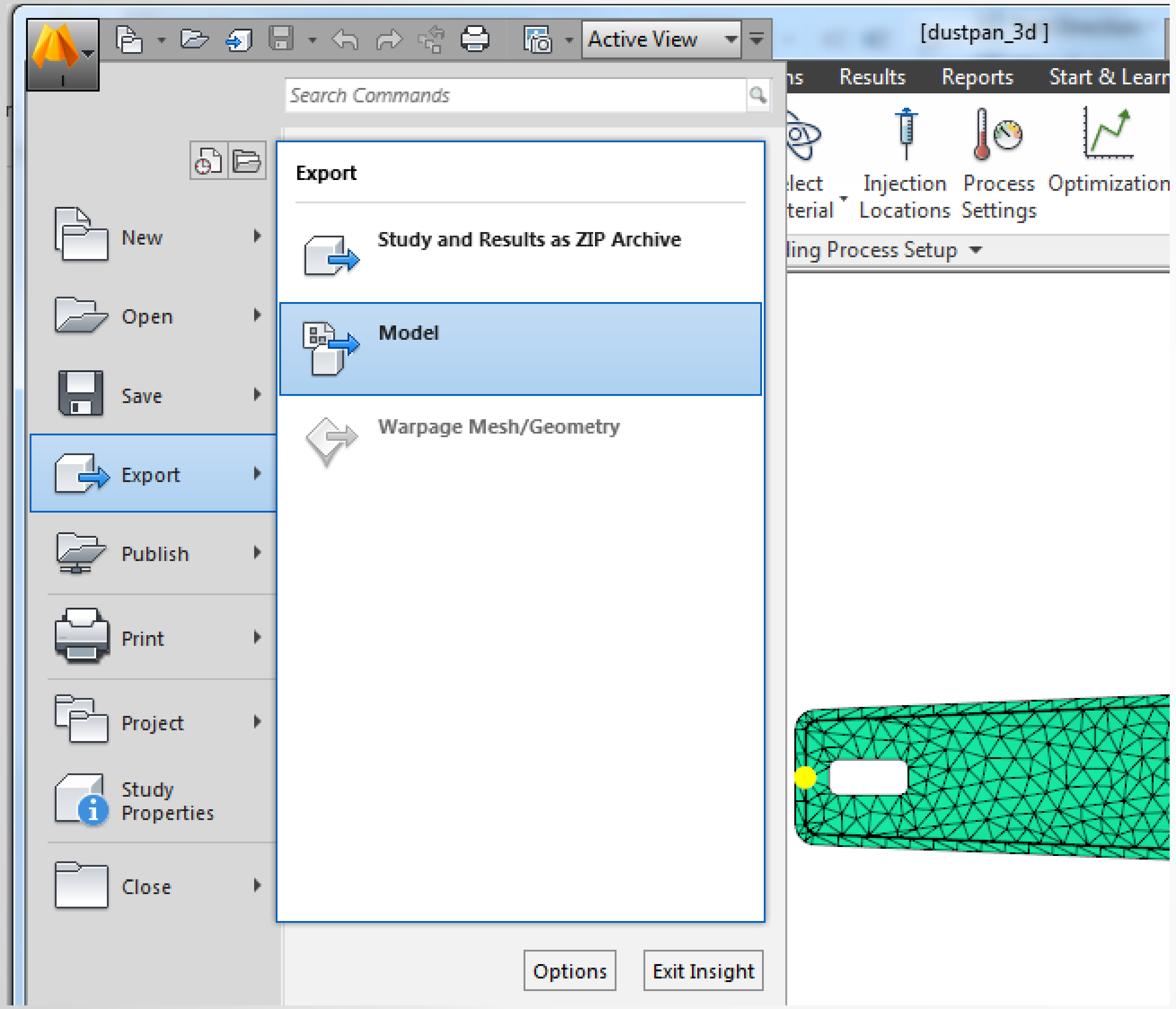
Exercise: Reading Screen Output

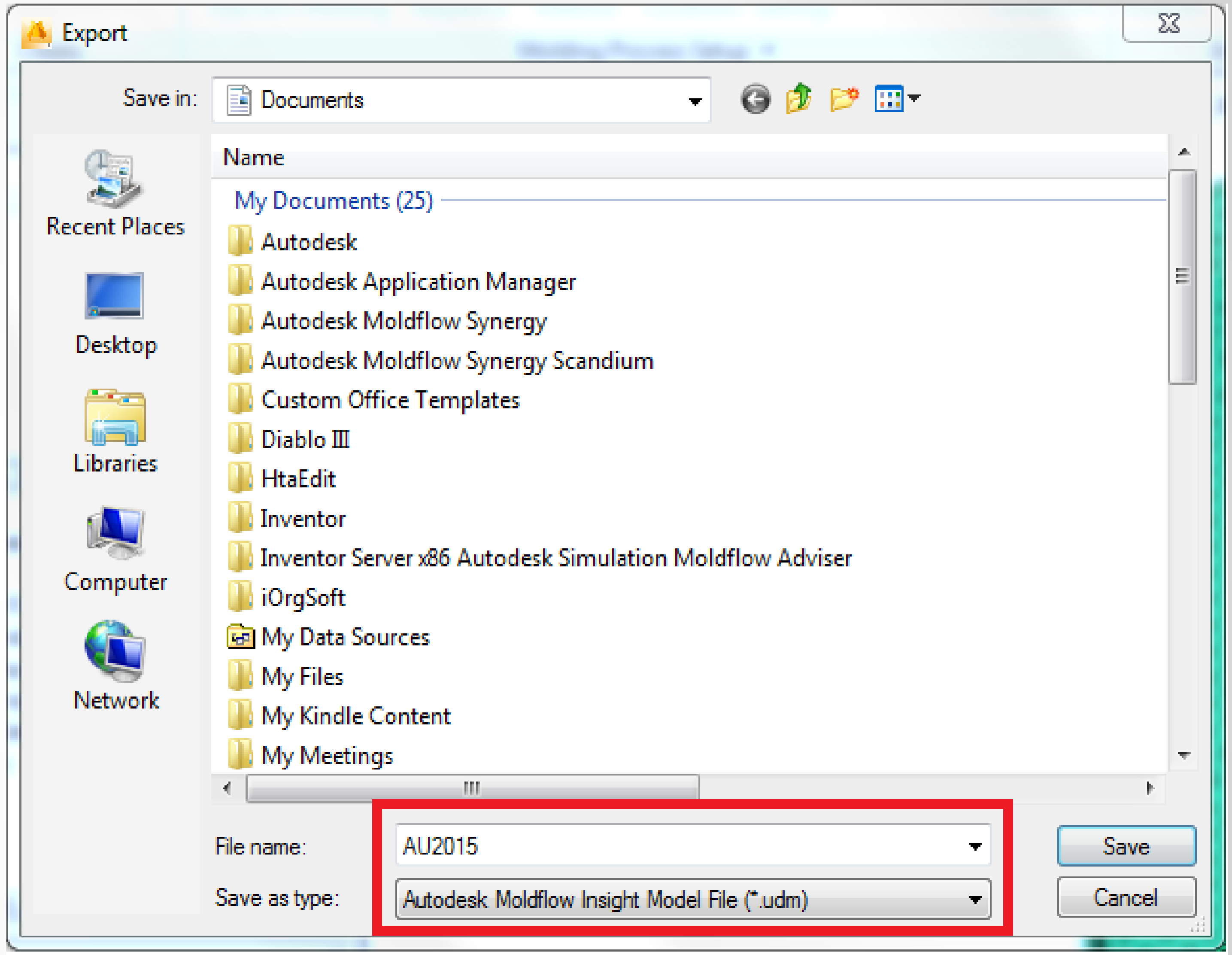
Module: Understanding the .SDY File

Understanding the Study File

- The default study file is a zipped binary file
  - To reduce the physical size of the file
  - Encrypted to protect content
- To view the content of a Study File
  - Export an Autodesk Moldflow Insight Model File (*.udm)
    - This file is ASCII
    - The file contains all data except material data.

Understanding the Study File





File name: AU2015

Save as type: Autodesk Moldflow Insight Model File (*.udm)

Example: Autodesk Moldflow Insight Model File

Understanding the Study File

- Sections in the Study File
  - Content covered today

TITL	Title Information	
GLBL	Global Data	
SUMR	Entity Summary Data	
NODE	Node Data	
1DET	1DET (Beam) Data	
TRI3	TRI3 (Shell) Data	
TET4	TET4 (Tetrahedral) Data	
TSET	TSET Data	- Used to store attribute/property data
NDBC	NDBC (Nodal Boundary Condition) Data	

Understanding the Study File

- Sections in the Study File
  - Not covered today

COLR	Colour Data
LYER	Layer
CURV	Curves Data
SURF	Surface Data
BLOB	Blob Data
REGN	Region Data
LOCS	Local Coordinate System Data
SUBC	Surface Boundary Condition
STLR	STL Facet

Understanding the Study File -TITL

- TITL Data
    - Study Header Data

```
TITL{
      NAME{"Untitled"}
      VRSN{"MOLDFLOW"}
      KEYW{"4869"}
      DATE{"NOV-14-2015"}
      TIME{"10:40:18"}
      UDMV{"UDM V7"} // Do not change the UDMV keyword.}
```

Understanding the Study File - GLBL

- GLBL Data
  - Global Data

```
//  Beginning of global data set : MOLDFLOW application keywords storage.
GLBL{  LUDM{    "Version" "asms2017:20151110.1516"}}
```

Understanding the Study File - SUMR

- SUMR Data
  - Summary of number of entities of each in the Study File

```
SUMR{
        NOCL{NUM_COLORS}                      NOLY{NUM_LAYERS}
        NOTS{NUM_TCODE_SETS}                  NOVW{NUM_VIEWS}
        NOND{NUM_NODES}                       NOTX{NUM_TEXTS}
        NO1D{NUM_1DET_ELEMENTS}               NOT3{NUM_TRI3_ELEMENTS}
        NOCV{NUM_CURVES}                      NOSF{NUM_SURFACES}
        NORG{NUM_REGIONS}                     NONB{NUM_NODE_BOUNDARY_CONDITIONS}
        NOSB{NUM_SURFACE_BOUNDARY_CONDITIONS}          NOCP{NUM_CUTTING_PLANES}
        NOLC{NUM_LOCAL_COORDINATE_SYSTEMS}             NOT4{NUM_TET4_ELEMENTS}
        NOSR{NUM_STL_REGIONS}                          NOBL{NUM_BLOBS}
        NOWD{NUM_WEDGE_ELEMENTS}                       NOIE{NUM_INTERFACE_ELEMENTS}}
```

Note: NUM_XXX is the count not the Maximum Entity Label

Reading Global Data from the Study

- Class: Synergy

Synergy.Edition()	Edition of Synergy
Synergy.Version()	Version ie. 2016, 2017
Synergy.Build()	Synergy.exe build number

Reading Global Data from the Study

- Class: StudyDoc

  StudyDoc.MoldingProcess()

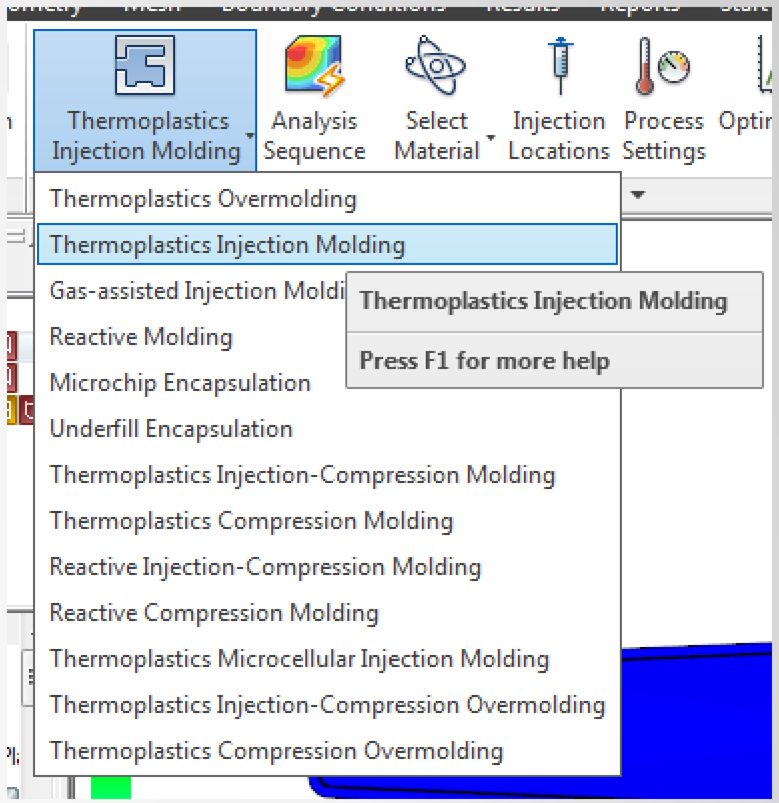
  StudyDoc.AnalysisSequence()

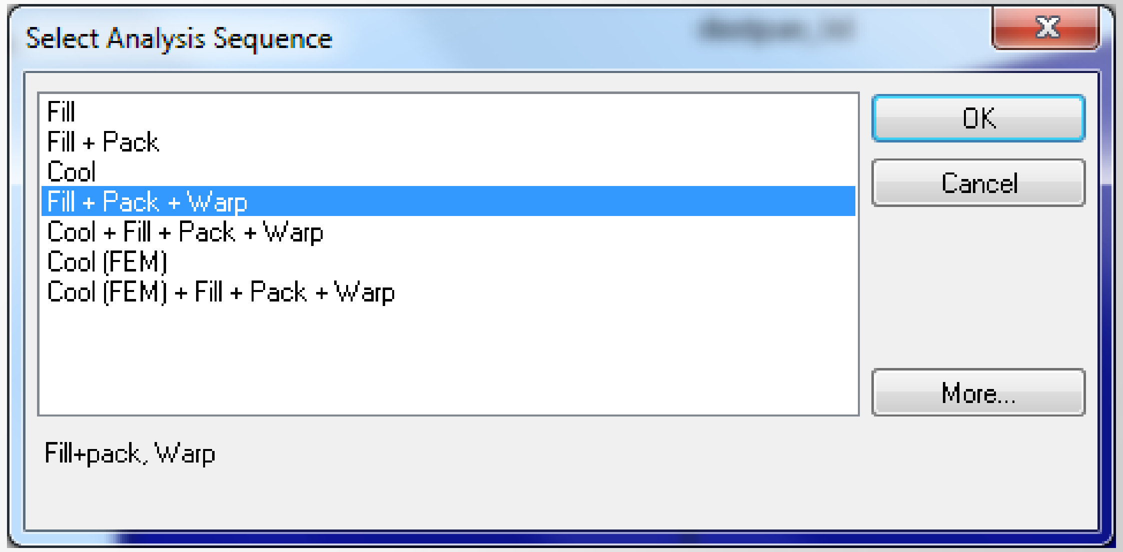
  StudyDoc.MeshType()

  StudyDoc.StudyName()

- Class: Project

  Project.Path()


Thermoplastics Injection Molding
Analysis Sequence
Select Material
Injection Locations
Process Settings
Optir
Thermoplastics Overmolding
Thermoplastics Injection Molding
Gas-assisted Injection Moldi
Reactive Molding
Microchip Encapsulation
Underfill Encapsulation
Thermoplastics Injection-Compression Molding
Thermoplastics Compression Molding
Reactive Injection-Compression Molding
Reactive Compression Molding
Thermoplastics Microcellular Injection Molding
Thermoplastics Injection-Compression Overmolding
Thermoplastics Compression Overmolding
Thermoplastics Injection Molding
Press F1 for more help



Understanding the Study File - NODE

- Nodal Data

NODE{LABEL LABEL_ON LAYER COLOR MESH_SIZE X Y Z}

LABEL	Node Number
LABEL_ON	Show node Label 0 = Off, 1 = On
LAYER	Layer Index
COLOR	Color Index
MESH_SIZE	Mesh Size (Currently not used)
X	X Coordinate
Y	Y Coordinate
Z	Z Coordinate

Understanding the Study File - NODE

- Nodal (NODE) Data

```
NODE{1 0 3 0 0.000000e+000 -1.673551e-004 -1.673551e-004 -1.917700e-002}
NODE{2 0 3 0 0.000000e+000 -1.673551e-004 2.332645e-003 -1.917700e-002}
NODE{4 0 3 0 0.000000e+000 -1.673551e-004 7.332645e-003 -1.917700e-002}
….
NODE{234655 0 10 0 0.000000e+000 -1.673551e-004 1.733264e-002 -1.917700e-002}
NODE{234656 0 10 0 0.000000e+000 -1.673551e-004 1.983264e-002 -1.917700e-002}
```

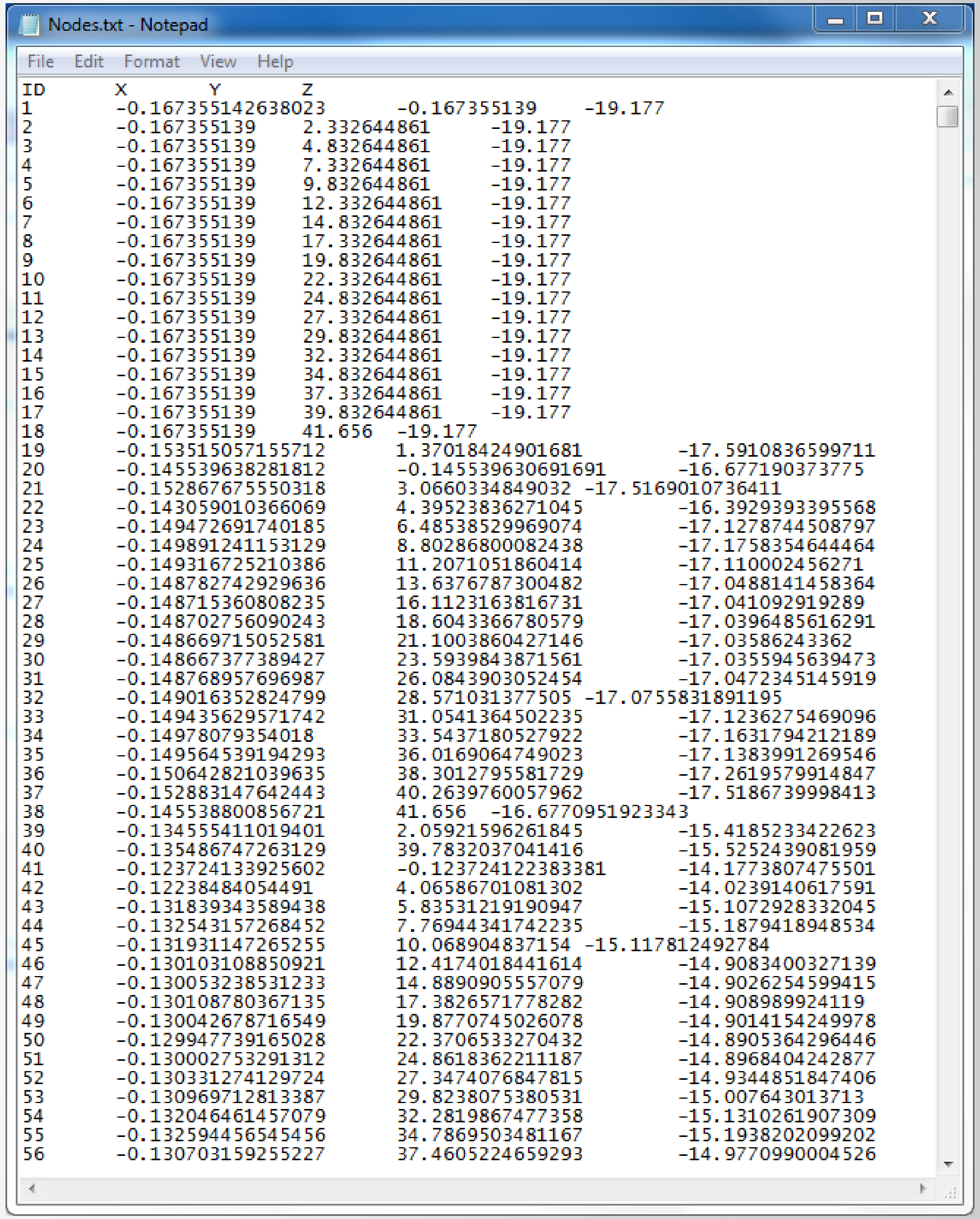
- Note: Gaps in node numbering

Example: Autodesk Moldflow Insight Model File

Understanding the Study File - NODE

- Reading Node (NODE) Data

```
DataFileName =  GetTempPath() & "\NODE.txt"
Set DataFile = FS.CreateTextFile(DataFileName, True)
DataFile.Write "ID" & Chr(9) & "X" & Chr(9) & _
                        "Y" & Chr(9) & "Z" & vbCRLF
Set StudyDoc = Synergy.StudyDoc()
Set Node = StudyDoc.GetFirstNode()
While Not Node Is nothing
    Label = StudyDoc.GetEntityID(Node)
    Set Coord = StudyDoc.GetNodeCoord(Node)
    DataFile.Write Label & Chr(9) & Coord.X & Chr(9) & _
                   Coord.Y & Chr(9) & Coord.Z & vbCrLf
    Set Node = StudyDoc.GetNextNode(Node)
Wend
DataFile.Close
```



```
ID      X       Y       Z
1       -0.167355142638023      -0.167355139    -19.177
2       -0.167355139    2.332644861     -19.177
3       -0.167355139    4.832644861     -19.177
4       -0.167355139    7.332644861     -19.177
5       -0.167355139    9.832644861     -19.177
6       -0.167355139    12.332644861    -19.177
7       -0.167355139    14.832644861    -19.177
8       -0.167355139    17.332644861    -19.177
9       -0.167355139    19.832644861    -19.177
10      -0.167355139    22.332644861    -19.177
11      -0.167355139    24.832644861    -19.177
12      -0.167355139    27.332644861    -19.177
13      -0.167355139    29.832644861    -19.177
14      -0.167355139    32.332644861    -19.177
15      -0.167355139    34.832644861    -19.177
16      -0.167355139    37.332644861    -19.177
17      -0.167355139    39.832644861    -19.177
18      -0.167355139    41.656  -19.177
19      -0.153515057155712      1.37018424901681        -17.5910836599711
20      -0.145539638281812      -0.145539630691691      -16.677190373775
21      -0.152867675550318      3.0660334849032 -17.5169010736411
22      -0.143059010366069      4.39523836271045        -16.3929393395568
23      -0.149472691740185      6.48538529969074        -17.1278744508797
24      -0.149891241153129      8.80286800082438        -17.1758354644464
25      -0.149316725210386      11.2071051860414        -17.110002456271
26      -0.148782742929636      13.6376787300482        -17.0488141458364
27      -0.148715360808235      16.1123163816731        -17.041092919289
28      -0.148702756090243      18.6043366780579        -17.0396485616291
29      -0.148669715052581      21.1003860427146        -17.0358624336 2
30      -0.148667377389427      23.5939843871561        -17.0355945639473
31      -0.148768957696987      26.0843903052454        -17.0472345145919
32      -0.149016352824799      28.571031377505 -17.0755831891195
33      -0.149435629571742      31.0541364502235        -17.1236275469096
34      -0.14978079354018       33.5437180527922        -17.163179421 2189
35      -0.149564539194293      36.0169064749023        -17.1383991269546
36      -0.150642821039635      38.3012795581729        -17.2619579914847
37      -0.152883147642443      40.2639760057962        -17.5186739998413
38      -0.145538800856721      41.656  -16.6770951923343
39      -0.134555411019401      2.05921596261845        -15.418523342 2623
40      -0.135486747263129      39.7832037041416        -15.5252439081959
41      -0.123724133925602      -0.123724122383381      -14.1773807475501
42      -0.12238484054491       4.06586701081302        -14.0239140617591
43      -0.131839343589438      5.83531219190947        -15.1072928332045
44      -0.132543157268452      7.76944341742235        -15.1879418948534
45      -0.131931147265255      10.068904837154 -15.117812492784
46      -0.130103108850921      12.4174018441614        -14.908340032 7139
47      -0.130053238531233      14.8890905557079        -14.9026254599415
48      -0.130108780367135      17.3826571778282        -14.908989924119
49      -0.130042678716549      19.8770745026078        -14.9014154249978
50      -0.129947739165028      22.3706533270432        -14.8905364296446
51      -0.130002753291312      24.8618362211187        -14.8968404242877
52      -0.130331274129724      27.3474076847815        -14.9344851847406
53      -0.130969712813387      29.8238075380531        -15.007643013713
54      -0.132046461457079      32.2819867477358        -15.1310261907309
55      -0.132594456545456      34.7869503481167        -15.1938202099202
56      -0.130703159255227      37.4605224659293        -14.9770990004526
```

