

IM471182

Vom FDU Dummy Asset zum Inventor Fertigungsteil per iLogic und .net

Pascale Brassat (Dipl.-Ing.(FH))
Claudius Peters Projects GmbH

Lernziele

- Inventor Factory Design Utilities (FDU) kennenlernen
- Inventor iLogic kennenlernen
- Möglichkeiten von vb.net und Inventor/Vault kennenlernen
- Verständnis dafür bekommen, wieviel Möglichkeiten die Kombination der Tools bietet, um eigene Ideen zu verfolgen

Beschreibung

Bei den meisten Firmen gibt es Bereiche wo sich Änderungen in der 3D Planung immer bis kurz vor Projektübergabe ergeben. Gerade dort gilt es Wege zu finden, mit denen es gelingt schnell zum Ziel zu kommen. Anhand eines Beispiels soll dargestellt werden, wie mit Hilfe der FDU vereinfachte Assets kombiniert werden können. Diese werden dann aufgrund der FDU Vererbung und Parametrik angepasst. Anschließend werden die Parametereinstellungen auf ein Fertigungsmodell übertragen, so dass am Ende aus einem simplen Planungsasset ein vollwertiges identisches Fertigungsmodell geworden ist. Die Demo soll zeigen, welche vielfältigen Möglichkeiten geboten werden, neue Lösungsansätze zu verfolgen.

Referent(en)

In 2006 habe ich mein Studium in der Studienrichtung Maschinenbauinformatik an der Hochschule für angewandte Wissenschaften in Hamburg abgeschlossen. Danach arbeitete ich in der Komponenten Konstruktionsabteilung von Claudio Peters Technologies GmbH und wechselte 2011 in einen Bereich der sich mit allen Themen rund um CAx beschäftigt. Dieser Bereich wurde 2013/2014 in die IT Abteilung von Claudio Peters Projects GmbH eingegliedert. In dieser Zeit entstanden mehrere eigenständige Applikationen, um die täglichen Arbeitsabläufe der CAx Nutzer auf Basis der Autodesk Produkte zu vereinfachen und zu optimieren. Neben dieser hauptsächlich administrativen Tätigkeit bin ich auch immer wieder bei konstruktiven Themen behilflich. Zum Beispiel im Bereich Generative Design (mit dem Thema wurde CPP bei vielen Autodesk Anwendern bekannt) oder Optimierung von Bauteilen mit Blick auf die effektive Nutzung von Parametrik. Seit 2006 nutze ich hauptsächlich Inventor und alle darin zur Verfügung stehenden Techniken. Besonderes Augenmerk liegt seit ca. 2 Jahren auf iLogic in Kombination mit der FDU und Vault Pro.

Grundlegende Informationen für die weiteren Themenbereiche

In diesem Teil möchte ich kurz die Grundlagen erklären, die wichtig sind die nachfolgenden Themenbereiche zu verfolgen. Z.B. die verwendeten Software Produkte.

Autodesk Inventor (Inventor)

Inventor ist die Basis aller folgenden Themen. Da Inventor wie eine Art 3D CAD Betriebssystem genutzt werden kann (3D Engine und grundlegende Funktionen sind alle vorhanden) ist Inventor perfekt geeignet, um eigene Ideen umzusetzen. Inventor bietet eine breite Palette von Anpassungsmöglichkeiten an. Angefangen bei CAD Parametrik, die für alle Anwender geeignet ist, bis hin zur Programmierung via .net API für die fortgeschrittenen Entwickler.

iLogic

iLogic ist eine in Inventor integrierte Programmierumgebung, welche sehr dem altbekannten VBA oder .net ähnelt. Der große Vorteil bei iLogic ist, dass sehr schnell auf alle im Modell/Dokument verfügbaren Ressourcen zugegriffen werden kann. Durch die große Nähe zu Visual Basic ist die Sprache besonders leicht für Umsteiger von VBA zu erlernen.

Autodesk Factory Design Utilities (FDU)

Die FDU ist Bestandteil der Product Design and Manufacturing Collection (PDMC) und steht damit allen deren Anwendern zur Verfügung.

Die FDU ist perfekt geeignet, um Bauteile miteinander zu kombinieren. Im Wesentlichen besteht die FDU aus der Möglichkeit eine eigene Bibliothek an sogenannten Assets zu erstellen, diese miteinander über Connectoren zu verbinden und dank der Connectoren Klassen auch Parameterwerte von einem Teil auf ein anderes zu vererben.

