



Was ist iLogic

Darstellung der Funktionen
www.CAD-Concept.de

Was ist Inventor-iLogic?

iLogic ist eine Intelligente Design Technologie für Inventor.

Design Intelligenz

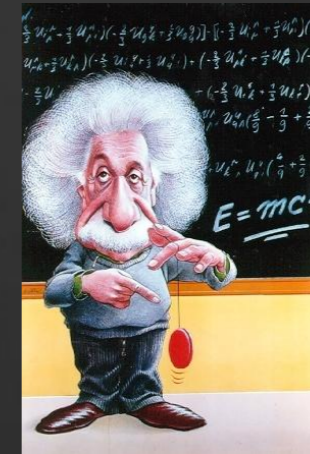
- Parametrisch maßlich getrieben (Konstruktionsabsicht)
- iLogic Regel-getrieben (Konstruktionsintelligenz)

Design Re-use Tool

- iFeatures, iParts, iAssemblies, tabellen-gesteuerte Vorlagen
- Dokument-Vorlagen, Bibliothek und Inhaltscenter Komponenten
- iCopy (adaptives kopieren)
- iLogic (themenbasierte Konfigurationen, "schlaue" Teile und Baugruppen)

Inventor Automation Tool

- VBA Macros (shortcuts)
- API Programmierung (angepasste Anwendungen/add-ins)
- iLogic (Inventor Automation)
- Intent (Business Automation)



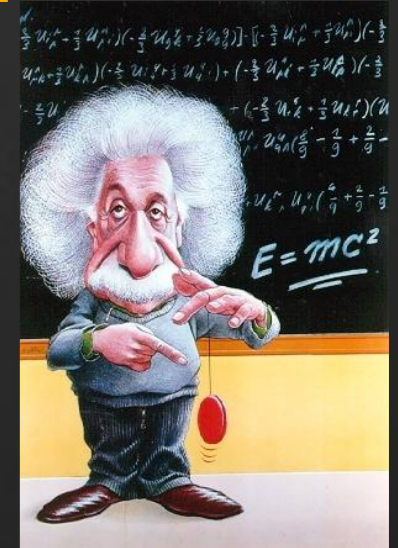
an image from <http://ssabah188.blogspot.com>

Was kann ich mit Inventor-iLogic tun?

Erfassen der “Design Intelligence” – Erstellen eines intelligenten digitalen Prototyps

Inventor/iLogic als Design Intelligence capture Tool

- Unterstützt unterschiedliche Parametertypen (Text und Boolean)
- Definieren einer Werteliste für Parameter / auch dynamisch nach Parameterauswahl
- Aktivieren von automatischer Fehlerkorrektur und Meldungen für Werte-Einschränkungen bei Parametern und gültigen Wertekombinationen
- Unterstützung von regelbasierenden Parameterbeziehungen. Verknüpfungen der Dateien sind nicht nötig.

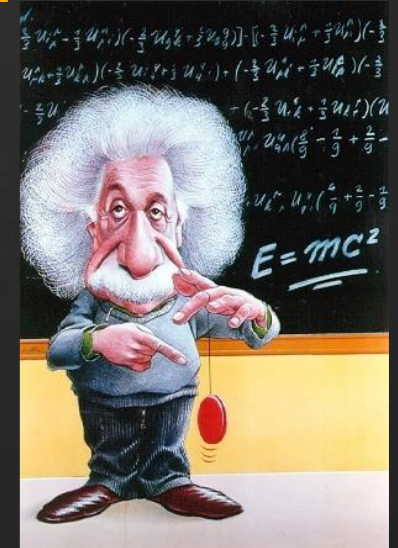


Was kann ich mit Inventor-iLogic tun?

Erfassen der “Design Intelligence” – Erstellen eines intelligenten digitalen Prototyps

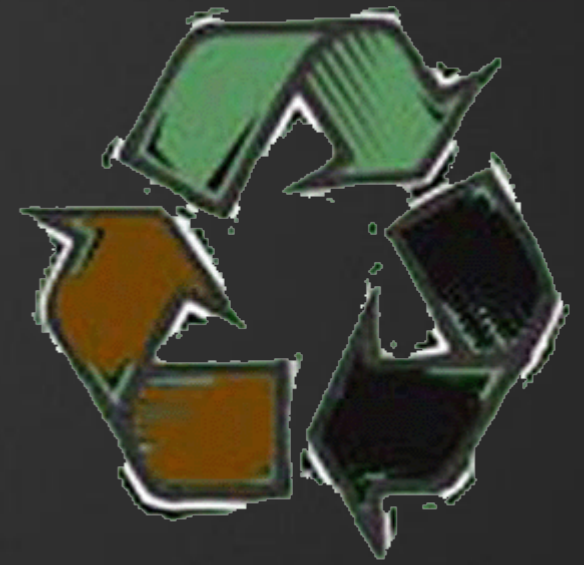
Inventor/iLogic als Design Intelligence capture Tool

- Unterstützung versch. Gleichungstypen (bzgl. Parameter, Attribute, Elemente und Komponenten)
 - Konditionale Bedingungen (if then, else if)
 - Optimierungsläufe (for, while und where loops)
 - Definieren und verwenden von verschiedensten Variablen (nicht nur Parameter)
- Konstruktionsabsicht ist statisch; Konstruktionsintelligenz ist dynamisch
- Intelligenz ist eingebettet (in Regeln) und integriert damit Firmen-Knowhow in das Inventor Dokument.
- Mit iLogic setzt Inventor neue Maßstäbe bzgl. der grundsätzlichen Leistungserwartung beim parametrischen Modelieren.



Was kann ich mit Inventor-iLogic tun?