WriteNODEtoFile.vbs Edit Execute

Understanding the Study File – 1DET

- Beam (1DET) Data

1DET{LABEL LABEL_ON LAYER COLOR DISPLAY GROUP TSET SUBID N1 N2}

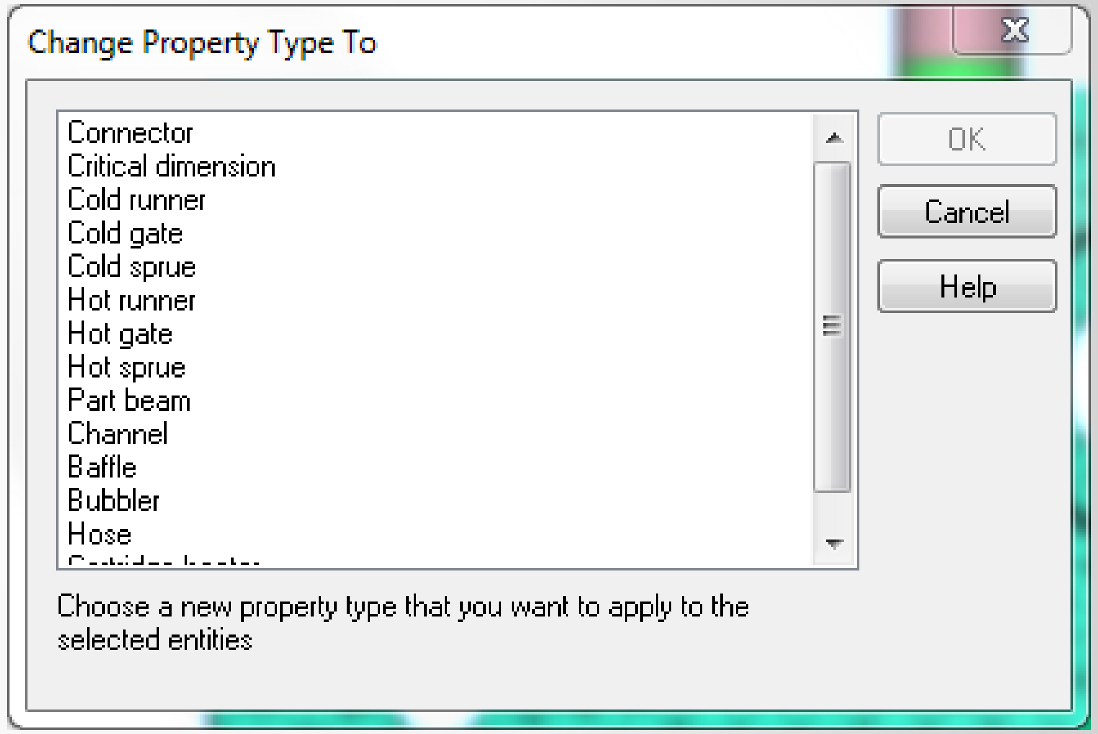
LABEL	Element Number
LABEL_ON	Show node Label 0 = Off, 1 = On
LAYER	Layer Index
COLOR	Color Index
DISPLAY	
GROUP	
TSET	Tset type
SUBID	Tset Sub ID
N1	Node 1
N2	Node 2

Example: Autodesk Moldflow Insight Model File

Understanding the Study File – 1DET

- Beam (1DET) Data

```
1DET{10857 0 5 0 0 "C1" 40428 35 5404 5405}
1DET{10858 0 5 0 0 "C1" 40428 36 5405 5406}
1DET{10860 0 5 0 0 "C1" 40428 38 5407 5408}
.....
1DET{10860 0 5 0 0 "C27" 40428 23 5407 5408
1DET{10861 0 5 0 0 "C67" 40420 23 5300 5301
1DET{10862 0 5 0 0 "C67" 40420 23 5301 5302}
```



Choose a new property type that you want to apply to the selected entities

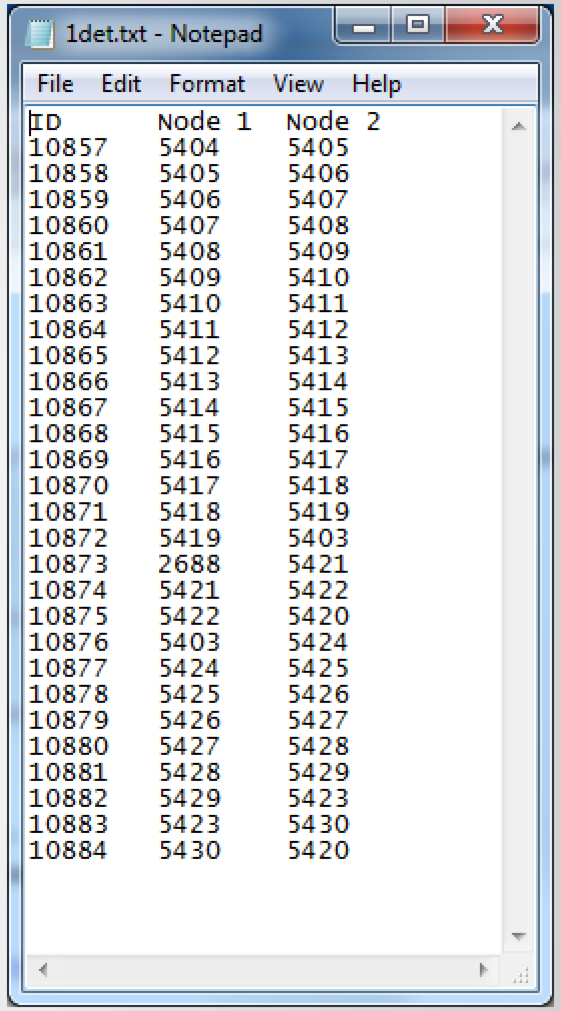
- Note: Gaps in numbering
- TSET defines the type of Beam
- Many elements many point to the same TSet and TSet Sub ID

Example: Autodesk Moldflow Insight Model File

Understanding the Study File – 1DET

- Reading Beam (1DET) Data

```
DataFileName =  GetTempPath() & "\1DET.txt"
Set DataFile = FS.CreateTextFile(DataFileName, True)
DataFile.Write "ID" & Chr(9) & "Node 1" & Chr(9) & "Node 2" & vbCRLF
Set StudyDoc = Synergy.StudyDoc()
Set Beam = StudyDoc.GetFirstBeam()
While Not Beam Is nothing
    Label = StudyDoc.GetEntityID(Beam)
    Set Nodes = StudyDoc.GetElemNodes(Beam)
    N1 = StudyDoc.GetEntityID(Nodes.Entity(0))
    N2 = StudyDoc.GetEntityID(Nodes.Entity(1))
    DataFile.Write Label & Chr(9) & N1 & Chr(9) & N2 & vbCrLf
    Set Beam = StudyDoc.GetNextBeam(Beam)
Wend
DataFile.Close
```



```
ID      Node 1  Node 2
10857   5404    5405
10858   5405    5406
10859   5406    5407
10860   5407    5408
10861   5408    5409
10862   5409    5410
10863   5410    5411
10864   5411    5412
10865   5412    5413
10866   5413    5414
10867   5414    5415
10868   5415    5416
10869   5416    5417
10870   5417    5418
10871   5418    5419
10872   5419    5403
10873   2688    5421
10874   5421    5422
10875   5422    5420
10876   5403    5424
10877   5424    5425
10878   5425    5426
10879   5426    5427
10880   5427    5428
10881   5428    5429
10882   5429    5423
10883   5423    5430
10884   5430    5420
```

Write1DETtoFile.vbs Edit Execute

Understanding the Study File – TRI3

- Triangle (TRI3) Data

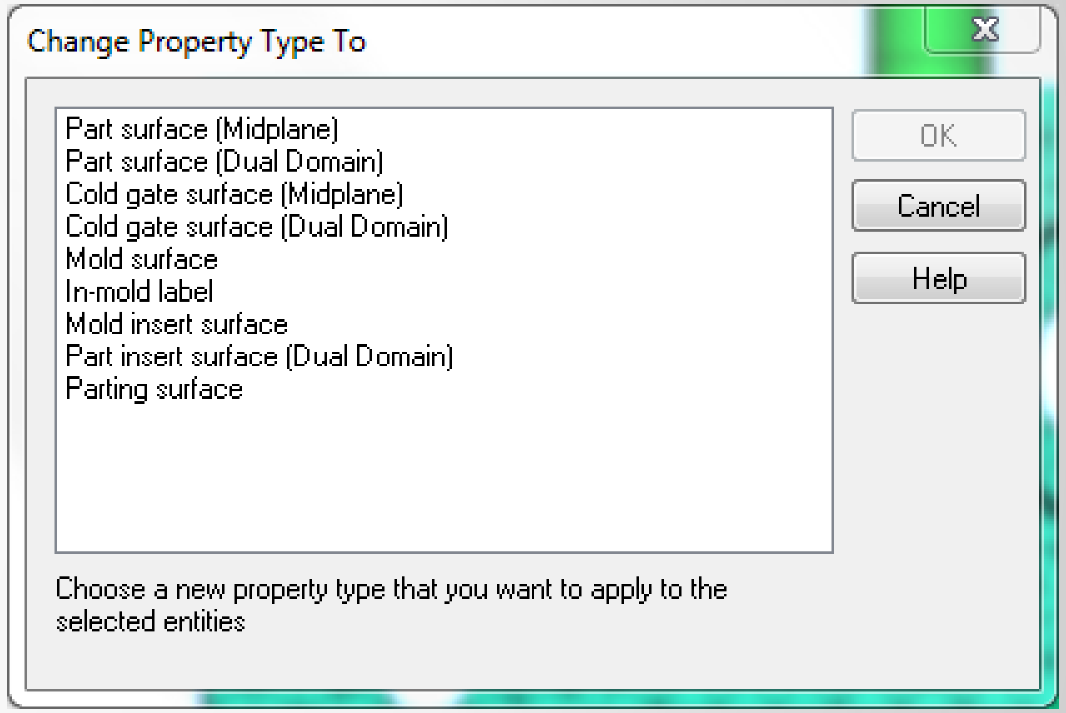
TRI3{LABEL LABEL_ON LAYER COLOR DISPLAY GROUP TSET SUBID N1 N2 N3}

LABEL	Element Number
LABEL_ON	Show node Label 0 = Off, 1 = On
LAYER	Layer Index
COLOR	Color Index
DISPLAY	
GROUP	
TSET	TSet type
SUBID	TSet Sub ID
N1	Node 1
N2	Node 2
N3	Node 3

Understanding the Study File – TRI3

- Triangle (TRI3) Data

```
TRI3{1 0 5 0 0 "S17" 40801 1525 2674 2602 2676}
TRI3{3 0 5 0 0 "S17" 40801 1526 2602 2539 2570}
TRI3{4 0 5 0 0 "S17" 40801 1526 2540 2603 2570}
…..
TRI3{9600 0 5 0 0 "S1" 40801 1527 2234 456 1}
```



Part surface (Midplane)
Part surface (Dual Domain)
Cold gate surface (Midplane)
Cold gate surface (Dual Domain)
Mold surface
In-mold label
Mold insert surface
Part insert surface (Dual Domain)
Parting surface

Choose a new property type that you want to apply to the selected entities

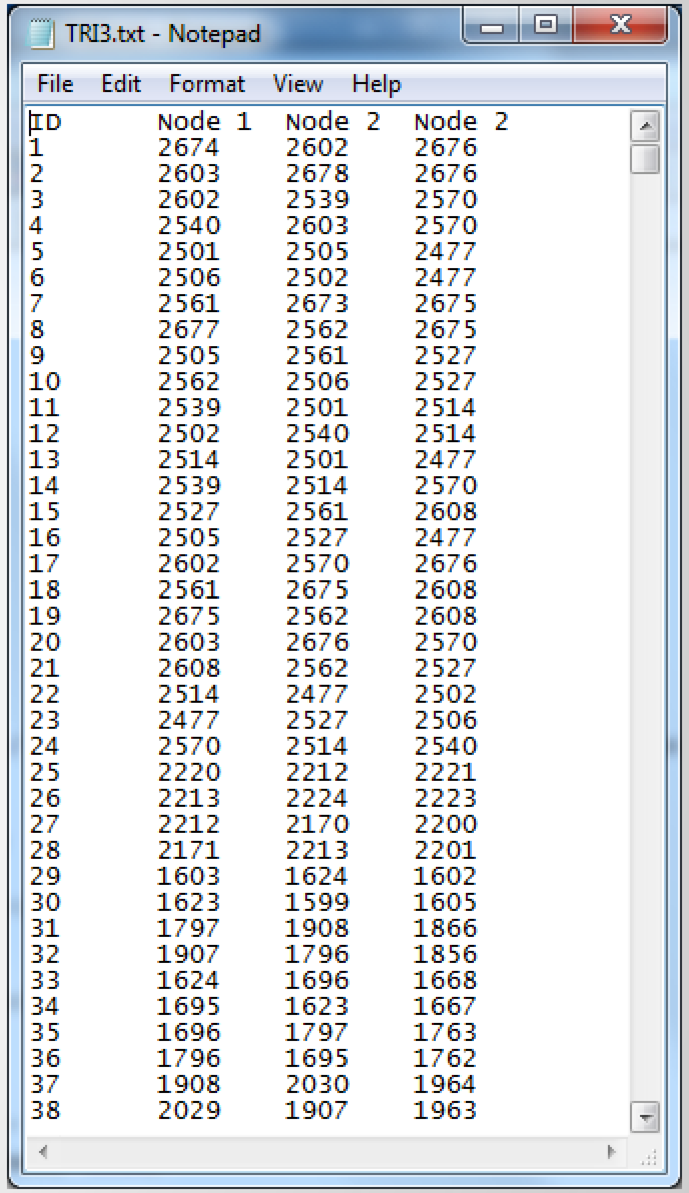
- Note: Gaps in numbering
- TSET defines the Triangle (TRI3) type
- Many elements many point to the same TSet and TSet Sub ID

Example: Autodesk Moldflow Insight Model File

Understanding the Study File – TRI3

- Reading Triangle (TRI3) Data

```
DataFileName =  GetTempPath() & "\TRI3.txt"
Set DataFile = FS.CreateTextFile(DataFileName, True)
DataFile.Write "ID" & Chr(9) & "Node 1" & Chr(9) & "Node 2" & Chr(9) & "Node 2" & vbCRLF
Set StudyDoc = Synergy.StudyDoc()
Set Tri = StudyDoc.GetFirstTri()
While Not Tri Is nothing
    Label = StudyDoc.GetEntityID(Tri)
    Set Nodes = StudyDoc.GetElemNodes(Tri)
    N1 = StudyDoc.GetEntityID(Nodes.Entity(0))
    N2 = StudyDoc.GetEntityID(Nodes.Entity(1))
    N3 = StudyDoc.GetEntityID(Nodes.Entity(2))
    DataFile.Write Label & Chr(9) & N1 & Chr(9) & N2 & Chr(9) & N3 & vbCrLf
    Set Tri = StudyDoc.GetNextTri(Tri)
Wend
DataFile.Close
```



```
ID          Node 1      Node 2      Node 2
1           2674        2602        2676
2           2603        2678        2676
3           2602        2539        2570
4           2540        2603        2570
5           2501        2505        2477
6           2506        2502        2477
7           2561        2673        2675
8           2677        2562        2675
9           2505        2561        2527
10          2562        2506        2527
11          2539        2501        2514
12          2502        2540        2514
13          2514        2501        2477
14          2539        2514        2570
15          2527        2561        2608
16          2505        2527        2477
17          2602        2570        2676
18          2561        2675        2608
19          2675        2562        2608
20          2603        2676        2570
21          2608        2562        2527
22          2514        2477        2502
23          2477        2527        2506
24          2570        2514        2540
25          2220        2212        2221
26          2213        2224        2223
27          2212        2170        2200
28          2171        2213        2201
29          1603        1624        1602
30          1623        1599        1605
31          1797        1908        1866
32          1907        1796        1856
33          1624        1696        1668
34          1695        1623        1667
35          1696        1797        1763
36          1796        1695        1762
37          1908        2030        1964
38          2029        1907        1963
```

WriteTRI3toFile.vbs   Edit   Execute

Understanding the Study File – TET4

- Tetrahedral (TET4) Data

TRI3{LABEL LABEL_ON LAYER COLOR DISPLAY GROUP TSET SUBID N1 N2 N3 N4}

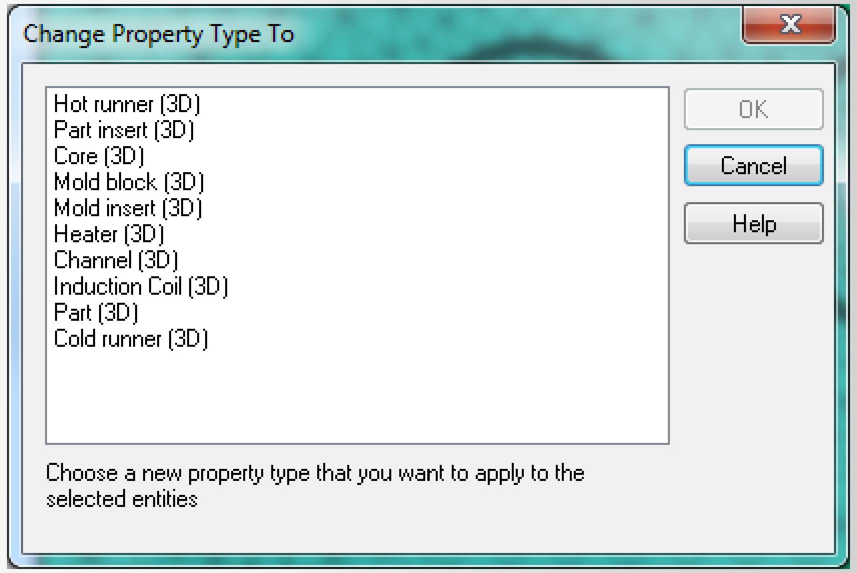
LABEL	Element Number
LABEL_ON	Show node Label 0 = Off, 1 = On
LAYER	Layer Index
COLOR	Color Index
DISPLAY	
GROUP	
TSET	Tset type
SUBID	Tset Sub ID
N1	Node 1
N2	Node 2
N3	Node 3
N4	Node 4

Example: Autodesk Moldflow Insight Model File

Understanding the Study File – TET4

- Tetrahedral (TET4) Data

```
TET4{1676 0 9 0 0 "TE1" 50400 1 4275 6871 4152 6867}
TET4{2455 0 9 0 0 "TE1" 50400 1 3211 6624 6625 6617}
TET4{3421 0 9 0 0 "TE1" 50400 1 2291 2447 6280 6329}
....
TET4{3472 0 9 0 0 "TE1" 50400 1 6334 2310 6332 6314}
TET4{3545 0 9 0 0 "TE1" 50400 1 6286 6300 2178 6285}
```