Autodesk Vault Professional (Vault Pro)

Vault Pro ist das Standard PDM für die Verwendung mit Inventor. Es dient zur Ablage und Verwaltung der Dokumente. Darüber hinaus können in Vault z.B. auch Stücklistenstrukturen erstellt werden.

Vault Pro API

Die .net Schnittstelle zu Vault ist das Tor zum Vault für fortgeschrittene Entwickler in Kombination mit einer entsprechenden .net Entwicklungsumgebung (z.B. Microsoft Visual Studio) und Sprache (z.B. C# oder VB)

Visual Basic .net (VB.net)

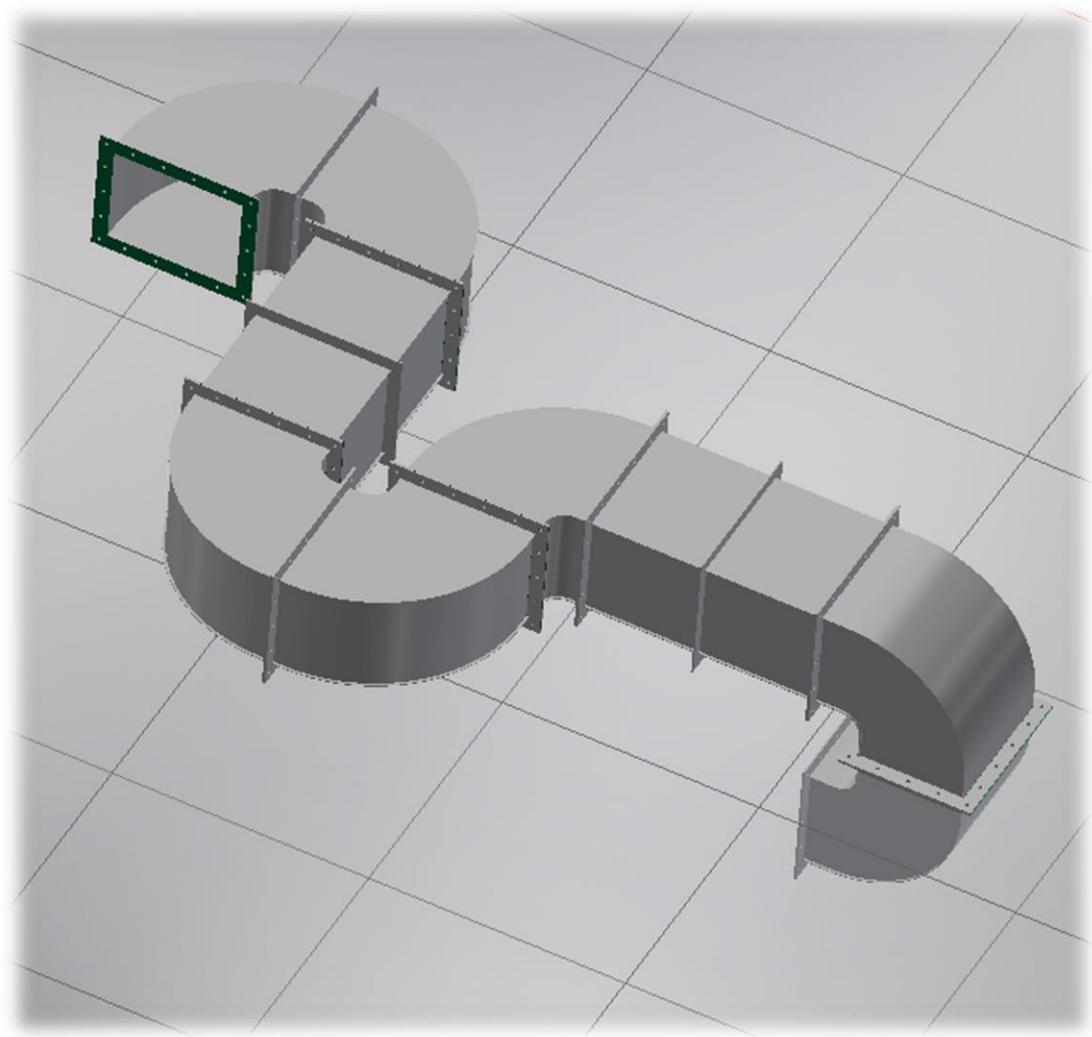
VB.net verwende ich aufgrund der großen Nähe zu iLogic und der Mehrzahl an Beispielen und Programmierressourcen die sich mit Inventor beschäftigen. Zudem fällt der Umstieg von VBA für viele Anwender leichter, daher hat sich diese Sprache bei uns intern sehr bewährt. Es kann auch jede andere .net fähige Sprache genutzt werden. Da sich der Bereich .net eher an die fortgeschrittenen Anwender richtet und eine eigenständige Entwicklungsumgebung erfordert (z.B. Visual Studio von Microsoft) ist der Anteil von .net in diesem Dokument eher gering, soll aber deutlich machen was alles mit Hilfe von .net machbar ist und Lust auf „Mehr“ machen.