Effizientes und effektives Wiederverwenden von bestehenden Konstruktionen



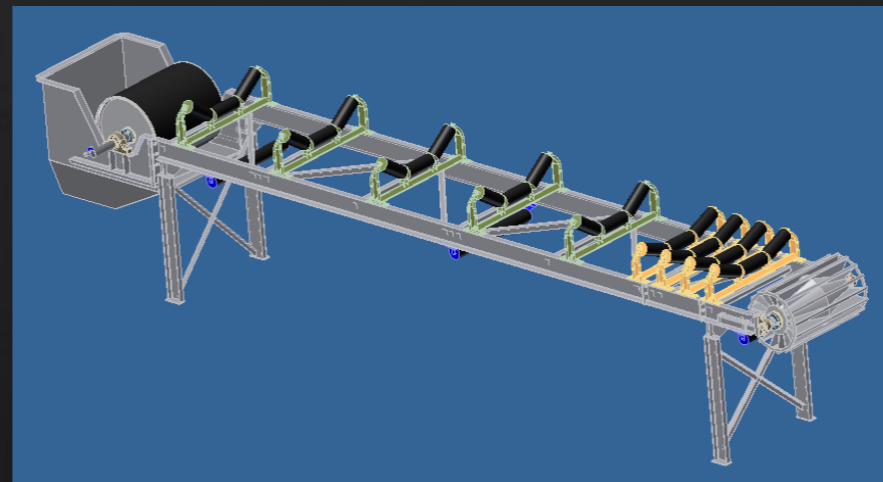
iLogic ist ein Design Re-use Tool

- Erstellen von Vorlagen mit vordefinierten wiederverwendbaren und rekonfigurierbaren Konstruktionselementen.
- Erstellen von themenbasierten Konfigurationen

“In my humble opinion, iLogic actually adds a new paradigm to Inventor's parametric modeling experience. Where iParts and iAssemblies are for variations on a configuration, iLogic is for variations on a theme. Inventor's iParts and iAssemblies provide for value based configurations where iLogic provides for contextual based configuration on my models. When I have a design that is solving a problem, but does so in an infinite number of different ways, then iLogic allows me to add "logic" to that design which will account for each of the infinite number of different approaches to the theme of that design. Straight parametric modeling can't address that.”

Design Re-use Thema:

- > Parametrische Komplexität
- > Mangel an Produktkenntnis
- > Ineffizient und ineffektiv



Brian Hall
Qube-It Design

Was kann ich mit Inventor-iLogic tun?

Automatisiere Inventor

Mit Regeln kann man: (z.B.)

- Prüfungen durchführen (iProperties, Maße, etc)
- Optimierungsberechnungen durchführen
- Mehrere Arbeitsgänge basierend auf einer einzigen Benutzereingabe durchführen
- Automatisch ungültige Parameter und andere Benutzereingaben korrigieren
- Lesen/Schreiben in Tabellen und Datenbanken



Was kann ich mit Inventor-iLogic tun?

Automatisiere Inventor

Mit Regeln kann man: (z.B.)

- iFeature, iPart und iAssembly Konfigurationen steuern
- Produktkonfiguratoren erstellen
- Parameter und Regeln mittels kundenspezifischer VB Formulare steuern
- Ausführen von Makros (API, VBA/VB.net) nach Dokumenten Ereignissen
 - (Datei neu, öffnen, speichern, schließen, Geometrieänderung, etc.)
- Einsatz und Zugriff auf externe Applikationen (Aufruf von deren API)
- Nutzen des gesamten Leistungsspektrum der Inventor API und VB.net

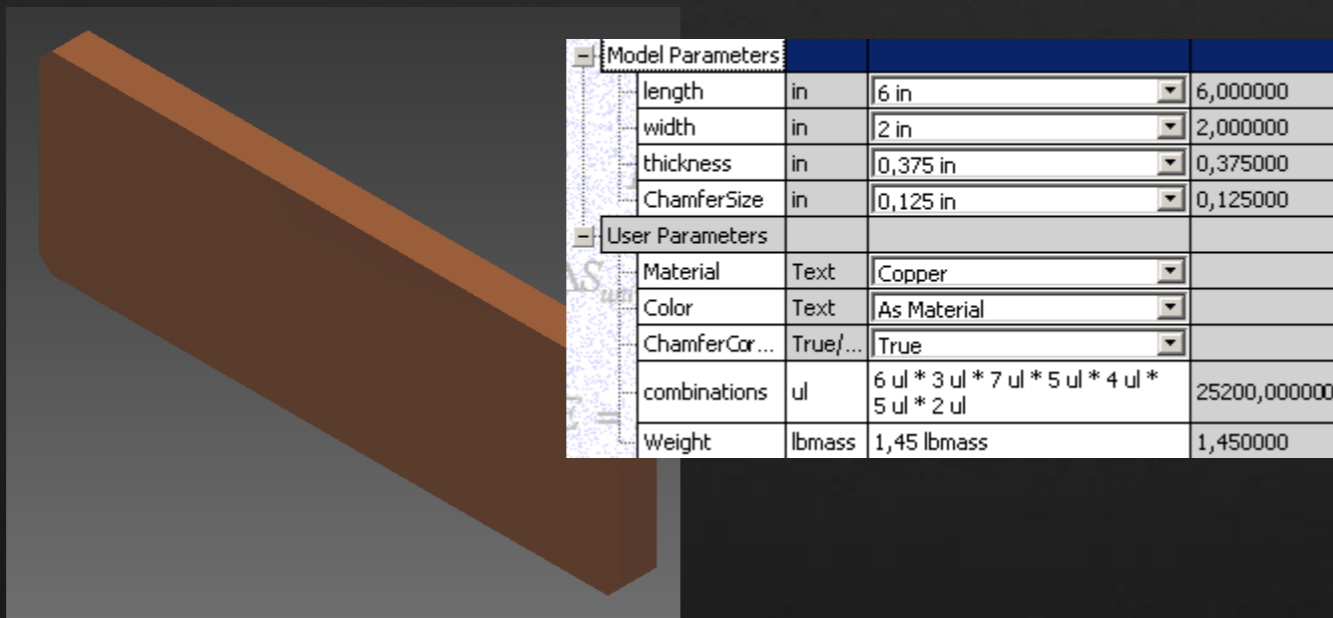


Wer kann iLogic brauchen?

iLogic erweitert die Leistungsfähigkeit und Möglichkeiten von Inventor in einer Art und Weise, dass jeder Inventor Anwender davon profitiert.