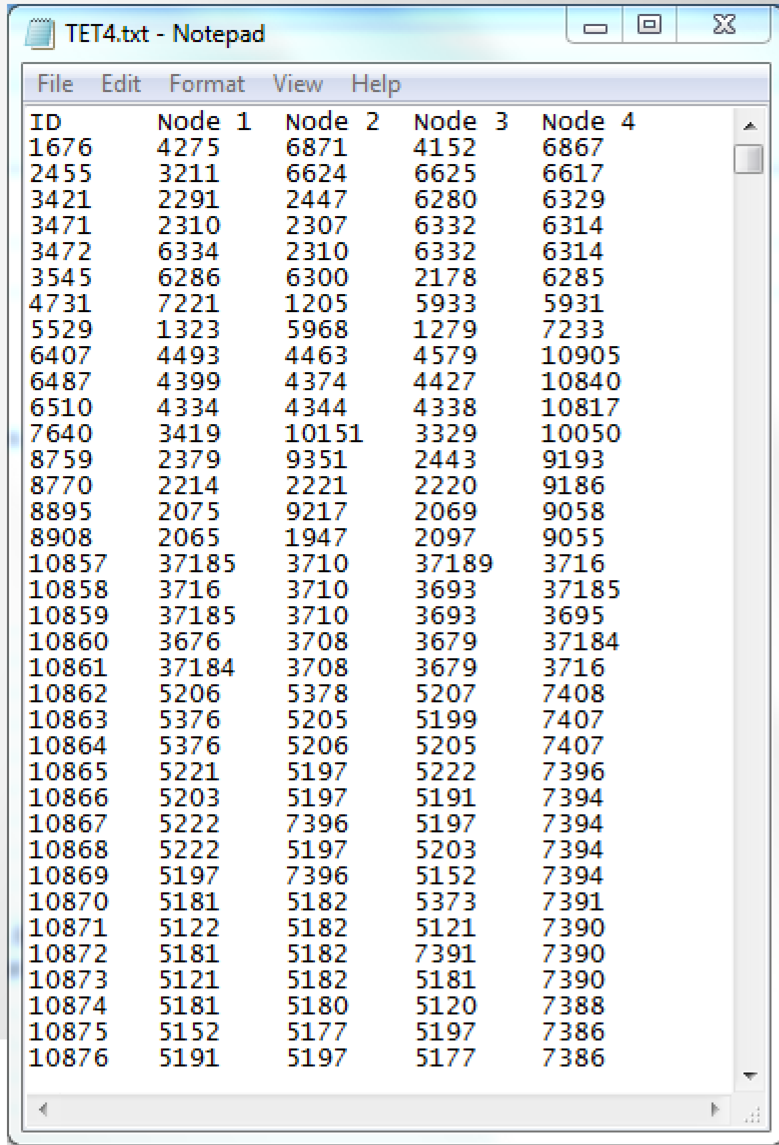
Choose a new property type that you want to apply to the selected entities

  - Note: Gaps in numbering
  - TSET defines the type of Tetrahedra
  - Many elements many point to the same TSet and TSet Sub ID

Understanding the Study File – TET4

- Reading Tetrahedral (TET4) Data

```
DataFileName =  GetTempPath() & "\TET4.txt"
Set DataFile = FS.CreateTextFile(DataFileName, True)
DataFile.Write "ID" & Chr(9) & "Node 1" & Chr(9) & "Node 2" & Chr(9) & "Node 2" & Chr(9) & "Node 4" & vbCRLF
Set StudyDoc = Synergy.StudyDoc()
Set Tet = StudyDoc.GetFirstTet()
While Not Tet Is nothing
    Label = StudyDoc.GetEntityID(Tet)
    Set Nodes = StudyDoc.GetElemNodes(Tet)
    N1 = StudyDoc.GetEntityID(Nodes.Entity(0))
    N2 = StudyDoc.GetEntityID(Nodes.Entity(1))
    N3 = StudyDoc.GetEntityID(Nodes.Entity(2))
    N4 = StudyDoc.GetEntityID(Nodes.Entity(3))
    DataFile.Write Label & Chr(9) & N1 & Chr(9) & N2 & Chr(9) & N3 &  Chr(9) & N4 & vbCrLf
    Set Tet = StudyDoc.GetNextTet(Tet)
Wend
DataFile.Close
```



```
ID      Node 1  Node 2  Node 3  Node 4
1676    4275    6871    4152    6867
2455    3211    6624    6625    6617
3421    2291    2447    6280    6329
3471    2310    2307    6332    6314
3472    6334    2310    6332    6314
3545    6286    6300    2178    6285
4731    7221    1205    5933    5931
5529    1323    5968    1279    7233
6407    4493    4463    4579    10905
6487    4399    4374    4427    10840
6510    4334    4344    4338    10817
7640    3419    10151   3329    10050
8759    2379    9351    2443    9193
8770    2214    2221    2220    9186
8895    2075    9217    2069    9058
8908    2065    1947    2097    9055
10857   37185   3710    37189   3716
10858   3716    3710    3693    37185
10859   37185   3710    3693    3695
10860   3676    3708    3679    37184
10861   37184   3708    3679    3716
10862   5206    5378    5207    7408
10863   5376    5205    5199    7407
10864   5376    5206    5205    7407
10865   5221    5197    5222    7396
10866   5203    5197    5191    7394
10867   5222    7396    5197    7394
10868   5222    5197    5203    7394
10869   5197    7396    5152    7394
10870   5181    5182    5373    7391
10871   5122    5182    5121    7390
10872   5181    5182    7391    7390
10873   5121    5182    5181    7390
10874   5181    5180    5120    7388
10875   5152    5177    5197    7386
10876   5191    5197    5177    7386
```

WriteTET4toFile.vbs   Edit   Execute

Understanding the Study File

- Determining which Layer an Entity is in

```
Set Ent = StudyDoc.GetFirstNode()
Set Ent = StudyDoc.GetFirstBeam()
Set Ent = StudyDoc.GetFirstTet()
Set Ent = StudyDoc.GetFirstTri()


Set Layer = StudyDoc.GetEntityLayer(Ent)
MsgBox StudyDoc.GetEntityID( Layer )
```

Understanding the Study File

- Reading the TSet and Tset Sub ID of an Entity

```
    Set Ent = StudyDoc.GetFirstNode()
    Set Ent = StudyDoc.GetFirstBeam()
    Set Ent = StudyDoc.GetFirstTet()
    Set Ent = StudyDoc.GetFirstTri()

Set Prop = PropertyEditor.GetEntityProperty(Ent)
TSet = Prop.Type
TSetSubID = Prop.ID
```

Understanding the Study File

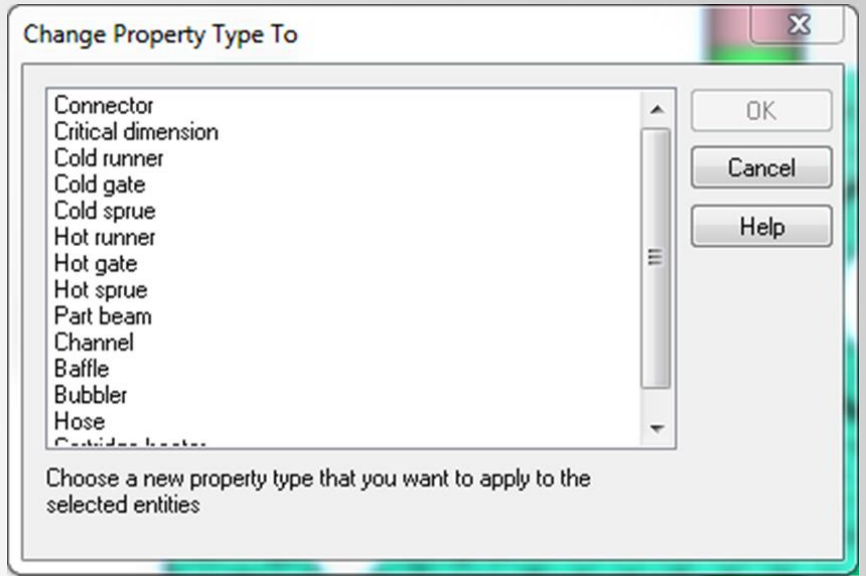
- Reading Geometry a better way
  - Why do we want a better way?
    - Using Standard API function can be very slow
    - We generally only want to read part of the data
      - Plastic Part Only
      - Cooling Circuits only
      - Note: There are no attributes on nodes so generally not filtered
    - Want to make it (relatively) simple

Understanding the Study File

- ReadMesh.vbs
  - Read NODE data
  - Read Filtered 1DET data
  - Read Filtered TRI3 data
  - Read Filtered TET4 data

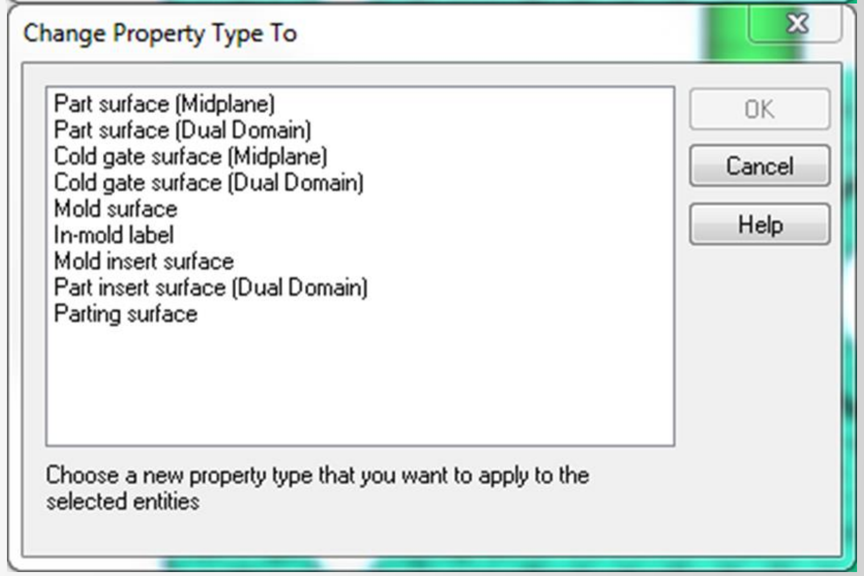
Edit ReadMesh.vbs

Execute ReadMesh.vbs



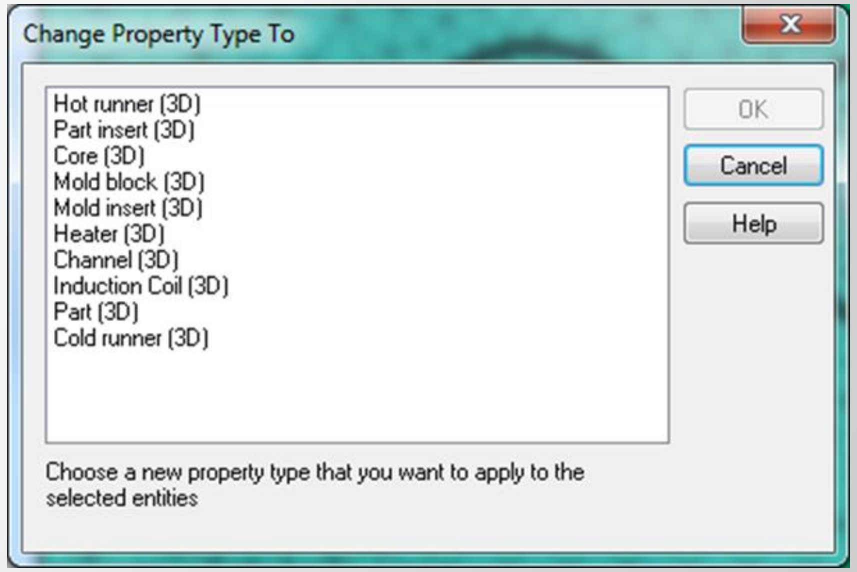
Connector
Critical dimension
Cold runner
Cold gate
Cold sprue
Hot runner
Hot gate
Hot sprue
Part beam
Channel
Baffle
Bubbler
Hose

Choose a new property type that you want to apply to the selected entities



Part surface (Midplane)
Part surface (Dual Domain)
Cold gate surface (Midplane)
Cold gate surface (Dual Domain)
Mold surface
In-mold label
Mold insert surface
Part insert surface (Dual Domain)
Parting surface

Choose a new property type that you want to apply to the selected entities

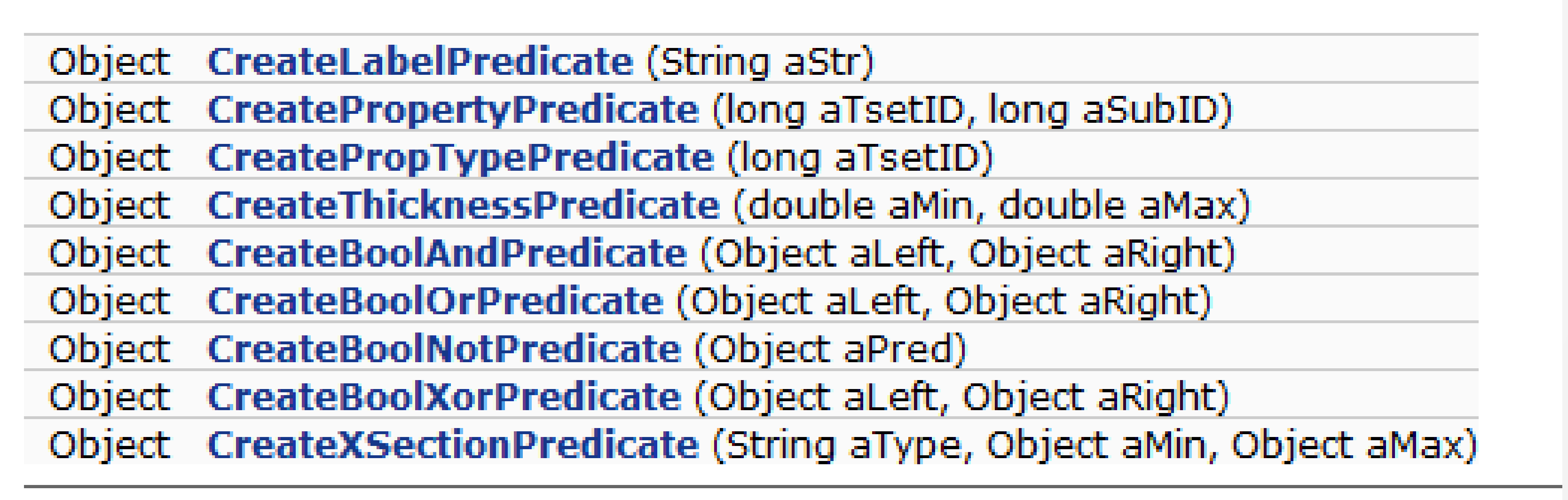


Hot runner (3D)
Part insert (3D)
Core (3D)
Mold block (3D)
Mold insert (3D)
Heater (3D)
Channel (3D)
Induction Coil (3D)
Part (3D)
Cold runner (3D)

Choose a new property type that you want to apply to the selected entities

Using the PredicateManager Class

- Selection by
  - Specified Properties/Attributes
  - Logical Operators to combine sets.



Object	**CreateLabelPredicate** (String aStr)
Object	**CreatePropertyPredicate** (long aTsetID, long aSubID)
Object	**CreatePropTypePredicate** (long aTsetID)
Object	**CreateThicknessPredicate** (double aMin, double aMax)
Object	**CreateBoolAndPredicate** (Object aLeft, Object aRight)
Object	**CreateBoolOrPredicate** (Object aLeft, Object aRight)
Object	**CreateBoolNotPredicate** (Object aPred)
Object	**CreateBoolXorPredicate** (Object aLeft, Object aRight)
Object	**CreateXSectionPredicate** (String aType, Object aMin, Object aMax)

Using the PredicateManager Class

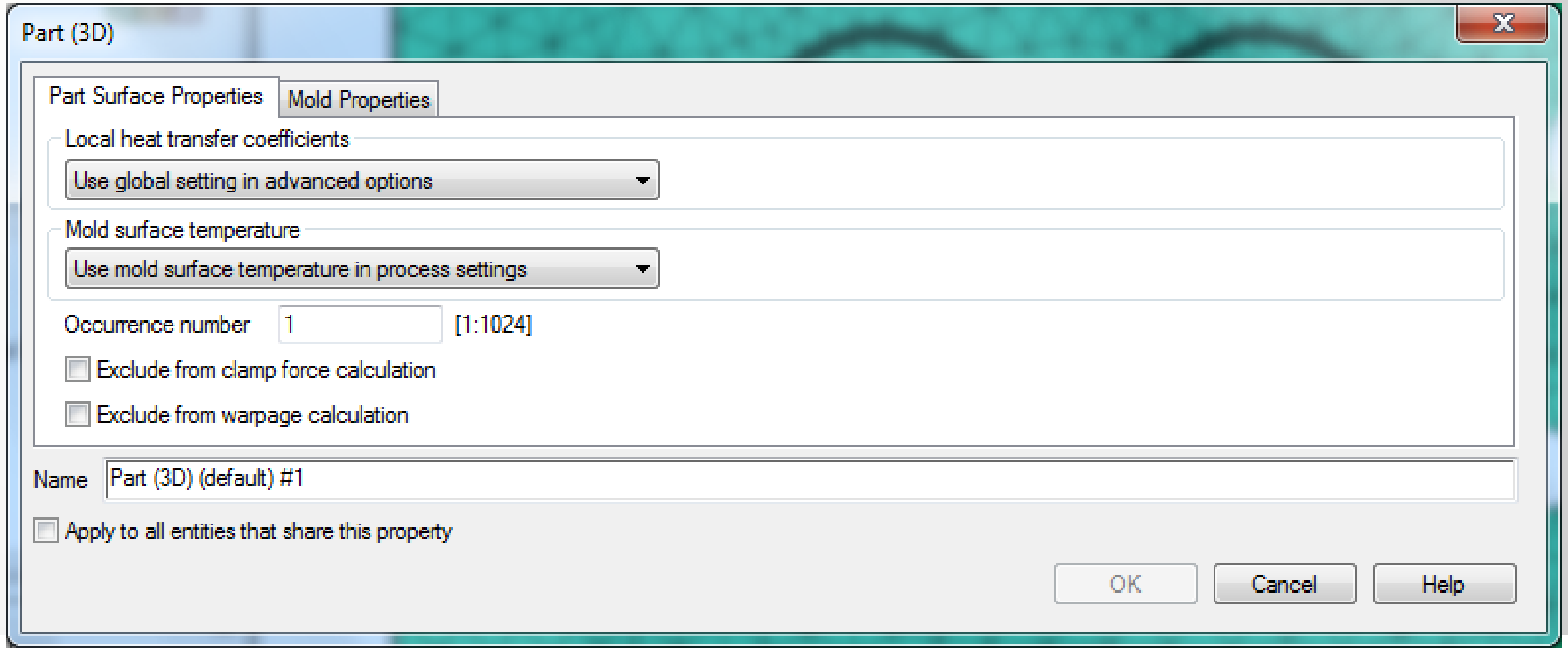
- Delete all Cooling Related Entities

```
' Select All elements Cooling Channels (40480)
Set Pred = PredicateManager.CreatePropTypePredicate(40480)
' Create an Entity List from the predicate
Set EntList = StudyDoc.CreateEntityList(40480)
EntList.SelectFromPredicate Pred
For I = 0 To EntList.Size()-1
        Set lEnt = EntList.Entity(I)
        Set Nodes = StudyDoc.GetElemNodes(lEnt)
        MeshEditor.Delete(lEnt)
        MeshEditor.Delete(Nodes)
Next
```

DeleteAllCoolingElements.vbs   Edit   Execute

Understanding the Study File

- Now we know how to read the Mesh/Geometry
  - How do we read Property/Attribute data?
- Property/Attribute Data
  - Example: Part 3D



Part (3D)

Local heat transfer coefficients: Use global setting in advanced options

Mold surface temperature: Use mold surface temperature in process settings

Occurrence number: 1 [1:1024]