Die meisten Bereiche von .net z.B. in Kombination mit der Inventor API werden auch von iLogic abgedeckt.

Beispiel

Das Dokument soll anhand eines hoffentlich interessanten Beispiels zeigen, welche Möglichkeiten sich bieten, wenn mehrere Techniken kombiniert werden. Generell gilt es in der heutigen Zeit immer schneller zu werden. Die Zeit zwischen letzter Änderung aufgrund eines Kundenwunsches und Auslieferung wird bei uns immer geringer. Das heißt die Zeit in der wir reagieren können und müssen wird ebenfalls immer kürzer. Da wir im Anlagengeschäft tätig sind, handelt es sich bei den Änderungen meistens um Änderungen die nicht eine einzelne Komponente von uns betreffen, sondern mehrere.

Dieses Beispiel soll sich um die Planung von Belüftungskanälen drehen.



Das sind im Grunde erst einmal nur viereckige oder Runde Blechteile die miteinander verbunden werden.



Da bei unseren Kunden keine normalen klimatechnischen Bauteile genutzt werden können, sondern robustere Technik zum Einsatz kommen muss, werden diese Kanäle immer für den jeweiligen Fall angepasst modelliert und später gefertigt.

Wie lief das bisher ab?

1. Der Leitungsverlauf wurde im Anlagenmodell grob skizziert
2. Dann wurde auf Basis dieser Skizze ein Sweeping erzeugt
3. Dieses Sweeping wurde dann in mögliche Komponenten zerlegt
4. Diese Komponenten wurden dann als Baugruppe modelliert (teils auf Basis existierender Teile, teils neu)
5. Aus den Teilen wurden dann Zeichnungen erstellt

Wurde dann eine Änderung am Verlauf vorgenommen, dann gab es einen iterativen Prozess durch die Punkte 2-5. Entweder für einen Teil der Leitung oder komplett.

Warum? Die Zeit am Ende einer Auftragslaufzeit war einfach zu kurz um die Punkte 3-5 ans Ende zu schieben und zu warten bis alle Änderungen eingepflegt waren.

Aus dem Grund gab es die Idee diesen Prozess 1-5 zu optimieren.

Und das wollen wir auf den folgenden Seiten einmal durchspielen. Damit sollte es möglich sein, das Gelesene auf eigene Themen anzuwenden.

Nur um es vorweg zu nehmen. Eine vorsichtige interne Untersuchung ergab, dass wir wenigstens 50% der Zeit einsparen, die für die Planung derartiger Teile bisher notwendig war.

Grundlegende Überlegungen für alle weiteren Schritte

Dieser Teil beschäftigt sich hauptsächlich damit, wieso welche Technologie ausgewählt wurde. In den darauffolgenden Teilen geht es dann um die eigentliche Umsetzung.

Optimierung ja, aber wo?

Zunächst einmal ist das Aufspüren von Baugruppen/Prozesse mit Potential für Optimierungen eine Kunst für sich und erfordert die Mithilfe der entsprechenden Abteilungen. Ohne dass jemand sagt: „Das dauert hier aber echt lange, da muss es eine simplere Lösung geben“, oder einer entsprechenden Prozessanalyse wird es schwer, dort etwas voran zu treiben.

In unserem Fall wurden die Belüftungskanäle als so eine kritische Stelle ausfindig gemacht.