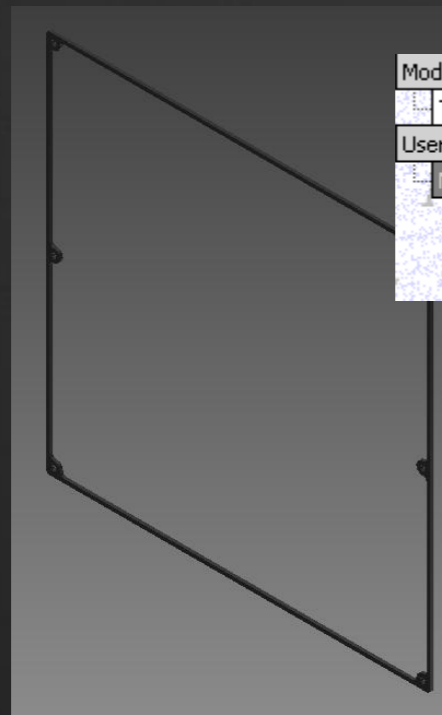
Allgemeine einfache Anwendungsfälle: (Kundenbeispiele)

1. Einschränken von Parametergrößen auf eine diskrete Liste von gültigen Werten



Allgemeine einfache Anwendungsfälle: (Kundenbeispiele)

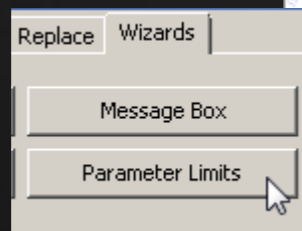
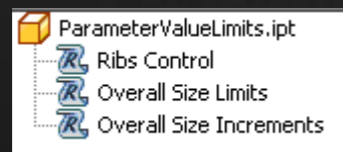
2. Erzwingen von gültigen Kombinationen nach Auswahllisten



Model Parameters			
Thickness	in	0,07 in	0,070000
User Parameters			
Material	Text	Silicone	

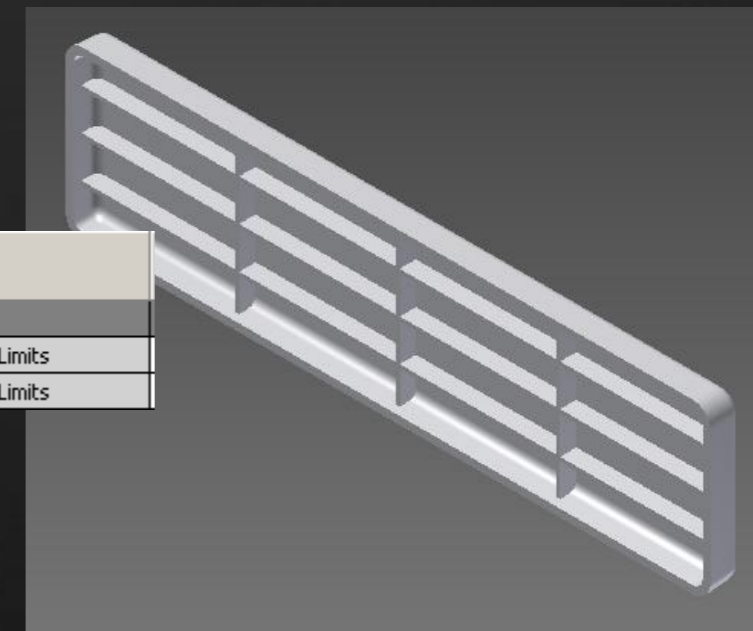
```
MultiValue.SetValueOptions(True, DefaultIndex := 0)  
  
iProperties.Material = Material  
  
If Material = "Rubber" Then  
    MultiValue.SetList("Thickness", .03, .06, .08, .10, .12)  
ElseIf Material = "Silicone" Then  
    MultiValue.SetList("Thickness", .02, .05, .07, .90, .11)  
End If
```