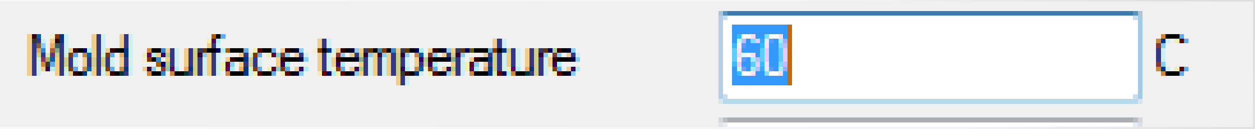
□ Exclude from clamp force calculation

□ Exclude from warpage calculation

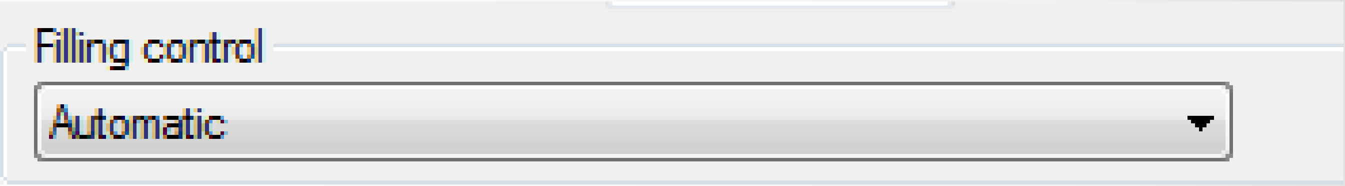
Name: Part (3D) (default) #1

□ Apply to all entities that share this property

TCodes

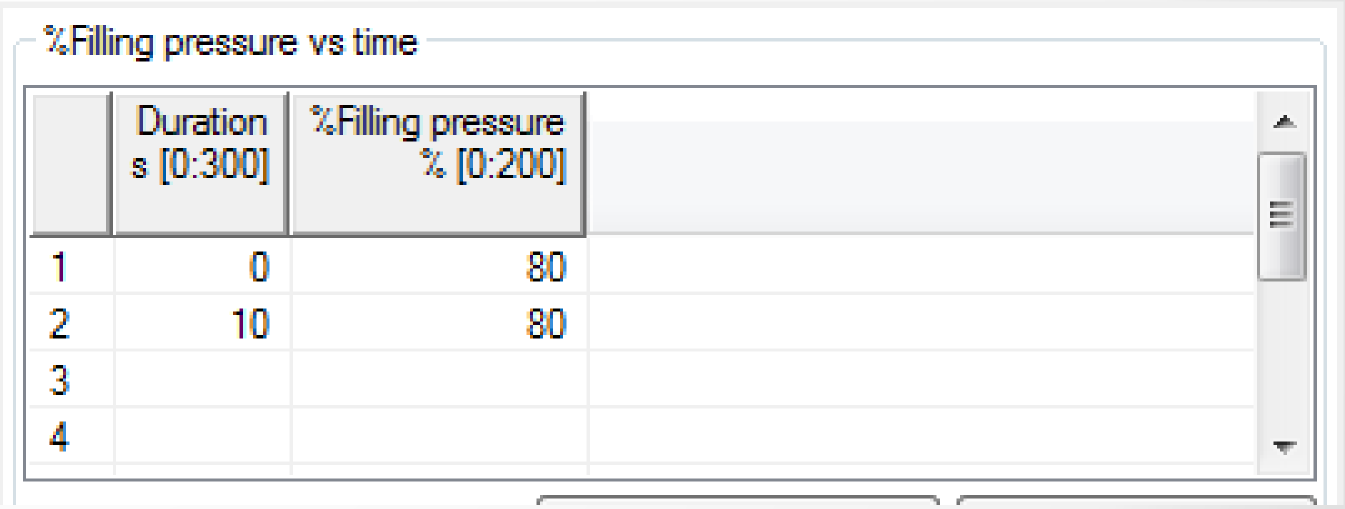


Mold surface temperature 60 C



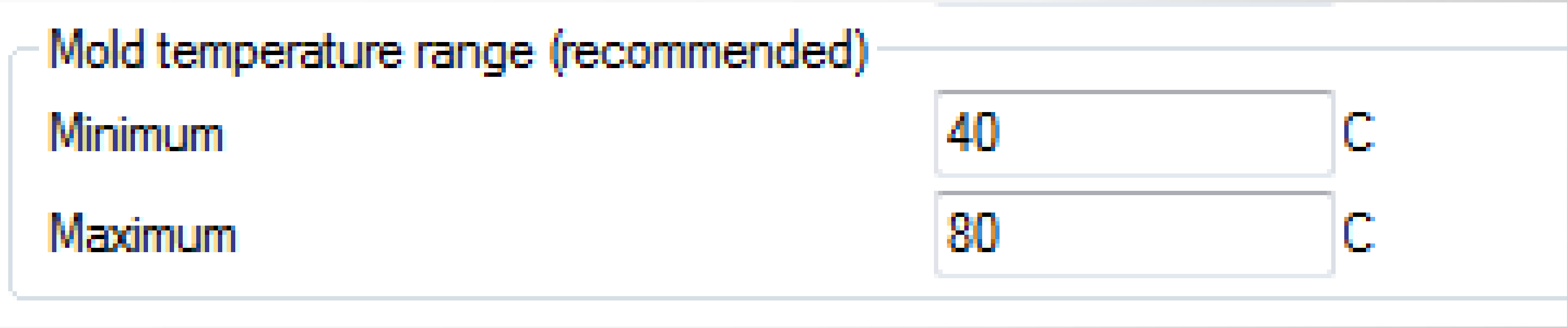
Filling control

Automatic



%Filling pressure vs time

	Duration s [0:300]	%Filling pressure % [0:200]
1	0	80
2	10	80
3		
4		

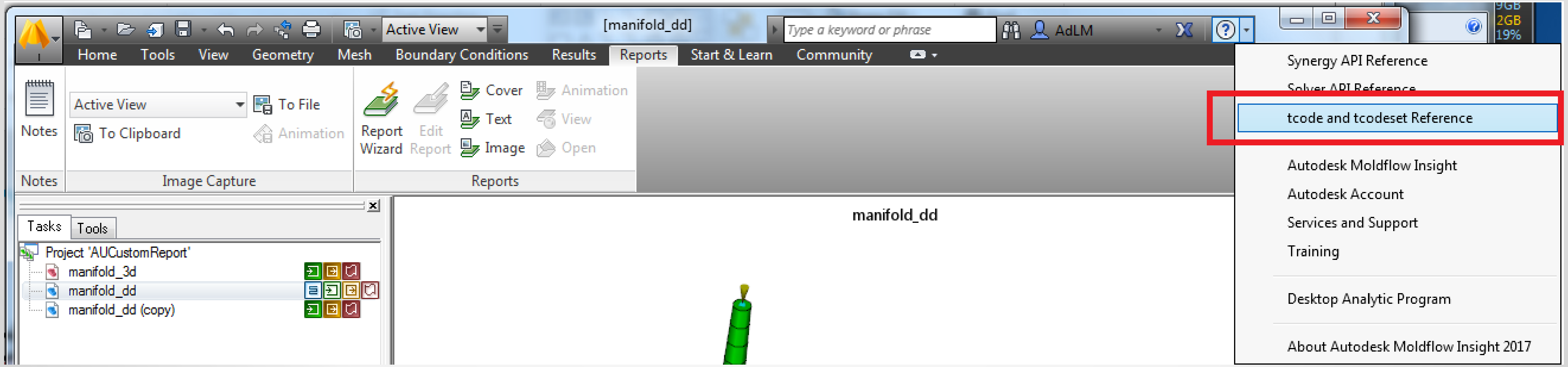


Mold temperature range (recommended)

Minimum	40	C
Maximum	80	C

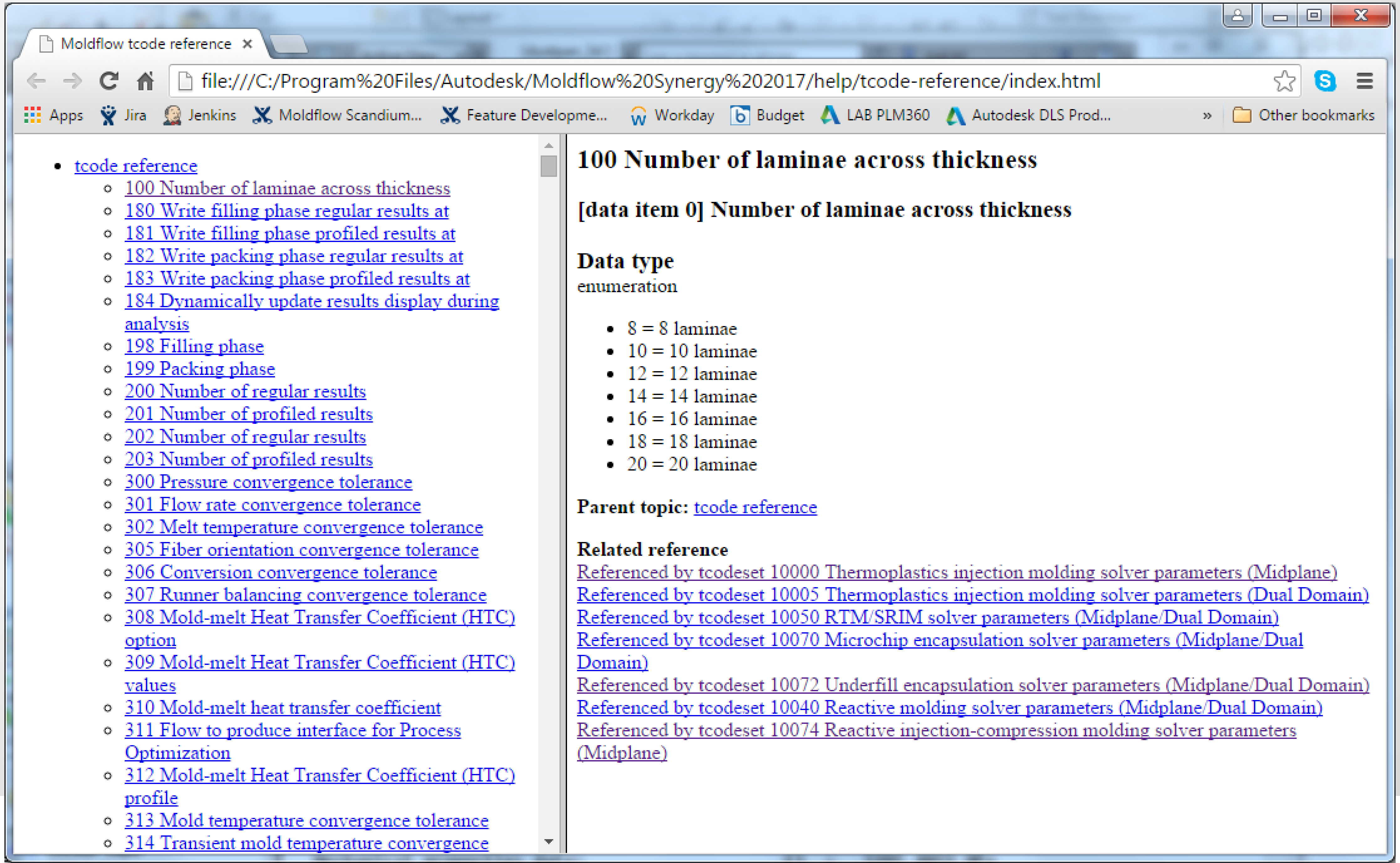
Understanding TCodes

- TCodes are the basic data store blocks
- See Online help: tcode and tcodeset reference
  - Comprehensive help
  - Shows Tcodeset (TSET) and Tcode (TCOD) hierarchy



Understanding TCodes

- TCode Reference



- 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

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)

Understanding TCodes: tcodes.dat

- TCOD Data

```
 TCOD
//   { tcodeNumber tcodeDescription tcodeDataRepeat
//    DATA
//    {   tcoodeDataSymbol tcodeDataFormat unitNameInSI unitConversionFlag
//        recommendMinimum recommendMinFlag recommendMaximum recommendMaxFlag
//        physicalMinimum  physicalMinFlag  physicalMaximum  physicalMaxFlag
//        increment
//    }
//    REFT
//    {
//        tcodeSetType tcodeSetType tcodeSetType ...
//    }
```

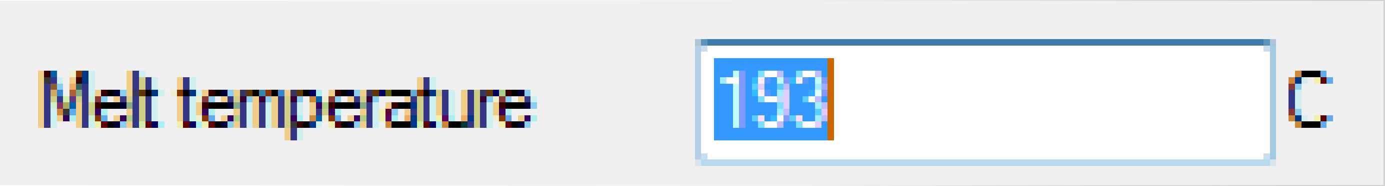
  - Note: Can also contain UI controls TCUI,TCUI2

Tcodes.dat   Edit

Understanding TCodes: tcodes.dat

- Single Value

```
TCOD
{ 11002 "Melt temperature" 0
 DATA
 {
      "Tmelt" "%12.5g" "K" 1
      0.0 0 1.0e+38 0
      -1.0e+38 0 1.0e+38 0
   0.
 }
}
```



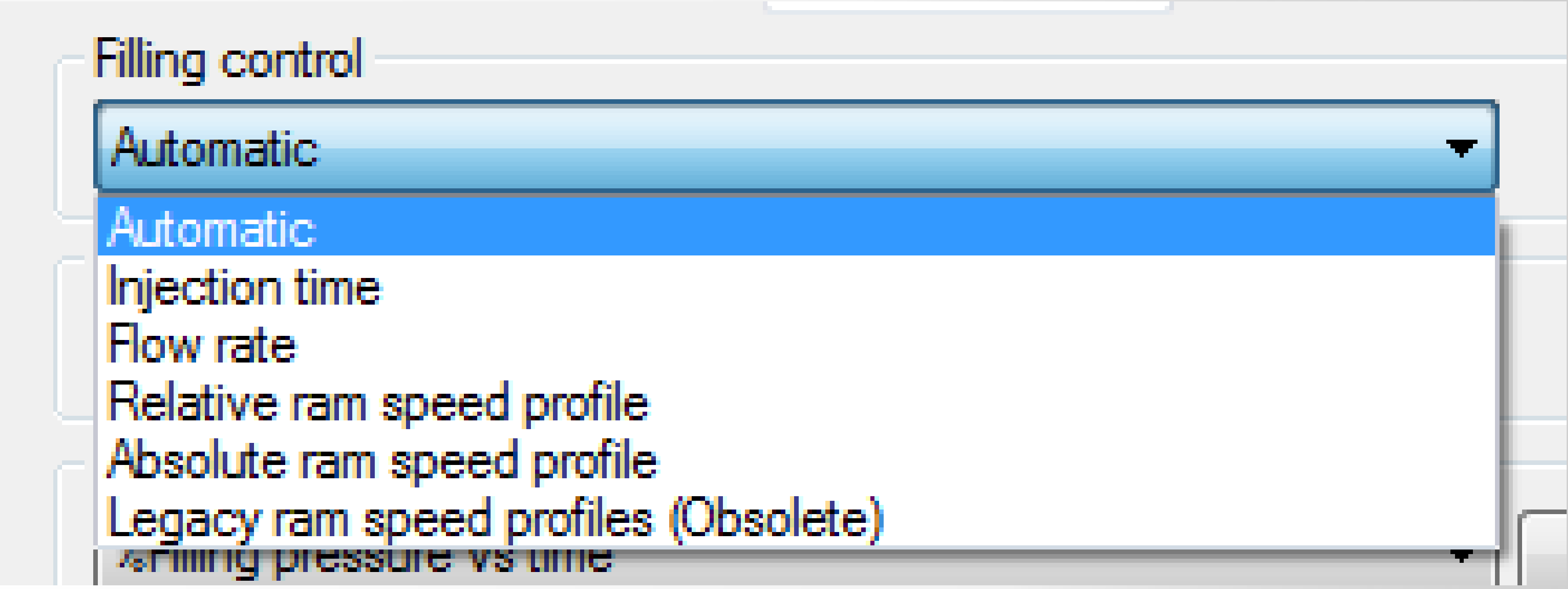
Melt temperature 193 C

```
TCOD
{ 11002 "Melt temperature“
4.66170E+02
}
```

Tcodes.dat   Edit

Understanding TCodes: tcodes.dat

- Menu



Filling control

Automatic

Automatic
Injection time
Flow rate
Relative ram speed profile
Absolute ram speed profile
Legacy ram speed profiles (Obsolete)

```
TCOD
{ 10109 "Filling control" 0
 DATA
 {
       "Filling control" "|1|Automatic[DEFAULT]|1|Injection time|2|Flow rate|3|Relative ram speed profile|5|Absolute ram speed profile|6|Legacy ram speed profiles (Obsolete)|4||" "" 0
       1 0 6 0
       1 0 6 0
     0.
 }
}
```

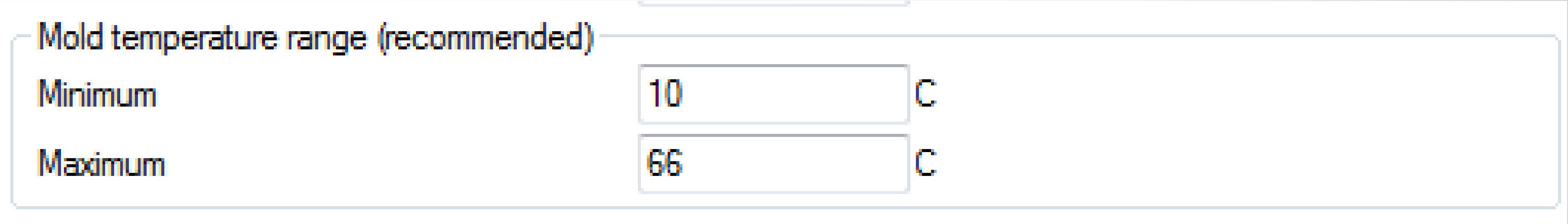
TCOD {10109 "Filling control" 1 }

Tcodes.dat   Edit

Understanding TCodes: tcodes.dat

- Multiple Values

```
TCOD
{ 1800 "Melt temperature range (recommended)" 0
 DATA
 {
      "Minimum" "%12.5g" "K" 1
      373.15 0  673.15 0
      273.15 1 1273.15 1
   0.
 }
 DATA
 {
      "Maximum" "%12.5g" "K" 1
      373.15 0  673.15 0
      273.15 1 1273.15 1
   0.
 }
}
```



Mold temperature range (recommended)

Minimum 10 C

Maximum 66 C

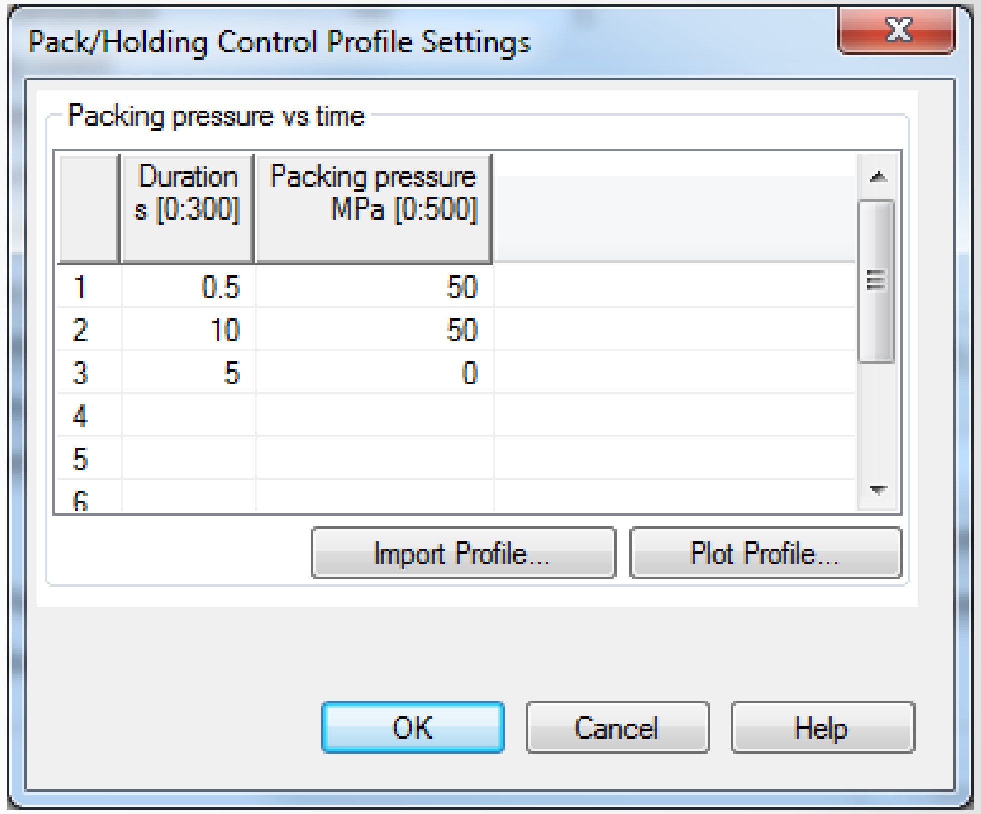
TCOD {1800 "Melt temperature range (recommended)" 2.831500E+02 3.391500E+02 }

Tcodes.dat   Edit

Understanding TCodes: tcodes.dat

- Repeated Data

```
TCOD
{ 10707 "Packing pressure vs time" 1
 DATA
 {
      "Duration" "%11.4g" "s" 1
      0. 0   300. 0
      0. 0   300. 0
    0.
 }
 DATA
 {
      "Packing pressure" "%12.6g" "Pa" 1
             0. 0 5.0e+08 0
           0. 0 5.0e+08 0
    0.
 }
}
```



Pack/Holding Control Profile Settings

Packing pressure vs time

	Duration s [0:300]	Packing pressure MPa [0:500]
1	0.5	50
2	10	50
3	5	0
4		
5		
6		

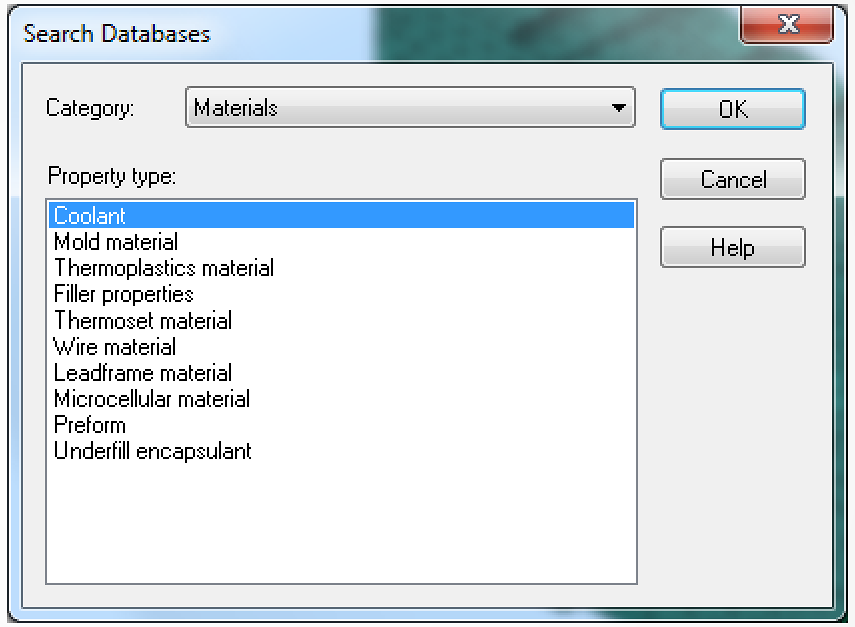
```
TCOD
{ 10707 "Packing pressure vs time"
5.00000E-01 5.0000E+07 1.00000E+01
5.0000E+07 5.00000E+00 0.00000E+00
}
```