Der erste Schritt war also zu überlegen, wie so etwas vereinfacht werden kann. Da die meiste Zeit nicht so sehr beim Skizzieren, oder dem Sweeping benötigt wurde, sondern eher in den Phasen der Erstellung von Fertigungsunterlagen, war klar, wo angesetzt werden musste.

Die Fertigungsmodelle und Unterlagen mussten schneller erstellt werden.

Varianten Identifizierung

Als Erstes wurde also betrachtet, ob die Anzahl der unterschiedlichen Modelle reduziert werden konnte.

Dazu wurde viel der bisherigen Leitungen betrachtet und überlegt, welche grundlegenden Teile notwendig wären, um diese Leitungen zu erstellen (unabhängig von der Größe)

Diese Teile wurden identifiziert und es gab am Ende 9 Standardformen, mit denen sich nahezu alle möglichen Leitungsverläufe planen lassen würden.

Schon ein bisschen vorgegriffen: Die FDU selbst hat einige Besonderheiten, die es in Bezug auf die Connectoren zu beachten gilt. So ist es beispielsweise notwendig Bögen sowohl horizontal als auch vertikal als Element zu haben. Das sind Dinge, die im Laufe der Zeit beim Einsatz der FDU klar werden und dann leicht zu kontrollieren sind. Also auch wenn ein Bogen benötigt wird, sind es unter Umständen 2 in jeweils einer Richtung.

Welche Technologien verwenden?

Der nächste Schritt war zu schauen, welche Technologien benutzt werden sollten. Es macht beispielsweise wenig Sinn bei einem Bauteil das lediglich aus 2 Varianten besteht an eine iLogic oder .net Parametrik zu denken. Meistens ist man dort schneller die Varianten eben schnell von Hand zu erzeugen und alles ist gut.

In unserem Fall hatten wir, wie geschrieben 9 Bauformen identifiziert mit einer Vielzahl von variablen Größen (Breite, Höhe, Länge, Winkel, Radius etc.).

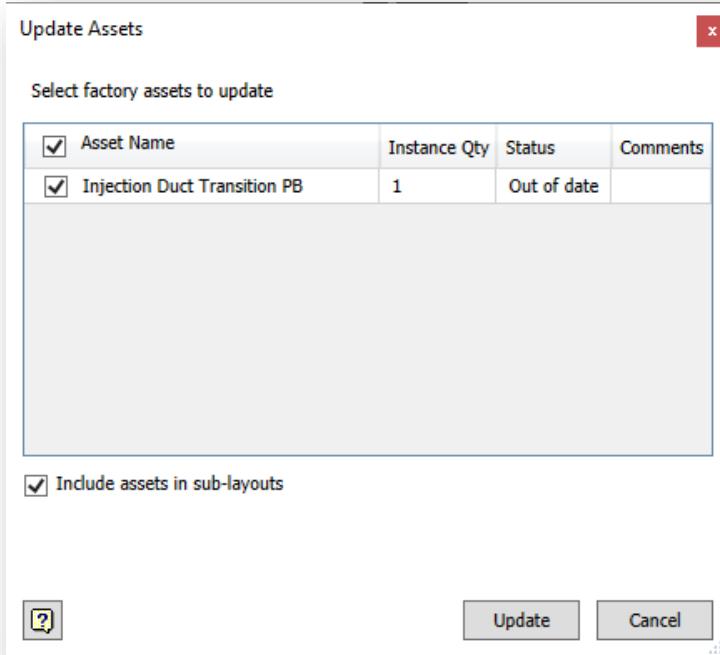
Also 9^n Varianten. Das bedeutet ganz klar: Parametrik. Je nach Variantenzahl dann meistens iLogic.

In iLogic ist es möglich mit sehr wenigen Programmzeilen tausende von Varianten zu erzeugen, was z.B. mit Hilfe von iParts oder iAssemblies schnell sehr unübersichtlich werden kann.

Im Fall unserer Belüftungskanäle handelt es sich um Bauteile die miteinander verbunden werden und deren Größe sich an das jeweils angeschlossene Teil angleichen sollte. Ein Kanalelement entspricht in Breite und Höhe dem Nachfolgenden. Diese Vererbung ist bereits in der FDU integriert und muss nur mit einer entsprechenden Connector Class Datei genutzt werden

Darüber hinaus bietet die FDU noch weitere Vorteile. Zum einen kümmert sich die FDU automatisch um die Vergabe von Dateinamen anhand der Änderungen an den Parametern (Das heißt ein eingefügtes FDU Asset hat denselben Namen, wenn es sich um dieselbe Ausprägung handelt). Die Zeit zum Erstellen von Kopien, bevor ein Teil verändert wird, entfällt.