3. Setzen von min/max Grenzen für Parameter inkl. Autokorrektur und Fehlermeldung



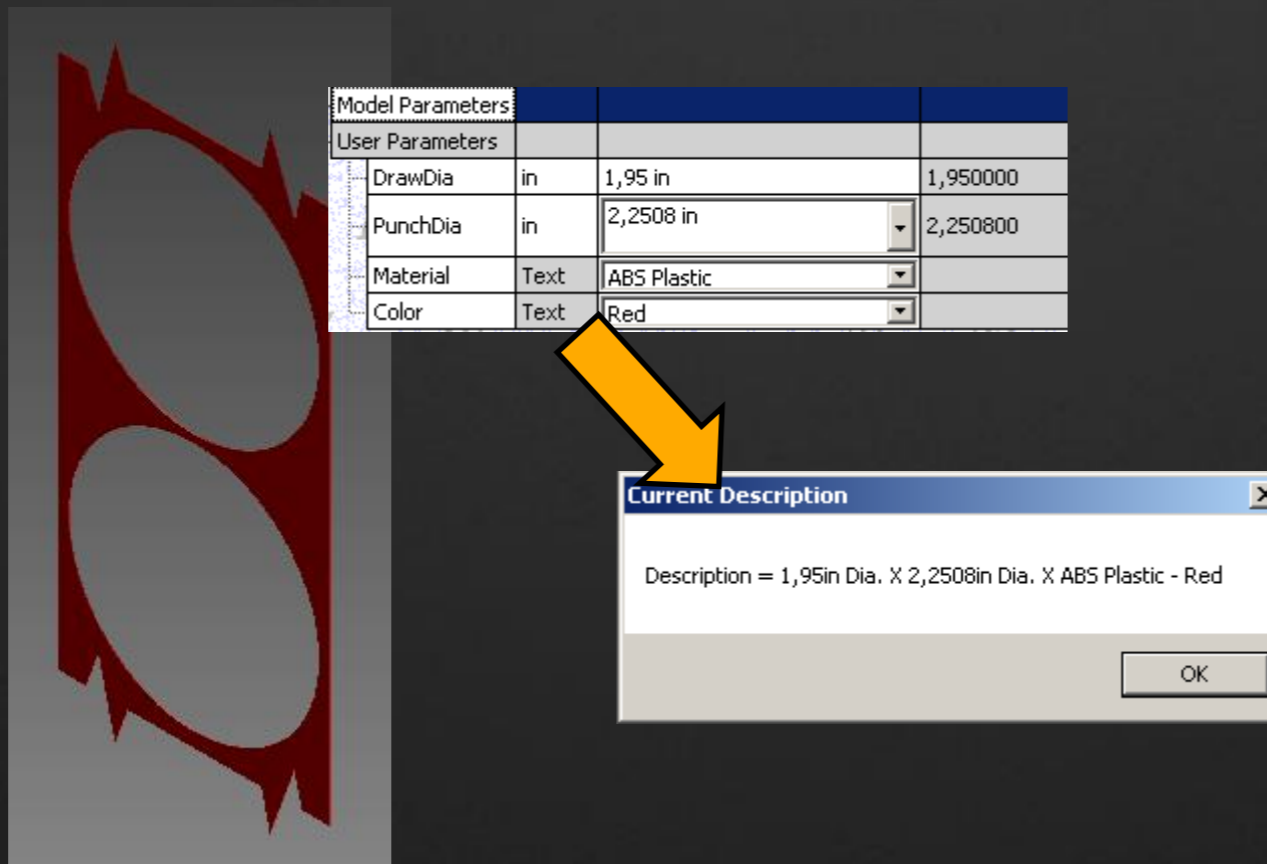
Parameter Name	Unit/Ty	Equation	Nominal Value	Driving Rule
Model Parameters				
Length	in	8,25 in	8,250000	Overall Size Limits
Width	in	2,125 in	2,125000	Overall Size Limits

$4 < \text{Length} < 10$
 $2 < \text{Width} < 3$



Allgemeine einfache Anwendungsfälle: (Kundenbeispiele)

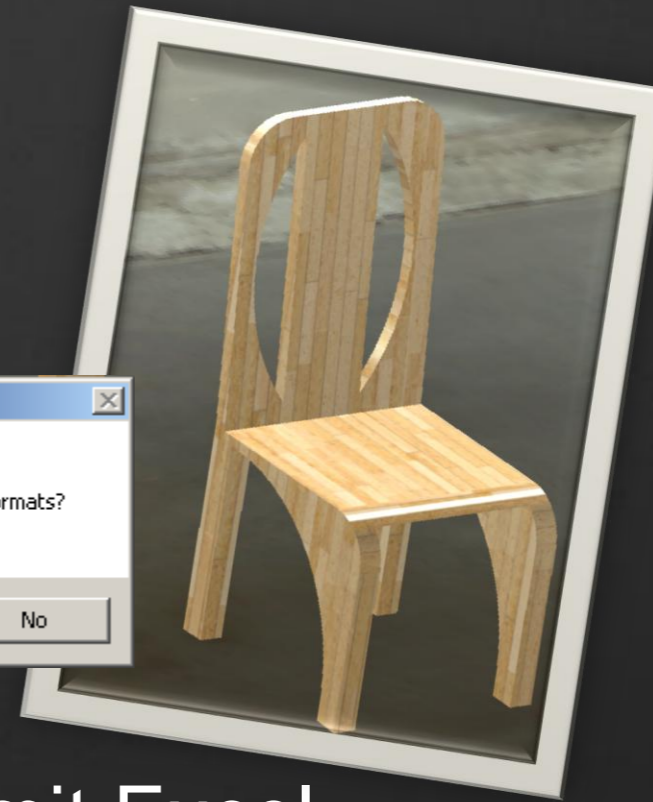
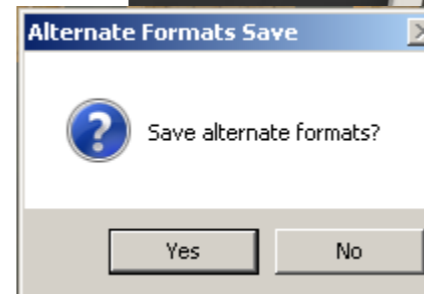
4. iProperty Wertprüfung und Steuern von iProperty Werten mittels Parametern



Allgemeine einfache Anwendungsfälle: (Kundenbeispiele)