Tcodes.dat   Edit

TCodesets (TSET)

Understanding TSETs

- TSETs contain TCode data
  - Material
  - Parameters
  - Process Conditions
  - Geometry/Mesh/BC

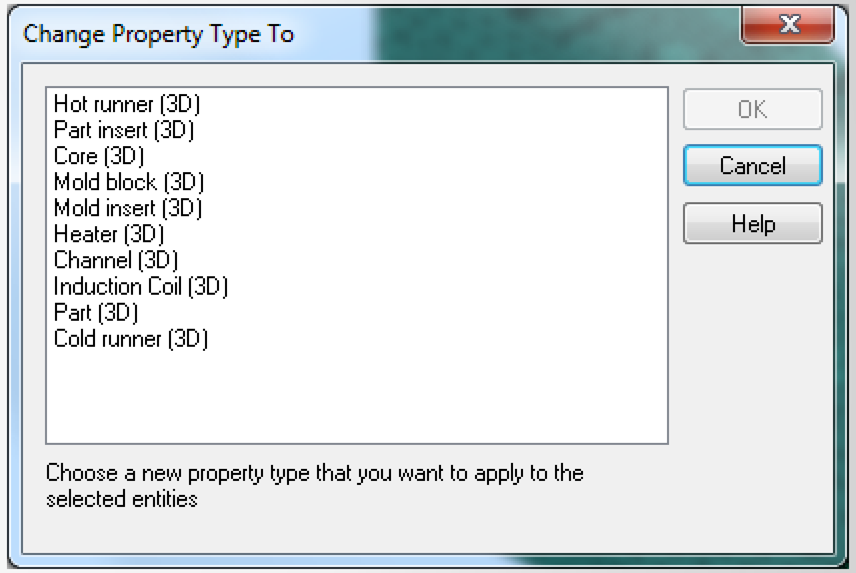


Search Databases

Category: Materials

Property type:

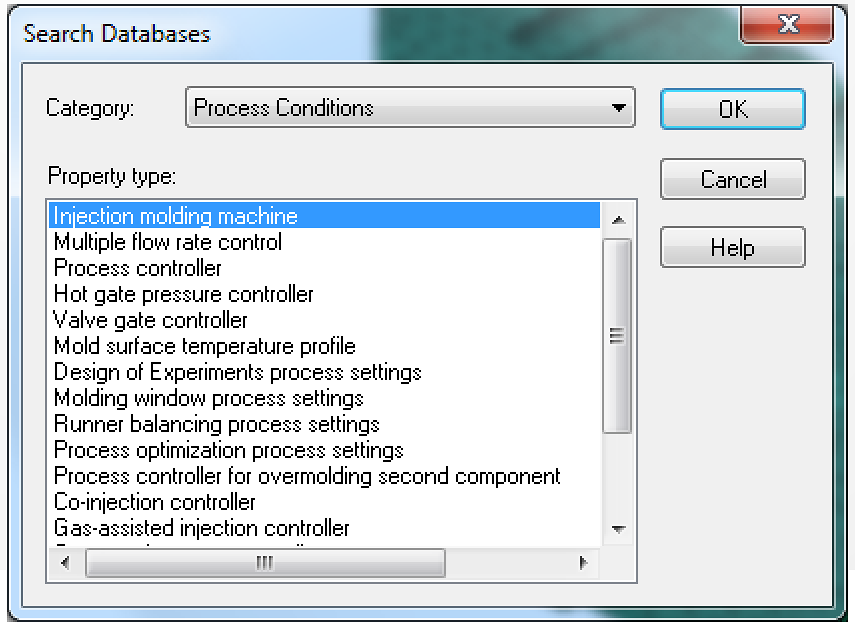
- Coolant
- Mold material
- Thermoplastics material
- Filler properties
- Thermoset material
- Wire material
- Leadframe material
- Microcellular material
- Preform
- Underfill encapsulant



Change Property Type To

- Hot runner (3D)
- Part insert (3D)
- Core (3D)
- Mold block (3D)
- Mold insert (3D)
- Heater (3D)
- Channel (3D)
- Induction Coil (3D)
- Part (3D)
- Cold runner (3D)

Choose a new property type that you want to apply to the selected entities

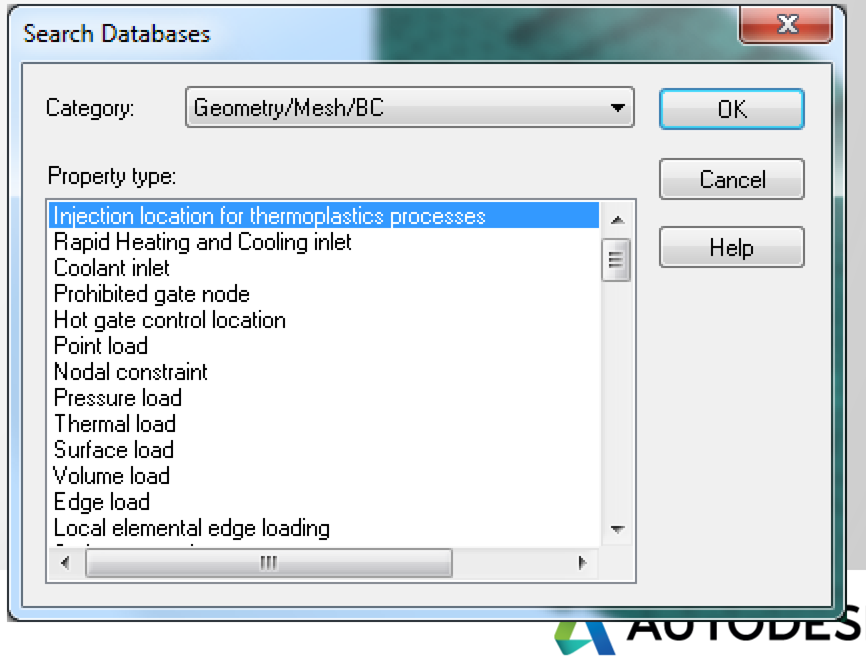


Search Databases

Category: Process Conditions

Property type:

- Injection molding machine
- Multiple flow rate control
- Process controller
- Hot gate pressure controller
- Valve gate controller
- Mold surface temperature profile
- Design of Experiments process settings
- Molding window process settings
- Runner balancing process settings
- Process optimization process settings
- Process controller for overmolding second component
- Co-injection controller
- Gas-assisted injection controller



Search Databases

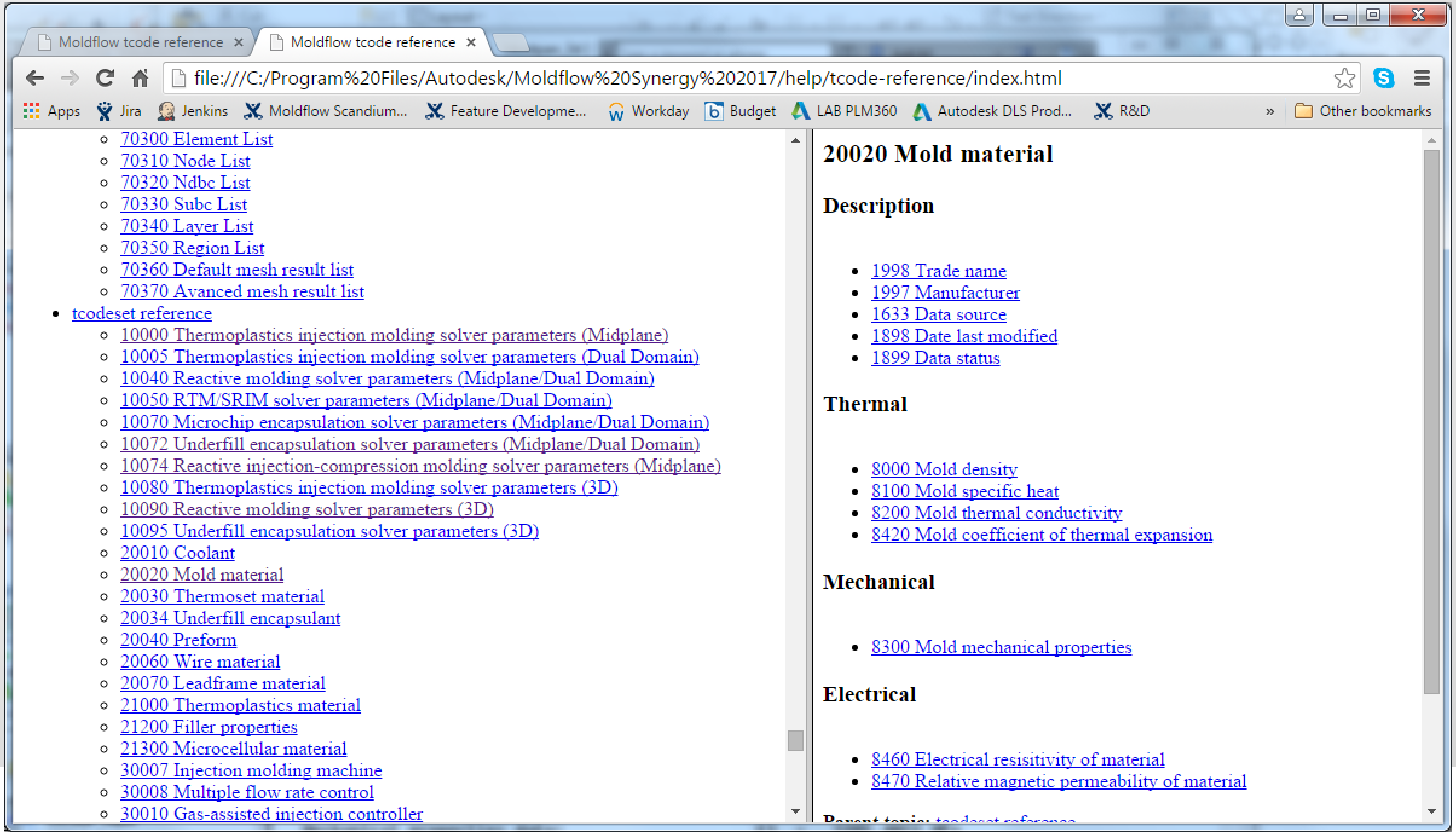
Category: Geometry/Mesh/BC

Property type:

- Injection location for thermoplastics processes
- Rapid Heating and Cooling inlet
- Coolant inlet
- Prohibited gate node
- Hot gate control location
- Point load
- Nodal constraint
- Pressure load
- Thermal load
- Surface load
- Volume load
- Edge load
- Local elemental edge loading

Understanding TSETs

- TCodeset Reference



- 70300 Element List
- 70310 Node List
- 70320 Ndbc List
- 70330 Subc List
- 70340 Layer List
- 70350 Region List
- 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

20020 Mold material

Description

- 1998 Trade name
- 1997 Manufacturer
- 1633 Data source
- 1898 Date last modified
- 1899 Data status

Thermal

- 8000 Mold density
- 8100 Mold specific heat
- 8200 Mold thermal conductivity
- 8420 Mold coefficient of thermal expansion

Mechanical

- 8300 Mold mechanical properties

Electrical

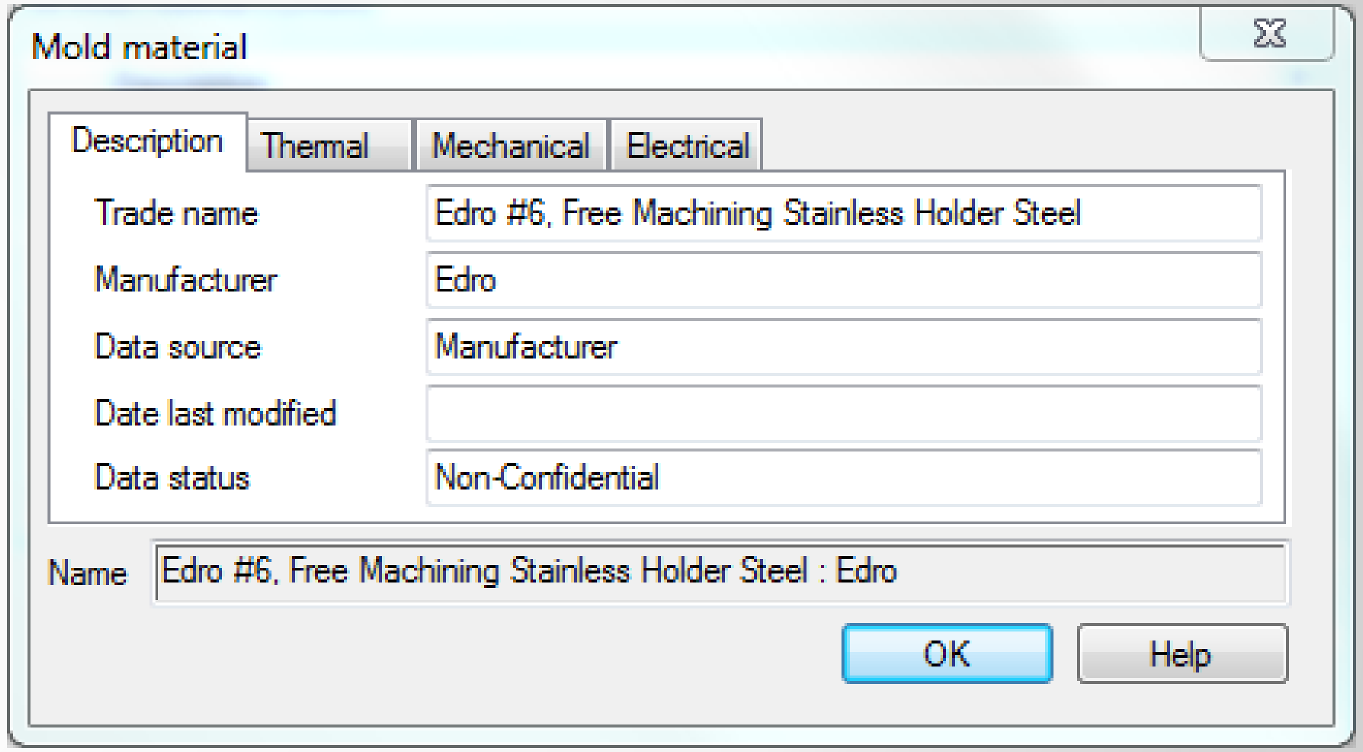
- 8460 Electrical resisitivity of material
- 8470 Relative magnetic permeability of material

Parent topic: tcodeset reference

Understanding TSETs

- Mold Material (ID = 20020)

```
TSET
{ 20020 "Mold material"
    TCOD
    {
        1998 1997 1633 1898 1899
        8000 8100 8200 8420
        8300  8460 8470
    }
    UITA { "Description" 1998 1997 1633 1898 1899 }
    UITA { "Thermal    " 8000 8100 8200 8420 }
    UITA { "Mechanical" 8300      }
    UITA { "Electrical" 8460 8470 }
}
```



Trade name	Edro #6, Free Machining Stainless Holder Steel
Manufacturer	Edro
Data source	Manufacturer
Date last modified	
Data status	Non-Confidential

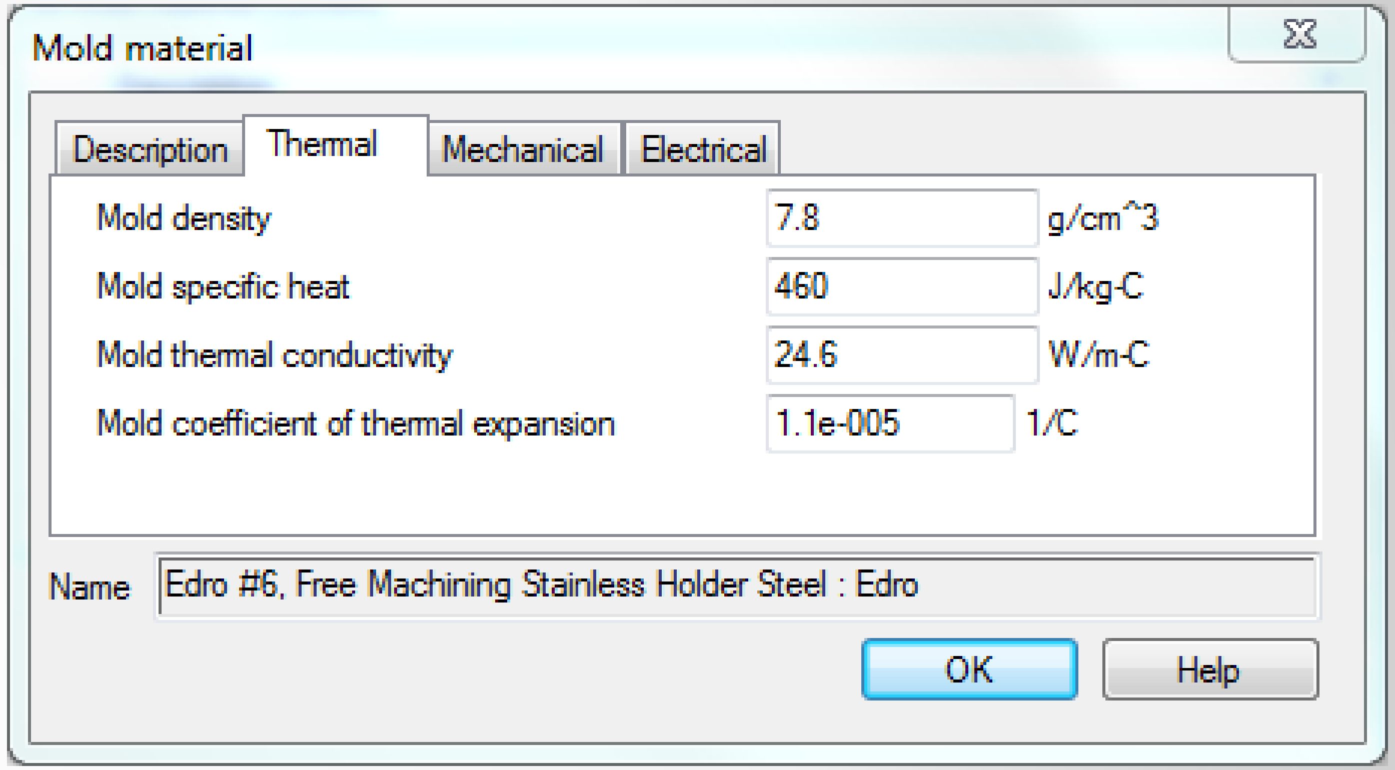
Name Edro #6, Free Machining Stainless Holder Steel : Edro

Tcodeset.dat Edit

Understanding TSETs

- Mold Material (ID = 20020)

```
TSET
{ 20020 "Mold material"
    TCOD
    {
        1998 1997 1633 1898 1899
        8000 8100 8200 8420
        8300  8460 8470
    }
    UITA { "Description" 1998 1997 1633 1898 1899 }
    UITA { "Thermal    " 8000 8100 8200 8420 }
    UITA { "Mechanical" 8300     }
    UITA { "Electrical" 8460 8470 }
}
```



Mold density	7.8	g/cm^3
Mold specific heat	460	J/kg-C
Mold thermal conductivity	24.6	W/m-C
Mold coefficient of thermal expansion	1.1e-005	1/C

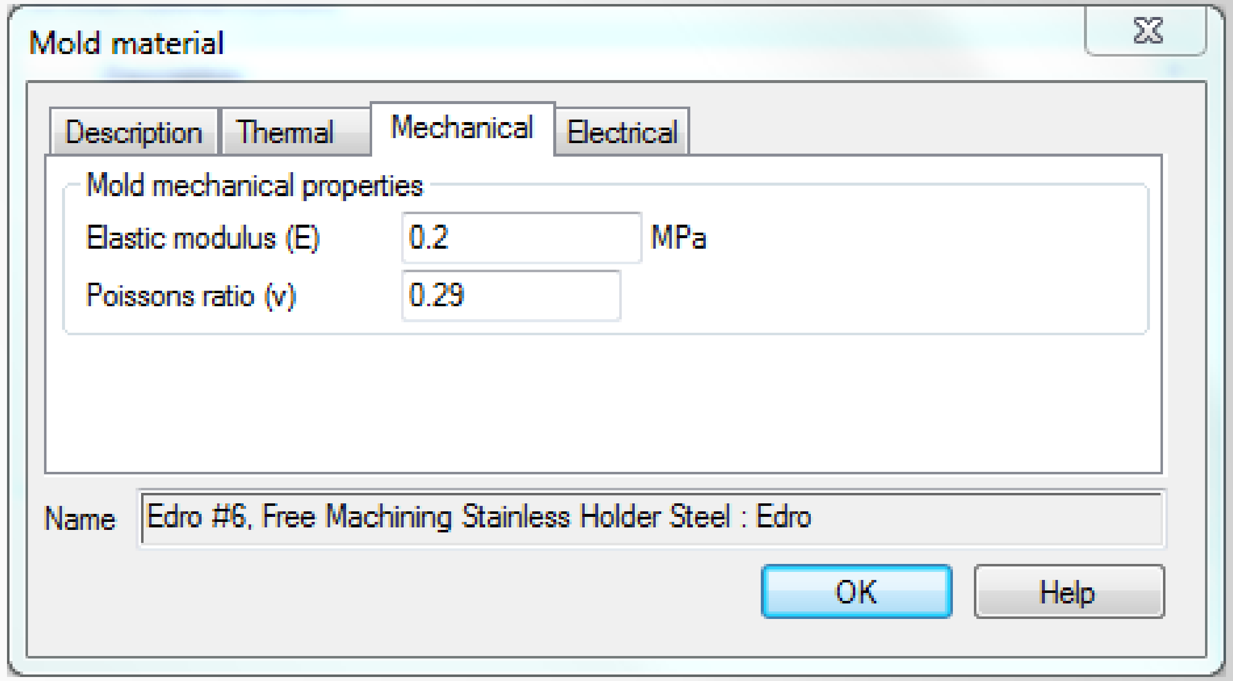
Name: Edro #6, Free Machining Stainless Holder Steel : Edro