Außerdem lassen sich während der Entwicklung leicht Änderungen durchführen. So ist es möglich eine Änderung an einem Bibliotheks-Asset per „Update Assets“ an alle bereits eingefügten Assets zu übernehmen. (z.B. wenn eine Variante hinzugefügt wurde)



Wichtig bei allen Überlegungen ist: Der Anwender soll sich nach Möglichkeit auf seine eigentliche Aufgabe konzentrieren können. In diesem Fall die Planung der Kanäle.

Gesetzt waren also bisher 2 Techniken: iLogic und die FDU.

Diese beiden Technologien laufen Hand in Hand und lassen sich wunderbar kombinieren.

Damit war klar, in Zukunft würden die Punkte 1-3 im Prozess anders ablaufen.

Die folgenden Punkte:

1. Der Leitungsverlauf wurde im Anlagenmodell grob skizziert
2. Dann wurde auf Basis dieser Skizze ein Sweeping erzeugt
3. Dieses Sweeping wurde dann in mögliche Komponenten zerlegt

werden zu:

1. Kanal planen mit Hilfe der 9 FDU Assets

Der nächste Schritt war neu und es stellte sich die Frage, wie das am elegantesten gelöst werden könnte.

Die FDU sollte nicht genutzt werden um Modelle im Fertigungsdetailgrad als Asset abzulegen. Varianten lassen sich an vereinfachten Assets deutlich leichter ausprobieren. Zudem ist die Performance bei sehr großen Anlagen deutlich besser.

Die Idee war, vereinfachte Assets zu haben und daneben die entsprechenden Fertigungsmodelle.

Doch wie lassen sich die Teile miteinander kombinieren, ohne dass der Anwender alle Parameter händisch übertragen muss?

Hier kam .net ins Spiel. Es werden zunächst beide Modelle erstellt, Asset und Fertigungsmodell. Das Asset als vereinfachtes iLogic gesteuertes Parametrikmodell und das Fertigungsmodell als iLogic gesteuerte Baugruppe, bestehend aus mehreren Bauteilen.

Die Steuerungsparameter welche die Varianten im FDU Asset steuern sind dieselben wie in der Fertigungsmodell Baugruppe. Das bedeutet, es musste nur noch ein Weg geschaffen werden die Parameter vom Asset in eine Kopie der Fertigungsmodells zu übertragen und man hätte jedes Asset als Variante im Fertigungsdetailgrad.

Innerhalb des Assets haben wir einen Hinweis auf das damit verbundene Fertigungsmodell hinterlegt. (Artikelnummer der Vorlage für das jeweilige Fertigungsmodell des Assets)

Per vb.net wurde dann ein kleines Programm geschrieben das folgende Schritte durchführt:

- a. Selektion durch den Nutzer identifizieren
- b. Hinweis auf das Fertigungsmodell aus dem Asset auslesen
- c. Fertigungsmodell im Vault Pro finden und herunterladen
- d. Kopie vom Heruntergeladenen Fertigungsmodell erzeugen
- e. Parameter vom Asset an die Kopie des Fertigungsmodells übertragen
- f. Auf Wunsch das fertiggestellte Fertigungsmodell anzeigen



In unserem Fall haben wir dann noch zu dem passenden Fertigungsmodell eine Zeichnung erstellt, die dann beim Kopieren ebenfalls mitgezogen wird.

Das bedeutet der Schritt zur fertigen Zeichnung besteht aktuell darin, das die Maße und Ansichten auf der Zeichnung je nach Wunsch hingerückt werden müssen. Der Rest ist fertig.

Die ursprünglichen Schritte 1-5:

1. Der Leitungsverlauf wurde im Anlagenmodell grob skizziert
2. Dann wurde auf Basis dieser Skizze ein Sweeping erzeugt
3. Dieses Sweeping wurde dann in mögliche Komponenten zerlegt
4. Diese Komponenten wurden dann als Baugruppe modelliert (teils auf Basis existierender Teile, teils neu)
5. Aus den Teilen wurden dann Zeichnungen erstellt

wurden nun reduziert auf die folgenden Schritte:

1. Kanal planen mit Hilfe der 9 FDU Assets
2. Assets selektieren und .net Programm ausführen
3. Zeichnungen überprüfen und kleinere Änderungen durchführen

Die Komplexität für den Anwender besteht nun nur noch in Punkt 1 und dem korrekten Kombinieren und Einstellen der einzelnen Assets um die gewünschte Leitung zu erhalten.