5. Automatisches Speichern von unterschiedlichen Formaten bei jedem Datei speichern

```
FileNameAndPath = ThisDoc.Path & "\SavedFormats\" & ThisDoc.FileName  
  
i = MessageBox.Show("Save alternate formats?", "Alternate Formats Save")  
  
If i = vbYes Then  
  
ThisDoc.Document.SaveAs(FileNameAndPath & ".jpg" , True)  
ThisDoc.Document.SaveAs(FileNameAndPath & ".bmp" , True)  
ThisDoc.Document.SaveAs(FileNameAndPath & ".sat" , True)  
ThisDoc.Document.SaveAs(FileNameAndPath & ".stp" , True)  
ThisDoc.Document.SaveAs(FileNameAndPath & ".dwf" , True)  
  
End If
```

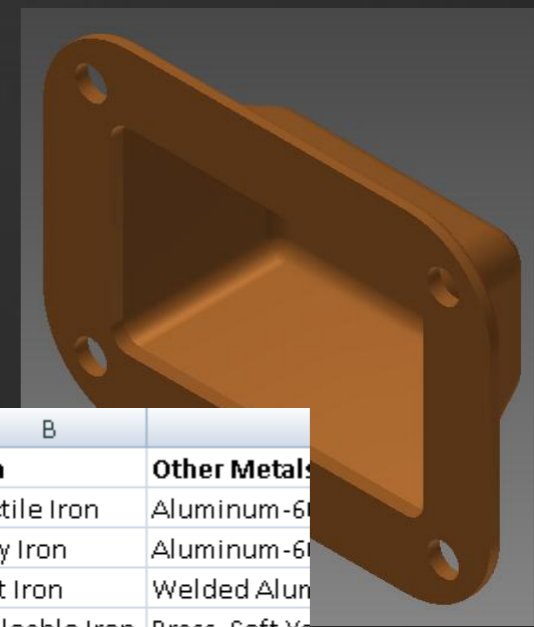


6. Erweiterte/Verbesserte Interaktion mit Excel

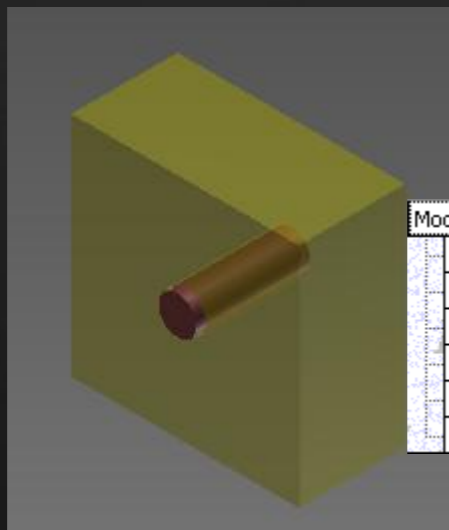
- Lesen/Schreiben von Parametern und iProperties in/von Tabellenkalkulation
- Keine Parametertyp-, Blatt- oder Format Einschränkungen
- Verwenden von eingebundenen oder nicht-eingebundenen, eingebettet oder externe Tabellen

iLogic Anbindung mit Excel

6. Erweiterte/Verbesserte Interaktion mit Excelsonst



	A	B	
1	Steel	Iron	Other Metals
2	Steel	Ductile Iron	Aluminum-60
3	Steel, Mild	Gray Iron	Aluminum-60
4	Alloy Steel	Cast Iron	Welded Alun
5	Carbon Steel	Malleable Iron	Brass, Soft Ye
6	Cast Steel		Welded Bras
7	Galvanized Steel		Bronze, Soft

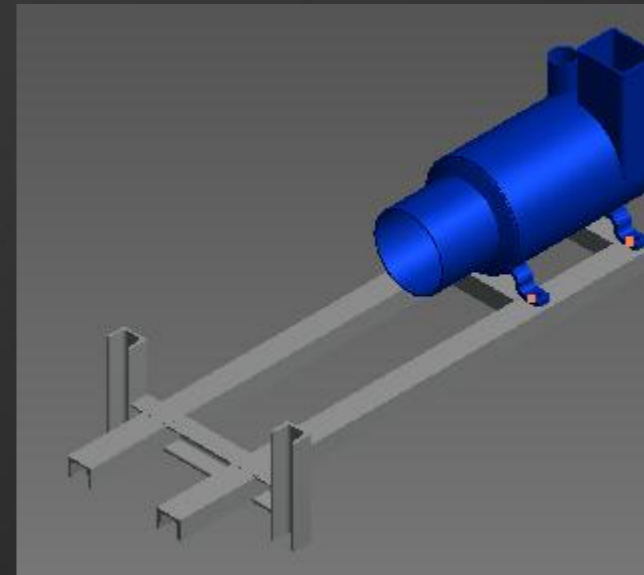


Model Parameters				
length	in	1,5 in	1,500000	
width	in	1,5 in	1,500000	
thickness	in	0,70 in	0,700000	
Hole_Dia	in	0,27 in	0,270000	
Pin_Dia	in	0,25 in	0,250000	Find the Right Pin Size from Excel
Pin_Length	in	0,75 in	0,750000	Find the Right Pin Size from Excel

iLogic Anbindung mit Excel

6. Erweiterte/Verbesserte Interaktion mit Excelsonst

```
GoExcel.Open("Excel UI.xlsx", "Sheet1")
Diameter_WroughCasing = GoExcel.CellValue("B1")
Height_WroughCasing = GoExcel.CellValue("B2")
Diameter_BottomOutlet = GoExcel.CellValue("B4")
Heigth_BottomOutlet = GoExcel.CellValue("B5")
Thickness_Wall = GoExcel.CellValue("B7")
Angle_of_BottomCone = GoExcel.CellValue("B9")
Angle_TopCone = GoExcel.CellValue("B10")
Width_SquareInlet = GoExcel.CellValue("B12")
```