Tcodeset.dat   Edit

Understanding TSETs

- Mold Material (ID = 20020)

```
TSET
{ 20020 "Mold material"
    TCOD
    {
        1998 1997 1633 1898 1899
        8000 8100 8200 8420
        8300  8460 8470
    }
    UITA { "Description" 1998 1997 1633 1898 1899 }
    UITA { "Thermal    " 8000 8100 8200 8420 }
    UITA { "Mechanical" 8300  }
    UITA { "Electrical" 8460 8470 }
}
```



Mold mechanical properties

Elastic modulus (E) 0.2 MPa

Poissons ratio (v) 0.29

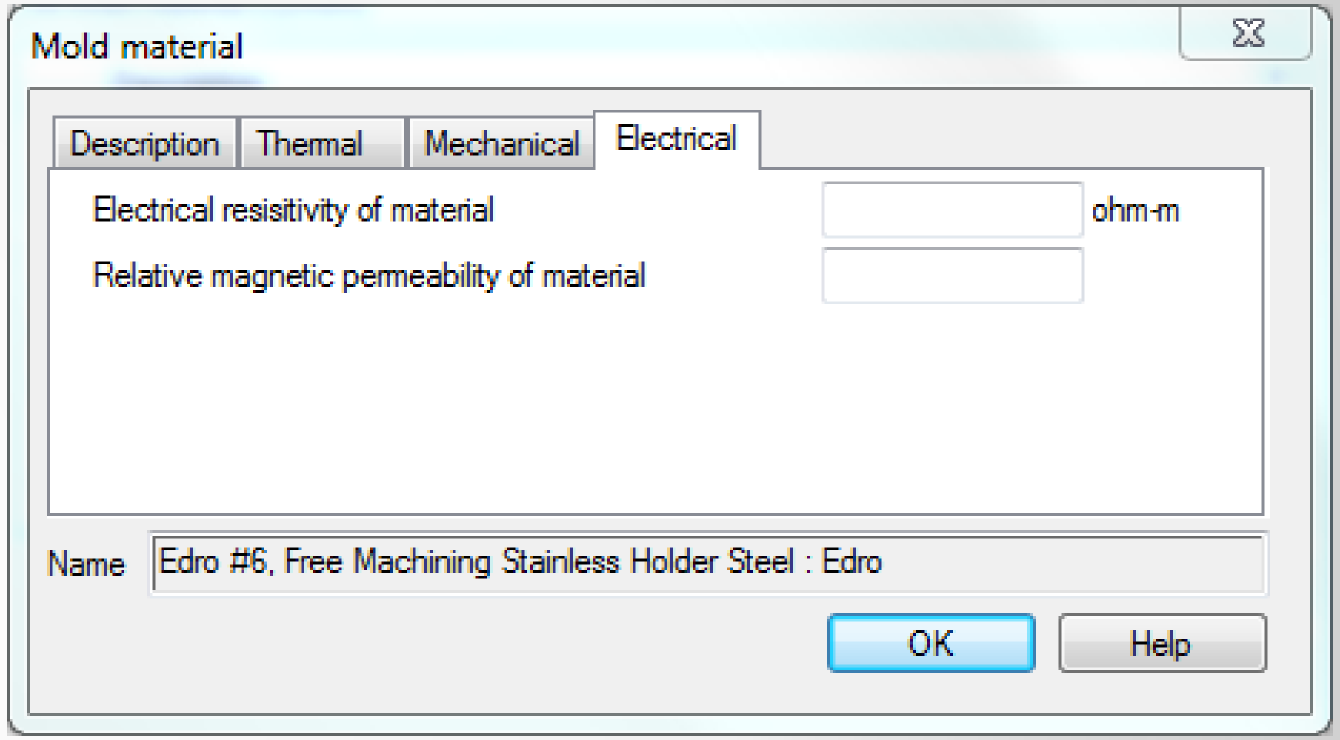
Name Edro #6, Free Machining Stainless Holder Steel : Edro

Tcodeset.dat Edit

Understanding TSETs

- Mold Material (ID = 20020)

```
TSET
{ 20020 "Mold material"
    TCOD
    {
        1998 1997 1633 1898 1899
        8000 8100 8200 8420
        8300  8460 8470
    }
    UITA { "Description" 1998 1997 1633 1898 1899 }
    UITA { "Thermal    " 8000 8100 8200 8420 }
    UITA { "Mechanical" 8300      }
    UITA { "Electrical" 8460 8470 }
}
```


Mold material
Description
Thermal
Mechanical
Electrical
Electrical resisitivity of material
ohm-m
Relative magnetic permeability of material
OK
Help

Name: Edro #6, Free Machining Stainless Holder Steel : Edro

Tcodeset.dat   Edit

How TSETs are stored in the Study File

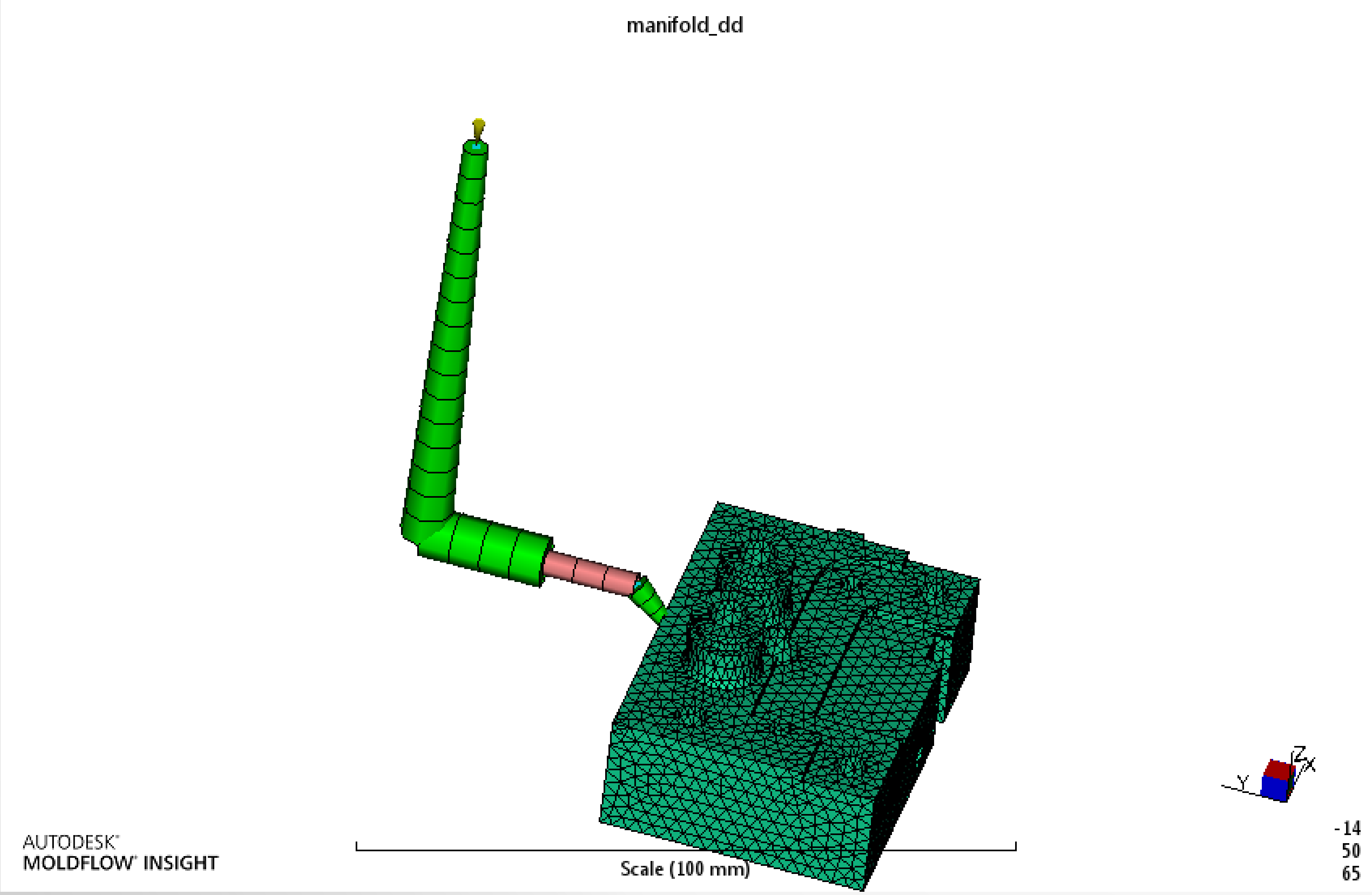
- Mold Material (ID = 20020)
  - What gets stored in the study file

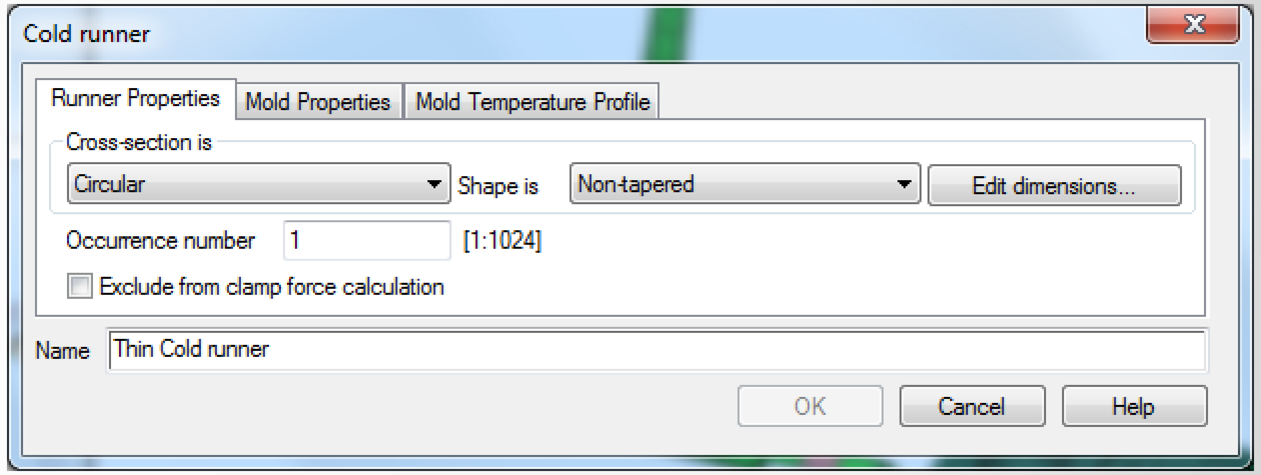
```
TSET { 20020  1  "Mold material"
    TCOD  { 1998 “” Value }
    TCOD  { 1997 “” Value }
    TCOD  { 1633 “” Value }
    TCOD  { 1898 “” Value }
    TCOD  { 1899 “” Value }
    TCOD  { 8000 “” Value }
    TCOD  { 8100 “” Value }
    TCOD  { 8200 “” Value }
    TCOD  { 8420 “” Value }
    TCOD  { 8300 “” Value }
    TCOD  { 8460 “” Value }
    TCOD  { 8470 “” Value }
}
```

Tcodeset.dat   Edit

How TSETs are stored in the Study File

- Geometry Properties





Cold runner

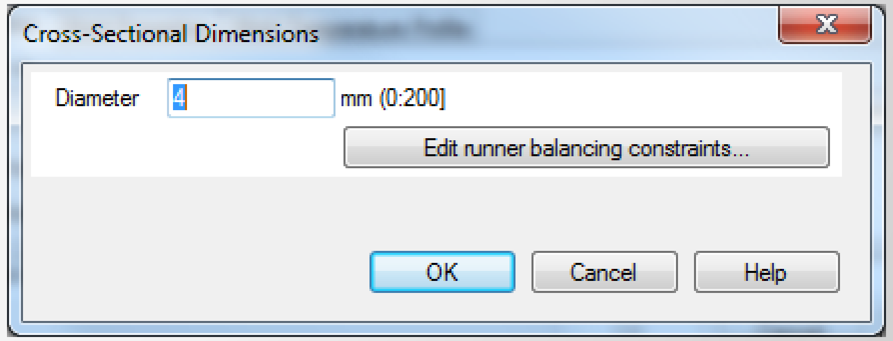
Cross-section is: Circular

Shape is: Non-tapered

Occurrence number: 1 [1:1024]

□ Exclude from clamp force calculation

Name: Thin Cold runner



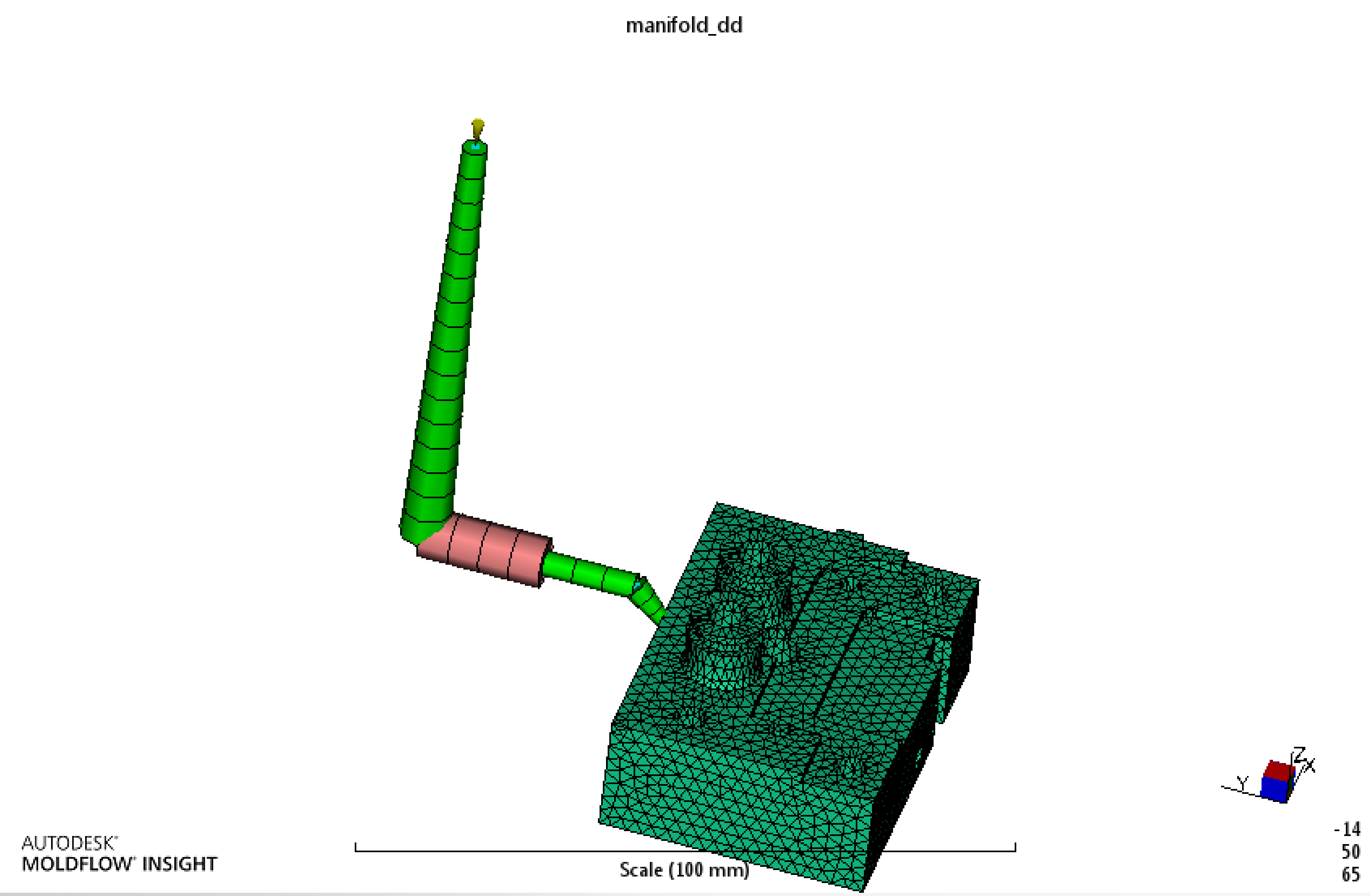
Cross-Sectional Dimensions

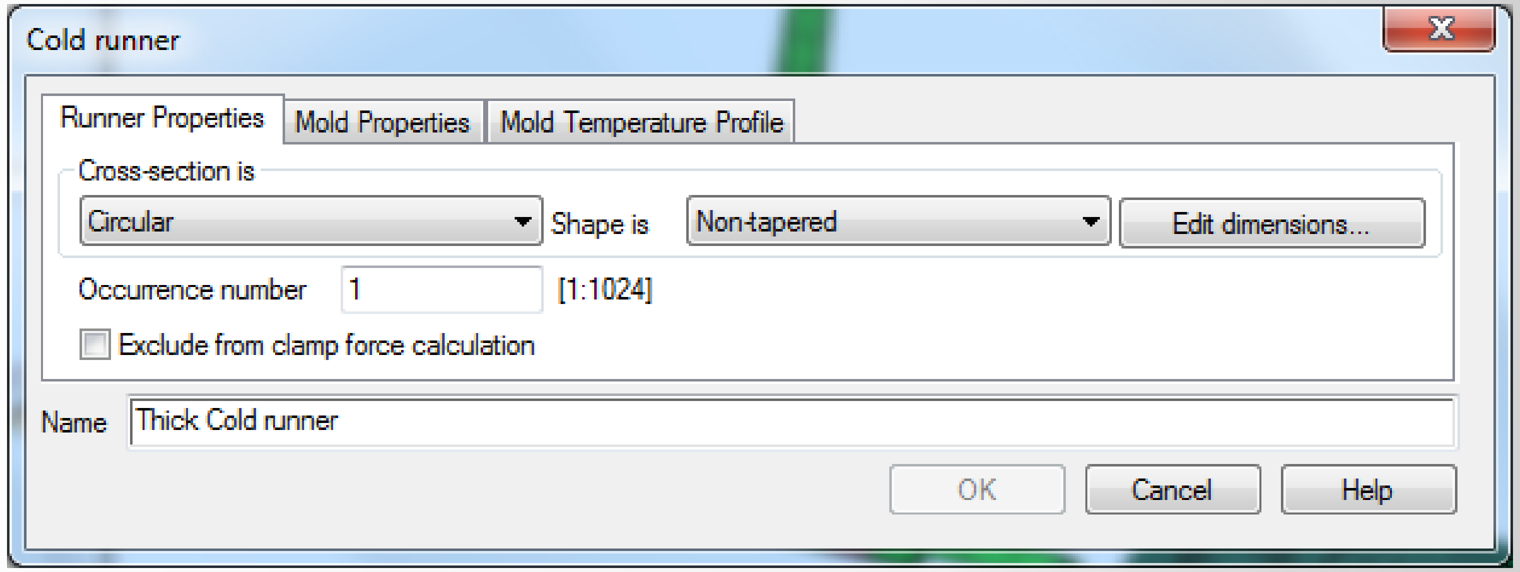
Diameter: 4 mm (0:200]

Tcodeset.dat Edit

How TSETs are stored in the Study File

- Geometry Properties




Cold runner

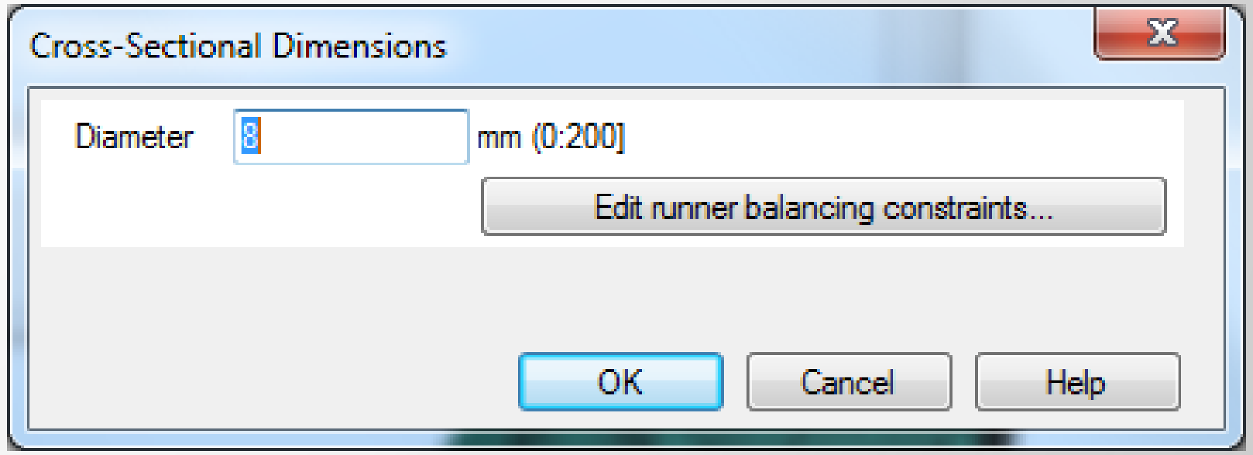
Cross-section is: Circular

Shape is: Non-tapered

Occurrence number: 1 [1:1024]