	A	B
1	Diameter_WroughCasing	300
2	Height_WroughCasing	200
3		
4	Diameter_BottomOutlet	150
5	Heigth_BottomOutlet	120
6		
7	Thickness_Wall	4
8		
9	Angle_of_BottomCone	120
10	Angle_TopCone	10
11		
12	Width_SquareInlet	100
13	Length_SquareInlet	100
14	Heigth_SquareInlet	80
15		
16	Change_Top	Cone
17	On_Off_RoundInlet	On
18		
19	Surface_Treatment	Blue (Flat)
20	Physical_Material	Steel
21		
22	Angle_Round_Inlet	0
23		
24	Heigth_Frame	1000
25	Change_Profiles	U-Profiles

Information Welle

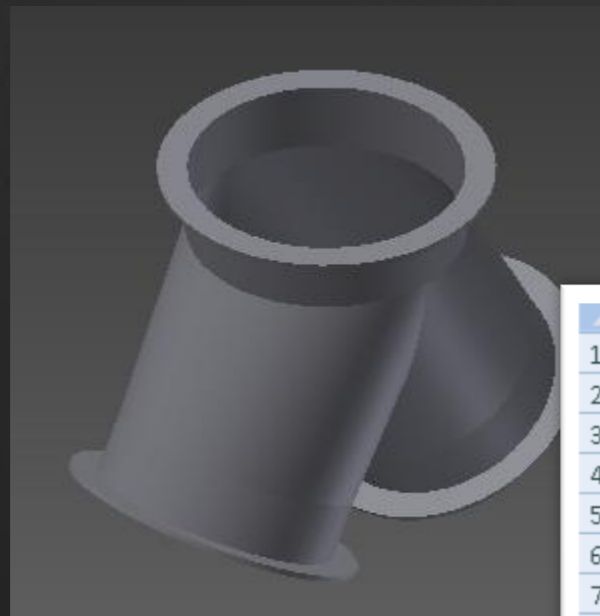
Teilenummer ist SH-0023
 Hersteller ist JIAOZUO MENGXIN SPECIAL STEEL CO., LTD

OK

	A	B	C	D	E	F
1	CrusherWidth	CrusherHeight		ShaftID	ShaftPartNumber	ShaftMFG
2	48	52		4852	001	Ningbo Yinzhou Qiuqing Machine Co., Ltd.
3	52	54		4854	002	Shanghai Teng-hui Forging Co., Ltd
4	56	56		4856	003	Ningbo Yinzhou Qiuqing Machine Co., Ltd.
5	60	58		4858	004	JIAOZUO MENGXIN SPECIAL STEEL CO., LTD
6	64	60		4860	005	Shanghai Teng-hui Forging Co., Ltd
7	65	62		4862	006	NINGBO YONGLONG MACHINERY CO., LTD
8				5252		JIAOZUO MENGXIN SPECIAL STEEL CO., LTD

iLogic Anbindung mit Excel

7. Erweiterte Excel-Anbindung mit Auswahllisten

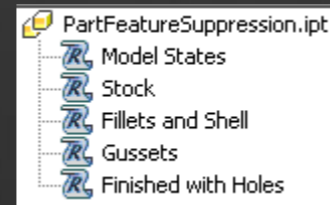


	A	B	C	D	E	F	G	H	I
1	Alle Varianten der Hosen	NWA Liste		NWA	NWB	NWA real	NWB real	Blechstärke	Innendurchm.
2	Blechstärke kann bei	140		140	125	140	127	2	140
3	Sonderausführungen	160		160	125	160	127	2	160
4		180		160	140	160	140	2	160
5		200		180	125	180	127	2	180
6		224		180	140	180	140	2	180
7		250		180	160	180	160	3	180
8		280		200	140	201	140	2	200
9		315		200	160	201	160	3	200

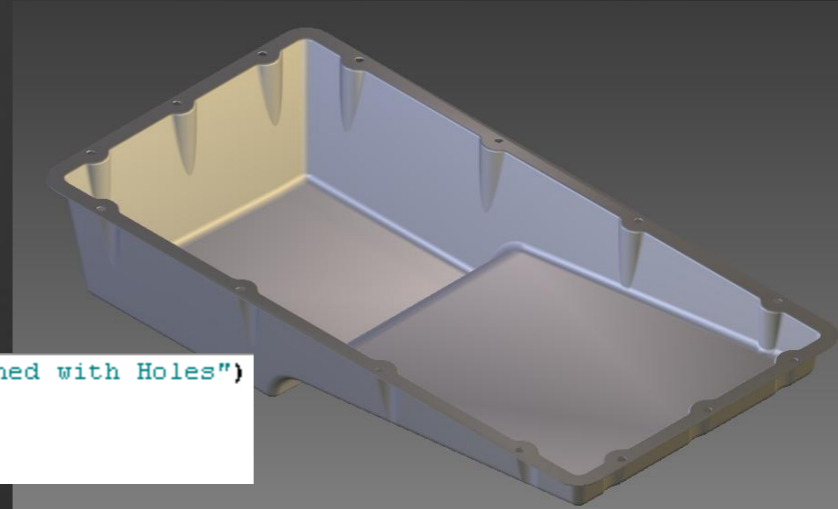
Spezielle Auswahlliste von Ausgangsdurchmesser (NWB) ausgehend vom Eingangsdurchmesser (NWA)