□ Exclude from clamp force calculation

Name: Thick Cold runner


Cross-Sectional Dimensions

Diameter: 8 mm (0:200]

How TSETs are stored in the Study File

- TSets are stored in Study File with an ID and Sub ID
- In our example we have two cold runners

```
TSET{40420 7 "Thick Cold runner "
TCOD{20023 ""}
TCOD{30107 ""    1.000000e+000}
....
TCOD{30260 ""    8.000000e-003}
......
}
```

```
TSET{40420 2 "Thick Cold runner "
TCOD{20023 ""}
TCOD{30107 ""    1.000000e+000}
....
TCOD{30260 ""    4.000000e-003}
......
}
```

Note: TSET 40420 Cold Runner   TCode 30260 = Diameter

How TSETs are stored in the Study File

- Why aren’t the SubIDs 1 and 2 ?
  - Why are the SubIDs 7 and 2?
    - Geometry can be added/deleted
    - Synergy does not renumber TSET data

```
TSET{40420 7 "Thick Cold runner "
TCOD{20023 ""}
TCOD{30107 ""    1.000000e+000}
….
TCOD{30260 ""    8.000000e-003}
……
}
```

```
TSET{40420 2 "Thick Cold runner "
TCOD{20023 ""}
TCOD{30107 ""    1.000000e+000}
….
TCOD{30260 ""    4.000000e-003}
……
}
```

Understanding the Study File

- NDBC (Nodal Boundary Condition) Data

```
NDBC{LABEL LABEL_ON LAYER COLOR DISPLAY TSET SUBID  NODE_LABEL
         TRAN{4x4_MATRIX} }
```

LABEL	Boundary condition number
LABEL_ON	Show node Label 0 = Off, 1 = On
LAYER	Layer Index
COLOR	Color Index
DISPLAY	
TSET	TSet type
SUBID	TSet Sub ID
NODE_LABEL	
TRAN	4x4 Matrix
NODE_LABEL	Node Number

Understanding the Study File

- NDBC (Nodal Boundary Condition) Data

NDBC{1 0 1 0 0 **40000** **1** 37 TRAN{ 0.000000e+000 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 -1.000000e+000 -0.000000e+000 0.000000e+000 1.000000e+000 0.000000e+000 -0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 1.000000e+000}}

Flow

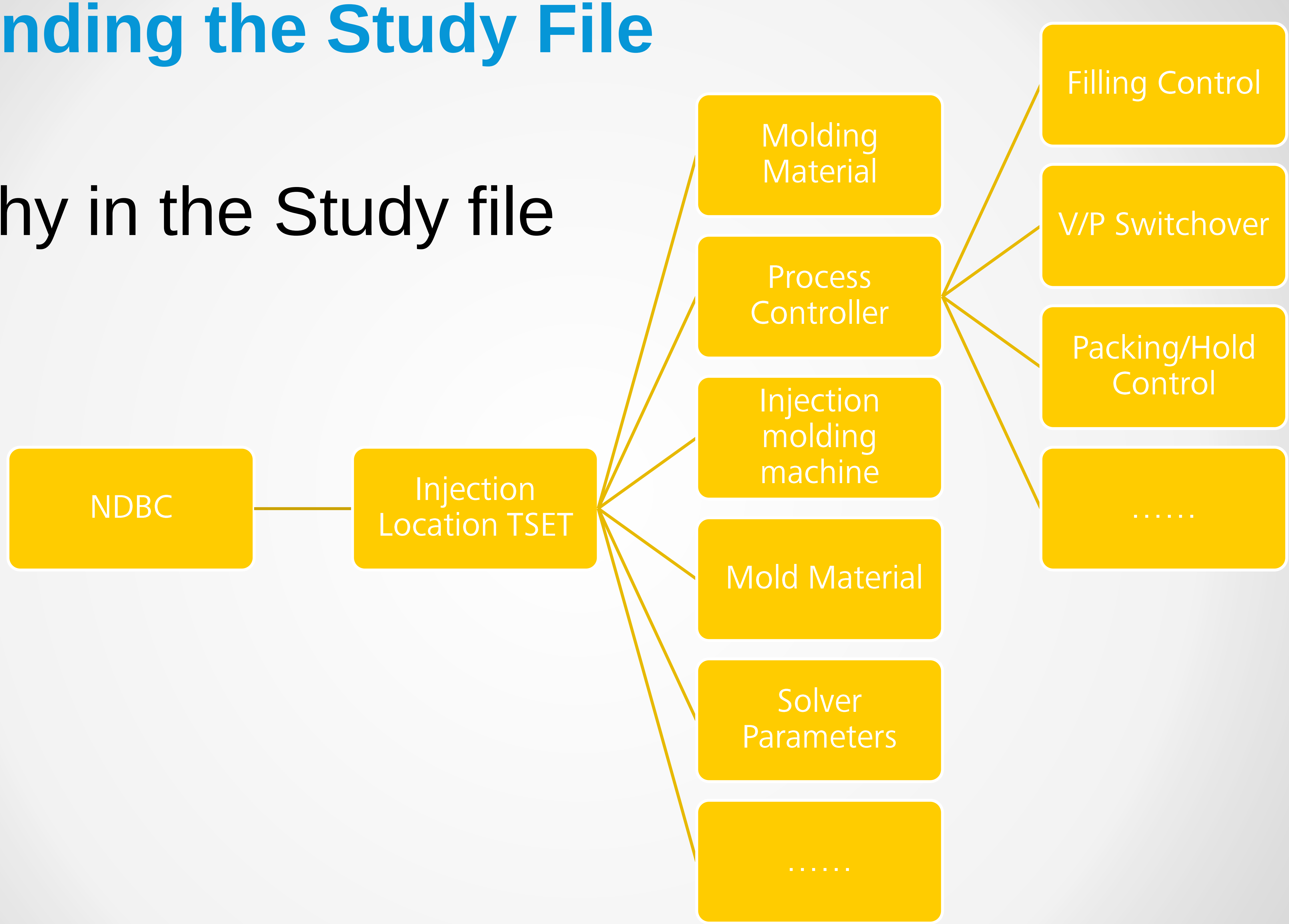
NDBC{1 0 12 0 0 **40000** **1** 5093 TRAN{ 0.000000e+000 0.000000e+000 1.000000e+000 0.000000e+000 -1.000000e+000 -0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 -1.000000e+000 -0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 1.000000e+000}}
NDBC{2 0 27 0 0 **40001** **2** 19169 TRAN{ 0.000000e+000 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 -1.000000e+000 -0.000000e+000 0.000000e+000 1.000000e+000 0.000000e+000 -0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 1.000000e+000}}

Overmolding

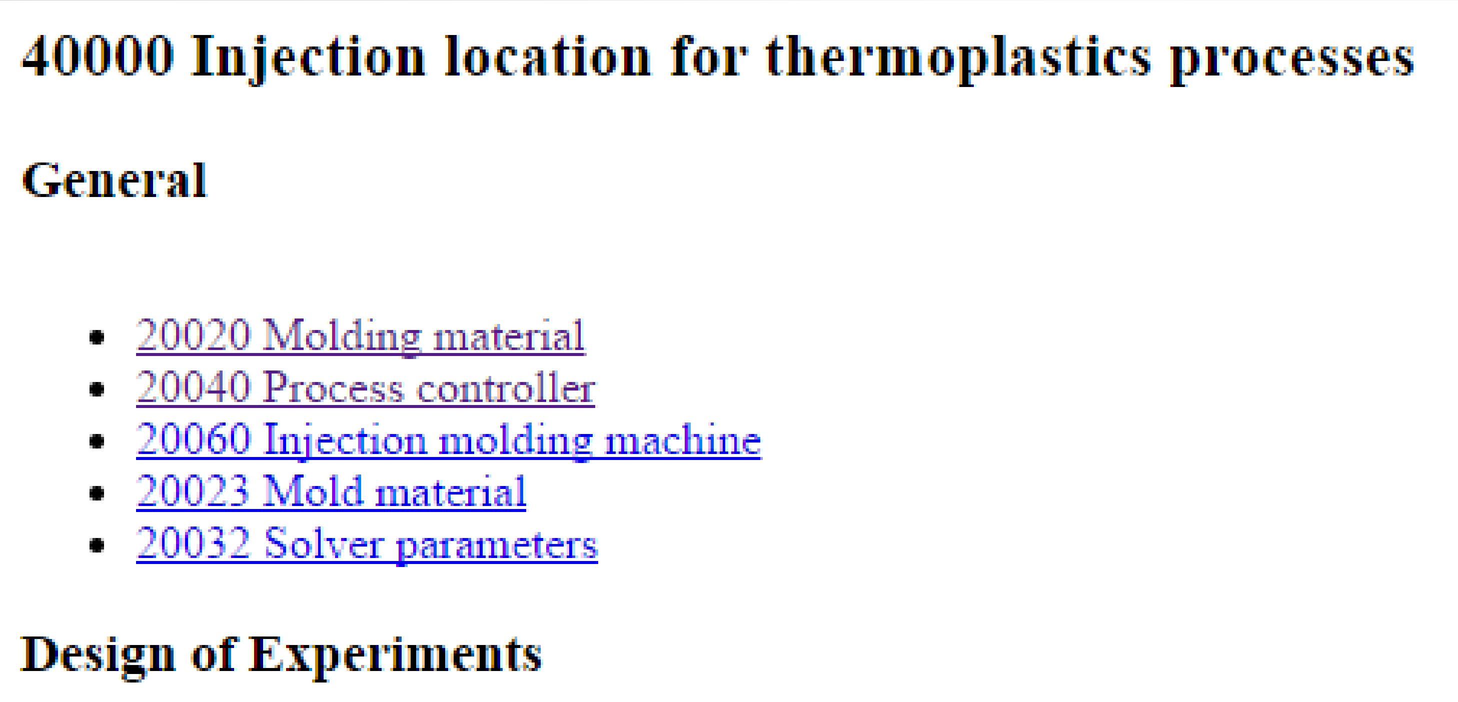
Example: Autodesk Moldflow Insight Model File

Understanding the Study File

- Hierarchy in the Study file



Understanding the Study File



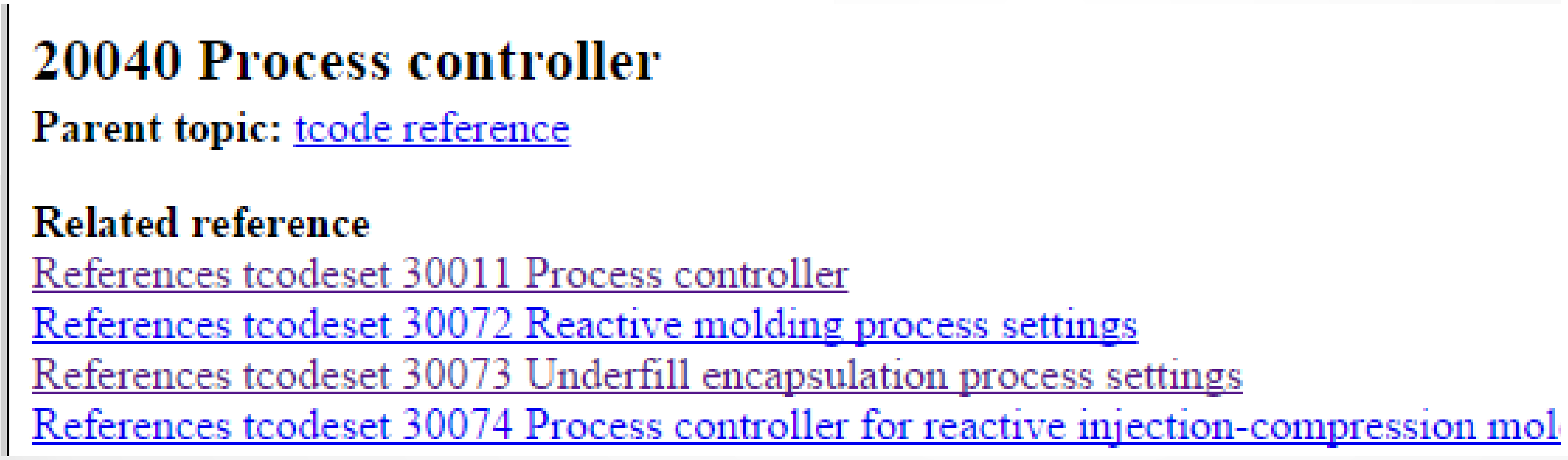
40000 Injection location for thermoplastics processes

General

- 20020 Molding material
- 20040 Process controller
- 20060 Injection molding machine
- 20023 Mold material
- 20032 Solver parameters

Design of Experiments

- 30510 DOE control

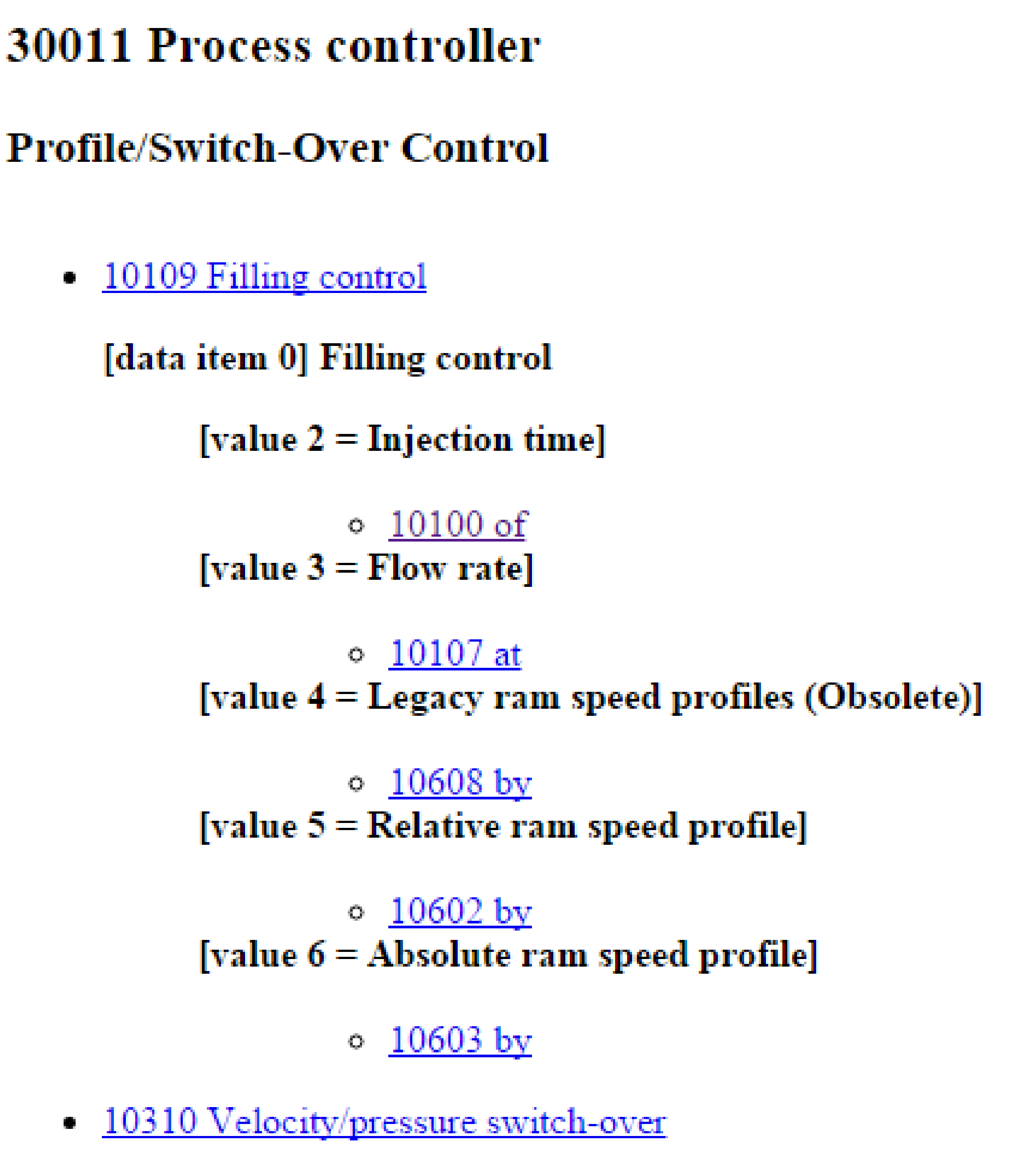


20040 Process controller

Parent topic: tcode reference

Related reference

References tcodeset 30011 Process controller
References tcodeset 30072 Reactive molding process settings
References tcodeset 30073 Underfill encapsulation process settings
References tcodeset 30074 Process controller for reactive injection-compression mol



30011 Process controller

Profile/Switch-Over Control

- 10109 Filling control

  [data item 0] Filling control

  [value 2 = Injection time]

  - 10100 of

  [value 3 = Flow rate]

  - 10107 at

  [value 4 = Legacy ram speed profiles (Obsolete)]

  - 10608 by

  [value 5 = Relative ram speed profile]

  - 10602 by

  [value 6 = Absolute ram speed profile]

  - 10603 by

- 10310 Velocity/pressure switch-over

Note: The file ***process.dat*** defines the content of each injection location for a given mesh type, molding process and analysis sequence

process.dat Edit

How to succeed with TSet and TCode data

- Reuse the following Subroutines and Functions
  - To ensure you are working with the correct TSet
    - Sub GetInjectionData (InjectionD, InjectionSubID, Injection2ID, Injection2SubID)
    - Sub GetProcessData(ProcessID, ProcessSubID, Process2ID, Process2SubID)
    - Sub GetMaterialData(MaterialID, MaterialSubID, Material2ID, Material2SubID)
  - To Read/Write TCode Data
    - Function GetTCodeValue(ID, SubID, TCode, Value)
    - Function SetTCodeValue(ID, SubID, TCode, Value)
    - Function GetTCodeUnit(ID, SubID, TCode, UnitStr)
    - Function GetTCodeDescription(ID, SubID, TCode, Value)
    - Function GetTCodeName(ID, TCode, Value)

Eight Functions to manipulate TSet and TCode data

Subroutine name: GetInjectionData

Purpose

Identify the injection TSet, taking into account the mesh type, analysis sequence and molding process

Usage

Sub GetInjectionData(InjectionID, InjectionSubID, Injection2ID, Injection2SubID)

Arguments/Return Values

- *InjectionID* TSet for first Injection location (0 = Not Found)
- *InjectionSubID* TSet Sub ID for first Injection location (0 = Not Found)
- *Injection2ID* TSet for second Injection location (0 = Not Found)
- *Injection2SubID* TSet Sub ID for second Injection location (0 = Not Found)

Eight Functions to manipulate TSet and TCode data

Subroutine name: GetInjectionData

Example

```
' Find Appropriate Injection Tsets and TSetSubID’s
Dim InjectionID, InjectionSubID, Injection2ID, Injection2SubID
Call GetInjectionData (InjectionID, InjectionSubID, Injection2ID, Injection2SubID)
```

Eight Functions to manipulate TSet and TCode data

Subroutine name: GetProcessData

Purpose

Identify the process controller TSet, taking into account the mesh type, analysis sequence and molding process

Usage

Sub GetProcessData(ProcessID, ProcessSubID, Process2ID, Process2SubID)

Arguments/Return Values

ProcessID	TSet for first process controller (0 = Not Found)
ProcessSubID	TSet Sub ID first process controller (0 = Not Found)
Process2ID	TSet for second process controller (0 = Not Found)
Process2SubID	TSet Sub ID for second process controller (0 = Not Found)

Eight Functions to manipulate TSet and TCode data

Subroutine name: GetProcessData

Example

```
' Find Appropriate Process TSets and TSetSubID's
Dim ProcessID, ProcessSubID, Process2ID, Process2SubID
Call GetProcessData (ProcessID, ProcessSubID, Process2ID, Process2SubID)
```

Eight Functions to manipulate TSet and TCode data

Subroutine name: GetMaterialData

Purpose

Identify the material data TSet, taking into account the mesh type, analysis sequence and molding process

Usage

Sub GetMaterialData(MaterialID, MaterialSubID, Material2ID, Material2SubID)

Arguments/Return Values

MaterialID	TSet for first Material (0 = Not Found)
MaterialSubID	TSet Sub ID first Material (0 = Not Found)
Material2ID	TSet for second Material (0 = Not Found)
Material2SubID	TSet Sub ID for second Material (0 = Not Found)

Eight Functions to manipulate TSet and TCode data

Subroutine name: GetMaterialData

Example

```
' Find Appropriate Material TSets and TSetSubID's
Dim MaterialID, MaterialSubID, Material2ID, Material2SubID
Call GetMaterialData (MaterialID, MaterialSubID, Material2ID, Material2SubID)
```

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeValue

Purpose

Read TCode data from a specified TSet ID and SubID

Usage

- *Function GetTCodeValue(ID, SubID, TCode, Value)*

Arguments

ID	Tset ID
SubID	Tset Sub ID
TCode	TCode we are trying to read data from
Value	a DoubleArray to hold values