iLogic ist für alle Inventor Anwender

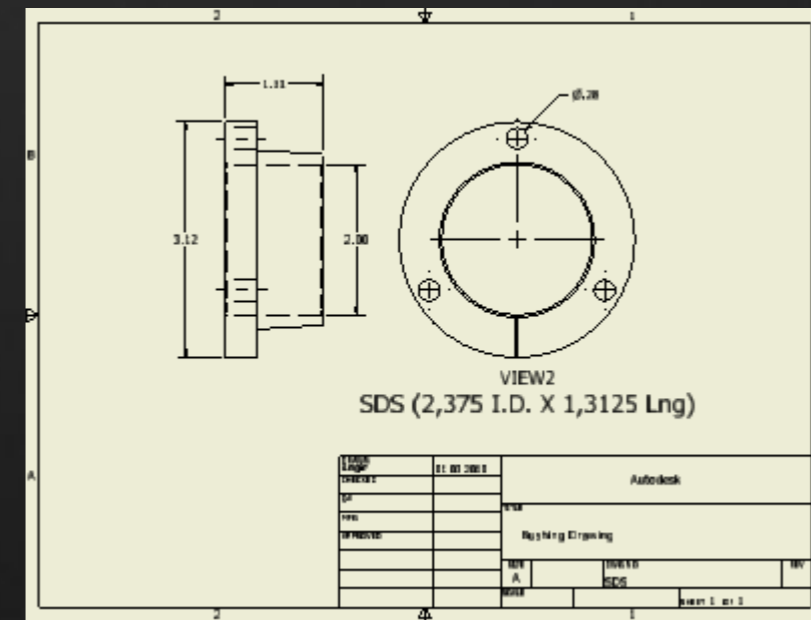
8. Unterdrücken von Elementen im Bauteil



```
MultiValue.SetList("ModelState", "Stock", "Fillets and Shell", "Gussets", "Finished with Holes")  
iLogicVb.RunRule(ModelState)
```



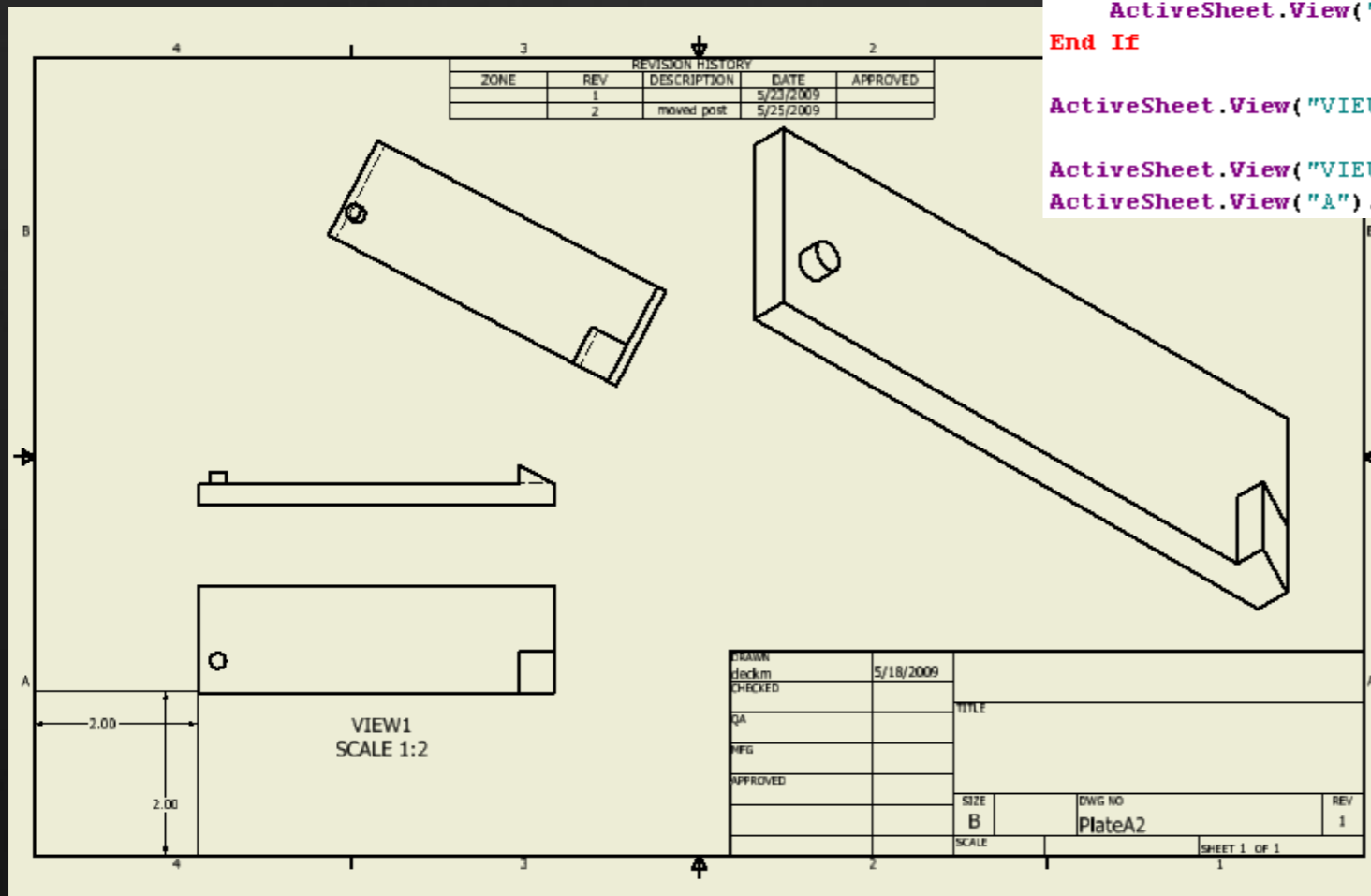
9. "Schlaue" Zeichnungsvorlagen



iLogic in Zeichnungen

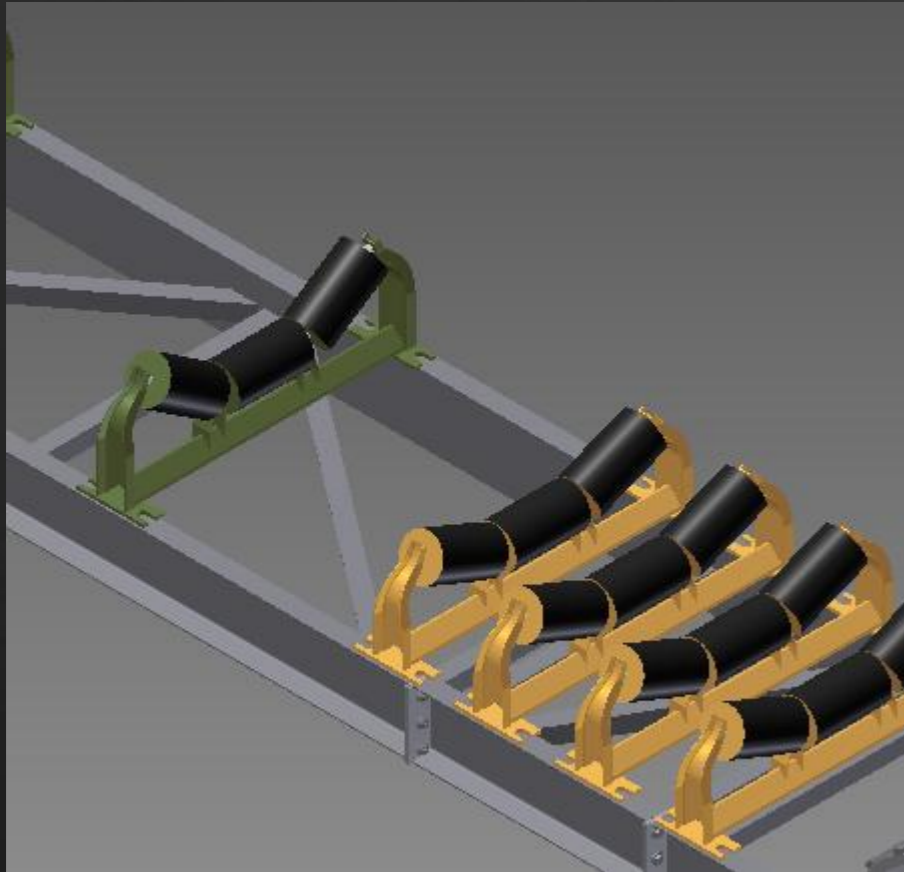
9. "Schlaue" Zeichnungsvorlagen

```
modelSize = ActiveSheet.View("VIEW1").Width / ActiveSheet.View("VIEW1").Scale  
Trace.WriteLine(" modelSize = " & modelSize)  
  
If (modelSize > 10.0) Then  
    ActiveSheet.View("VIEW1").ScaleString = "1:4"  
    ActiveSheet.View("Iso View1").ScaleString = "1:2"  
Else  
    ActiveSheet.View("VIEW1").ScaleString = "1:2"  
    ActiveSheet.View("Iso View1").ScaleString = "1:1"  
End If  
  
ActiveSheet.View("VIEW1").SetSpacingToCorner(2, 2, SheetCorner.BottomLeft)  
  
ActiveSheet.View("VIEW3").SpacingBetween("VIEW1") = 1.0  
ActiveSheet.View("A").SpacingBetween("VIEW3") = 1.0
```



iLogic und iParts

9. iPart-Steuerung und mehr



	Member	Part Number	BeltWidth	SlotLength	Slot
1	C35-24	C35-24	24	33	6
2	C35-30	C35-30	30	39	6
3	C35-36	C35-36	36	45	6
4	C35-42	C35-42	42	51	7
5	C35-48	C35-48	48	57	7
6	C35-54	C35-54	54	63	7

```
iPart.FindRow("TailIdler:1", "BeltWidth", "=", BeltWidth)  
iPart.FindRow("HeadIdler:1", "BeltWidth", "=", BeltWidth)
```

Bauteilübergreifende Parameter:

```
'*****Head Pulley*****'  
Parameter("DischargeChute:1", "Belt_Width") = BeltWidth  
Parameter("DischargeChute:1", "Pulley_Diameter") = HeadPulleyDiameter  
Parameter("DischargeChute:1", "Drive_Side") = HeadDriveSide  
Parameter("DischargeChute:1", "Hub_Type") = HeadHubType  
Parameter("DischargeChute:1", "Bearing_Bore") = HeadBearingSize  
  
'*****Tail Pulley*****'  
Parameter("TailSection:1", "Tail_Pulley_Diameter") = TailPulleyDiameter  
Parameter("TailSection:1", "Belt_Width") = BeltWidth  
Parameter("TailSection:1", "Hub_Type") = TailHubType  
Parameter("TailSection:1", "TailBearingBore") = TailBearingSize  
  
'*****Support Height*****'  
Parameter("HeadSupport:1", "Support_Height") = HeadSupportHeight  
Parameter("TailSupport:1", "Support_Height") = TailSupportHeight  
  
'*****Section Frame, Idler, Support Width*****'  
Parameter("HeadFrame:1", "Belt_Width") = BeltWidth  
Parameter("TailFrame:1", "Belt_Width") = BeltWidth
```

iLogic als Konfigurator

9. Konfigurator



Playset Configuration

How Many Bases?

Single Base

Double Base

Single Base Configuration

Climbing Bars

Rock Wall

Double Base Configuration

Rock Wall

Rope Climb

Rope Climb

Standard

Left Railing Configuration

No Slide

Left Side Slide

Right Side Slide

Right Railing Configuration

No Slide

Left Side Slide

Right Side Slide

Swing Set Configuration

2 Station

3 Station

Bridge Configuration

Bridge

Tunnel