Return Values

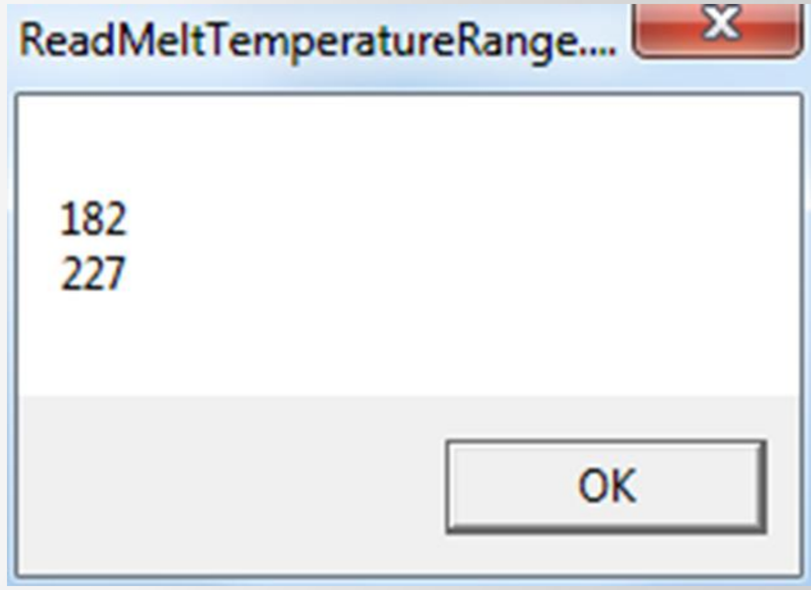
GetTCodeValue	false = failure, true = success
Value	The Tcode data returned as a DoubleArray

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeValue

Example

```
Dim MaterialID, MaterialSubID, Material2ID, Material2SubID
Call GetMaterialData(MaterialID, MaterialSubID, Material2ID, Material2SubID)
' Read  Melt Temperature range for First Material
If MaterialID > 0 Then
        lStr = ""
        Set Value = Synergy.CreateDoubleArray()
        OK = GetTCodeValue(MaterialID, MaterialSubID, 1800, Value)
        If OK Then
                For I = 0 To Value.Size()-1
                        lStr = lStr & Value.Val(I) & vbCrLf
                Next
                MsgBox  lStr,,WScript.ScriptName
        End if
End If
```



```
182
227
```

Eight Functions to manipulate TSet and TCode data

Function name: SetTCodeValue

Purpose

Set data for a specified TSet ID and SubID and TCode

Usage

- *Function SetTCodeValue(ID, SubID, TCode, Value)*

Arguments

ID	Tset ID
SubID	Tset Sub ID
TCode	TCode we are trying to read data from
Value	a DoubleArray to hold values

Return Values

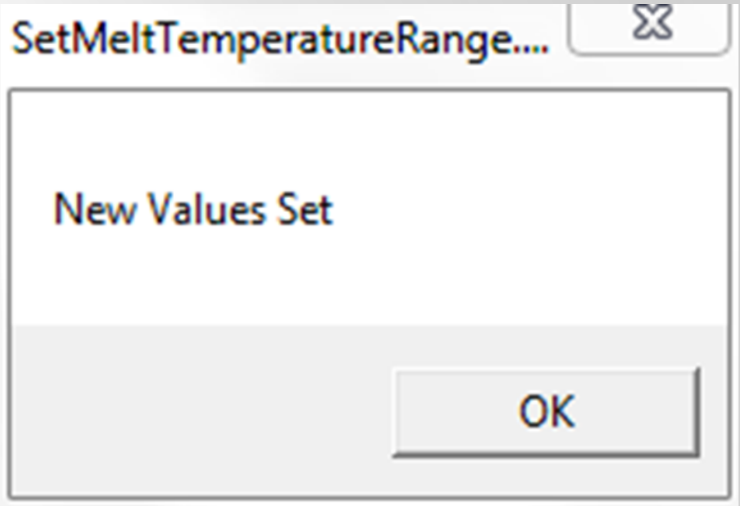
SetTCodeValue	false = failure, true = success

Eight Functions to manipulate TSet and TCode data

Function name: SetTCodeValue

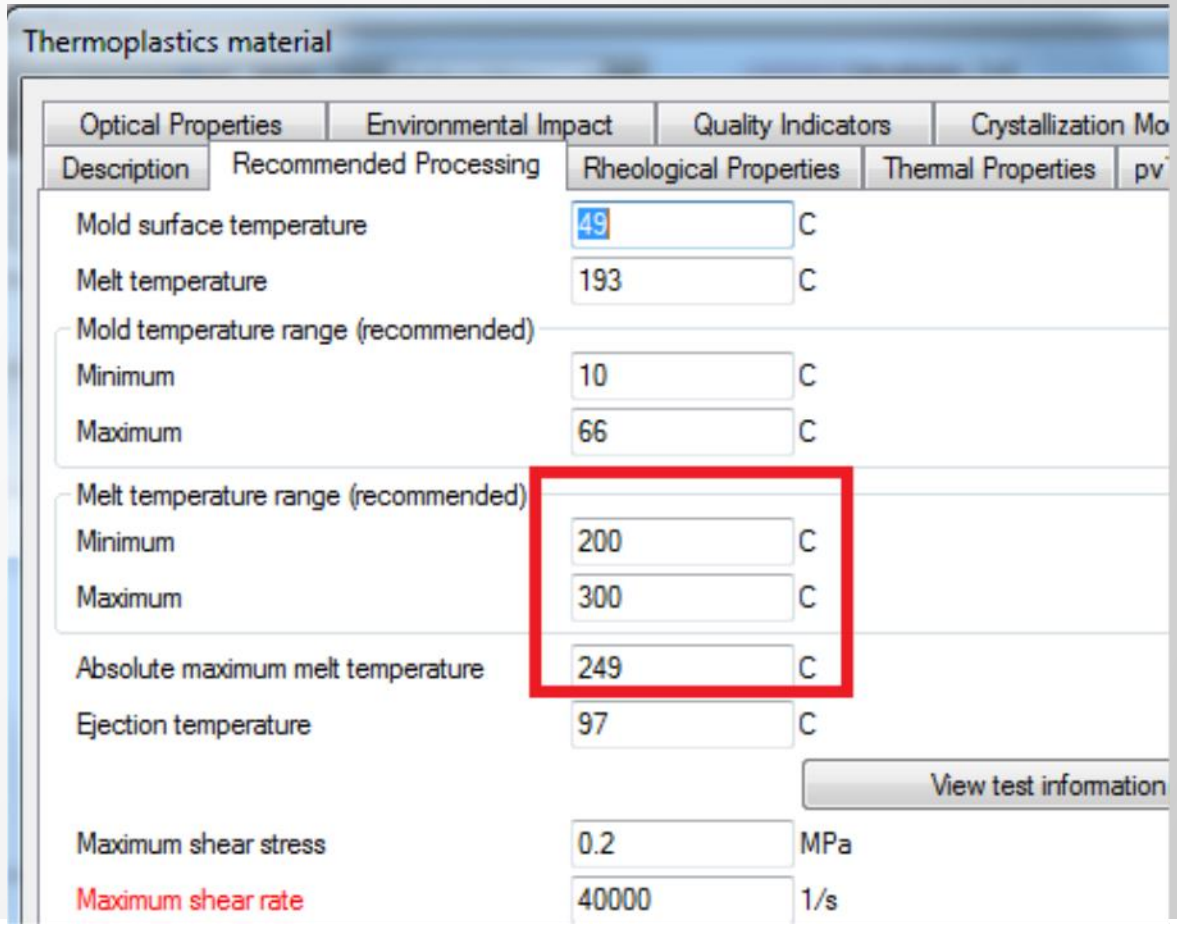
Example

```
Call GetMaterialData(MaterialID, MaterialSubID, Material2ID, Material2SubID)
' Set Melt Temperature range for First Material
Dim  OK, Value
If MaterialID > 0 Then
          Set Value = Synergy.CreateDoubleArray()
          Value.AddDouble(200)
          Value.AddDouble(300)
          OK = SetTCodeValue(MaterialID, MaterialSubID, 1800, Value)
          If OK Then
                    MsgBox  "New Values Set",,WScript.ScriptName
          End if
End If
```



SetMeltTemperatureRange....

New Values Set



Thermoplastics material

Mold surface temperature	49	C
Melt temperature	193	C
Mold temperature range (recommended)		
Minimum	10	C
Maximum	66	C
Melt temperature range (recommended)		
Minimum	200	C
Maximum	300	C
Absolute maximum melt temperature	249	C
Ejection temperature	97	C
Maximum shear stress	0.2	MPa
Maximum shear rate	40000	1/s

Eight Functions to manipulate TSet and TCode data

```
TCOD
{ 100 "Number of laminae across thickness" 0
 DATA
 {
       "n" "|1|8 laminae|8|10 laminae|10|12 laminae|12|14 laminae|14|16 laminae|16|18 laminae|18|20 laminae|20||" "" 0
       8. 0 20. 0
       8. 0 20. 0
    0.
 }
}
```

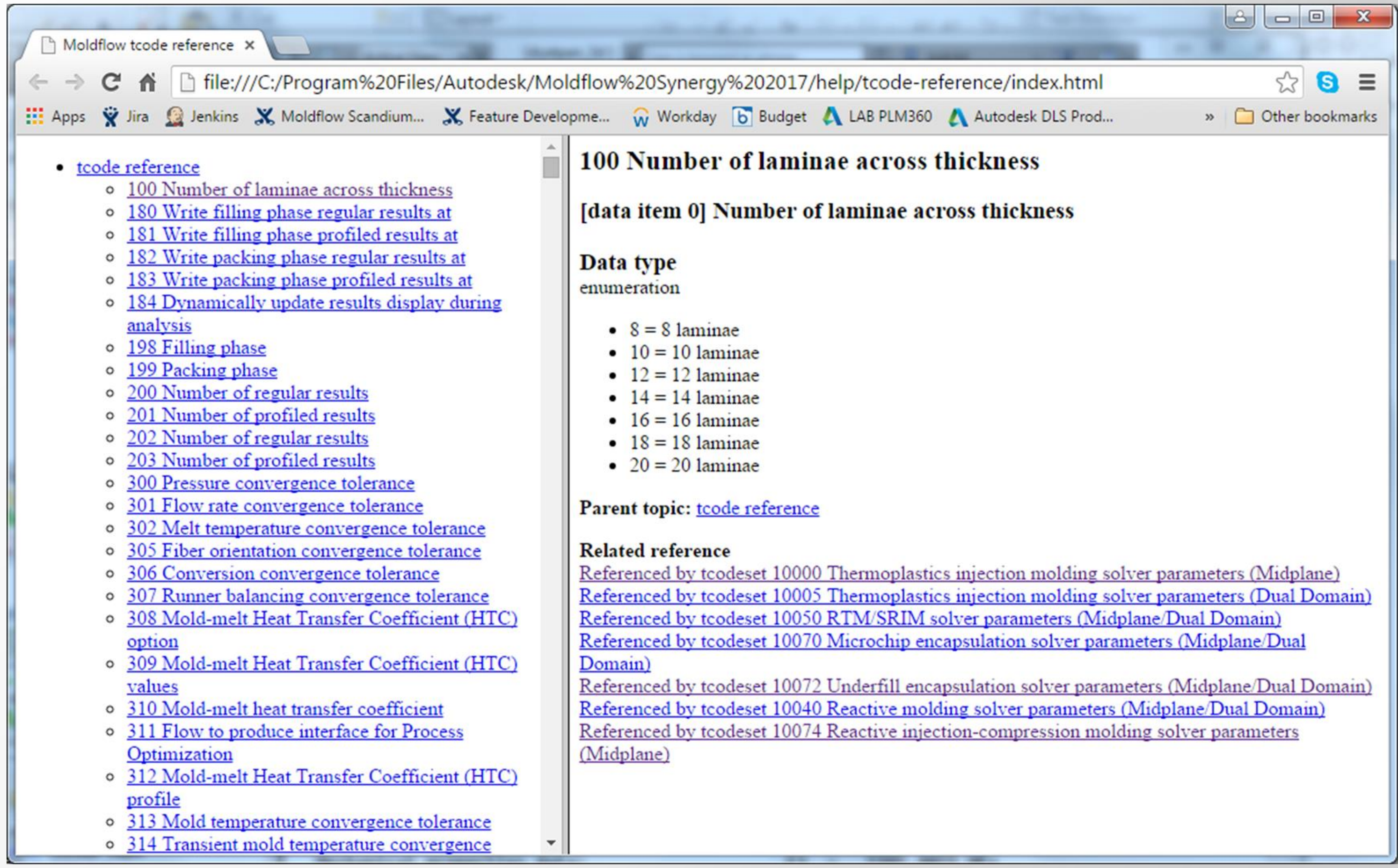
Function name: SetTCodeValue

Notes

The API does no explicit checking of the data provided

Example: Number of laminates across thickness

- Current Range [8-20]
- We can use the API to set a value of 40
  - More Laminates = Better accuracy
- What Happens
  - Solver Crashes
  - UI generate an error when study read
    - As TCode value outside the allowed range



- 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

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)

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeUnit

Purpose

Read unit data for a specified TSet ID, SubID and TCode

Usage

- *Function GetTCodeUnit(ID, SubID, TCode, UnitStr)*

Arguments

ID	Tset ID
SubID	Tset Sub ID
TCode	TCode we are trying to read data from
UnitStr	a StringArray to hold values

Return Values

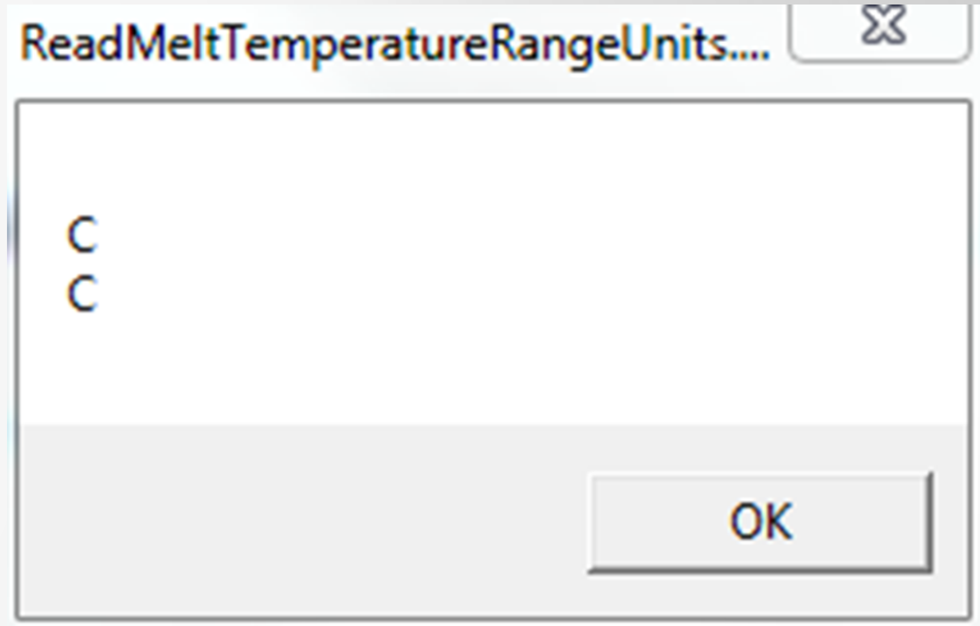
GetTCodeUnit	false = failure, true = success
UnitStr	The Unit String data returned as a StringArray

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeUnit

Example

```
Call GetMaterialData(MaterialID, MaterialSubID, Material2ID, Material2SubID)
' Read Units for Melt Temperature range for First Material
If MaterialID > 0 Then
        lStr = " "
        Set UnitStr = Synergy.CreateDoubleArray()
        OK = GetTCodeUnit(MaterialID, MaterialSubID, 1800, UnitStr)
        If OK Then
                For I = 0 To UnitStr.Size()-1
                        lStr = lStr & UnitStr.Val(I) & vbCrLf
                Next
                MsgBox  lStr,,WScript.ScriptName
        End if
End If
```



ReadMeltTemperatureRangeUnits....

```
C
C
```

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeDescription

Purpose

Read the Tcode description for a specified TSet ID, SubID and TCode

Usage

- *Function GetTCodeDescription(ID, SubID, TCode, Value)*

Arguments

ID	Tset ID
SubID	Tset Sub ID
TCode	TCode we are trying to read data from
Value	a String to hold the data

Return Values

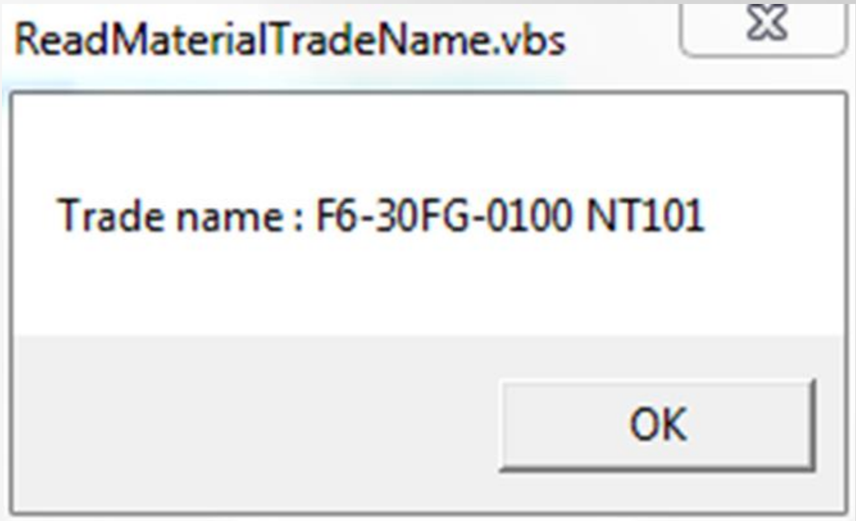
GetTCodeDescription	false = failure, true = success
Value	The Tcode description, returned as a String

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeDescription

Example

```
' Locate the Material Tset and TsetSubID data
Dim MaterialID, MaterialSubID, Material2ID, Material2SubID
Call GetMaterialData(MaterialID, MaterialSubID, Material2ID, Material2SubID)

If MaterialID > 0 Then
        Dim NameStr, ValueStr
        OK = GetTcodeName(MaterialID, 1998, NameStr)
        OK2 = GetTCodeDescription(MaterialID, MaterialSubID, 1998, ValueStr)
        If OK and OK2 Then
                MsgBox  NameStr & " : " & ValueStr,, Wscript.ScriptName
        End if
End if
```



ReadMaterialTradeName.vbs

Trade name : F6-30FG-0100 NT101

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeName

Purpose

Read the TCode name for a specified TSet ID, and TCode

Usage

- *Function GetTCodeName(ID, TCode, Value)*

Arguments

ID	Tset ID
TCode	TCode we are trying to read data from
Value	a String to hold the data

Return Values

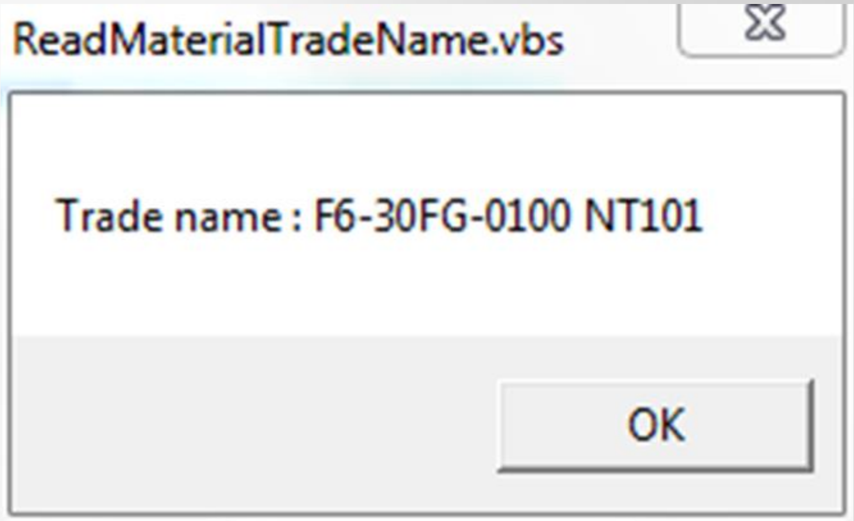
GetTCodeDescription	false = failure, true = success
Value	The TCode name, returned as a String

Eight Functions to manipulate TSet and TCode data

Function name: GetTCodeName

Example

```
' Locate the Material Tset and TsetSubID data
Dim MaterialID, MaterialSubID, Material2ID, Material2SubID
Call GetMaterialData(MaterialID, MaterialSubID, Material2ID, Material2SubID)

If MaterialID > 0 Then
        Dim NameStr, ValueStr
        OK = GetTcodeName(MaterialID, 1998, NameStr)
        OK2 = GetTCodeDescription(MaterialID, MaterialSubID, 1998, ValueStr)
        If OK and OK2 Then
                MsgBox  NameStr & " : " & ValueStr,, Wscript.ScriptName
        End if
End if
```



ReadMaterialTradeName.vbs

Trade name : F6-30FG-0100 NT101

Exercise: Changing Process Settings

Frequently Asked Questions

Frequently Asked Questions

- Can I use the API without showing the GUI
  - No, there is currently no way to do this

- Does the API work on Linux
  - No
  - The tool *studymod* provides capabilities to modify geometry, mesh, property and material data on linux (and windows)
  - The tool *studyrlt* provides capabilities to extract result data on linux (and windows)

Frequently Asked Questions

- Our Most Common API support Request

  “I am trying to change the material data and nothing is happening”

  - Invariably the script will contain a line like

```
Set Prop = PropEd.FindProperty(21000, 1)
```

  - Use the *GetMaterialID* Subroutine to avoid this type of problem

Frequently Asked Questions

- What Level of help can I get from Customer Support
  - We can provide:
    - Basic advice on how to do things
    - Advice on how to approach the task
    - Similar examples
  - Customer Support is not
    - Your personal script developer
    - You script debugging team

Our Advice to getting started with the API

- Review these notes in 1-2 weeks
- Run the Example codes
- Experiment with the Example codes
- Start small
- Talk with Autodesk

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®