Ein Nebeneffekt dieser ganzen Optimierung war auch noch die Standardisierung einiger Elemente, was dann zu den bekannten Vorteilen geführt hat, welche eine Varianten Reduzierung mit sich bringt.

Umsetzung

Im nachfolgenden Teil geht es um die Umsetzung der einzelnen Themenbereiche.

Erste Vorbereitungen in den Inventor Factory Design Utilities (FDU)

Die FDU ist ein Feature welches sehr viele hilfreiche Tools mit an Bord hat, die der Standard in Inventor nicht bietet.

Wenn es nicht bereits installiert ist, dann steht die FDU allen Nutzern der PDMC zur Verfügung und kann bei den Autodesk Produkten heruntergeladen werden.

In der FDU werden so genannte Assets (kleine vorbereitete Elemente) erstellt und in eine Bibliothek hineingeladen (lokal oder global).

Generell kann jedes Modell aus Inventor (Baugruppe oder Einzelteil) als Asset abgelegt werden.

Bei uns hat es sich bewährt spezielle vereinfachte Modelle zu verwenden, die nur als *.ipt abgelegt werden. Das hängt vor allem an den Größen unserer Anlagen und Maschinen.

Die Festlegung war, dass wir 9 Assets benötigen für die Erstellung aller möglichen Leitungen. Also wurden zunächst Modelle dieser 9 Assets erzeugt. Zudem wurde ermittelt, welche Größen möglich sein sollten. Also Breite von bis, Höhe von bis etc.

Alle notwendigen Parameter wurden erstellt. Hier zeigt sich der Vorteil eines vereinfachten ipt-Modells. Gerade für das Ausprobieren der einzelnen Varianten ist es deutlich schneller alle Parameter in einem Modell zu sehen und nicht durch verschachtelte Baugruppen zu navigieren.

Wo es nötig war, wurden sogenannte Multi-Value Parameter erzeugt, also Parameter mit Auswahllisten.

Um Konflikte mit internen Namensvergaben von Inventor zu haben, schreiben wir bei uns immer dim_ vor den Parametern.

Es wird dann noch festgelegt, welche Parameter Schlüsselparameter sein werden. Diese so genannten Key Parameter sind nachher auch die Parameter, welche innerhalb der Factory Properties im Factory Layout (Der Factory Baugruppen Vorlage) zum Editieren angefasst werden können.

Das ist dann das Grundgerüst für die weitere Parametrisierung.

FDU Asset Parametrisierung mit iLogic

Nun geht es an die Parametrisierung der simplen Assets per iLogic.

Dabei werden schon alle Steuerungen eingebaut, die auch später bei dem eigentlichen Fertigungsmodell zum Einsatz kommen.

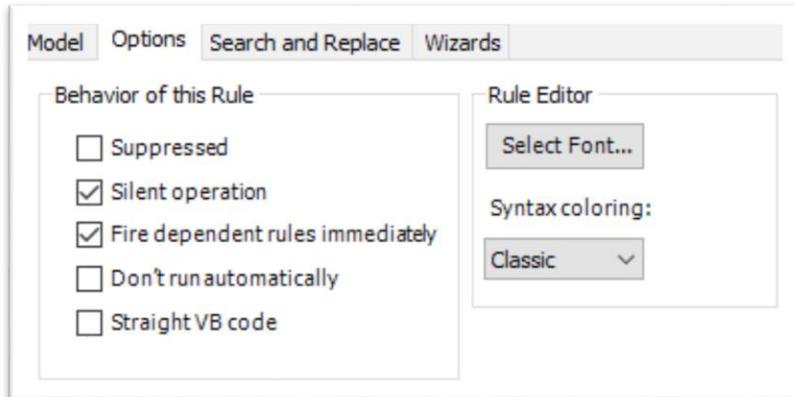
An diesem einfachen Modell lassen sich alle Varianten sehr leicht umsetzen und testen. Ohne großen Aufwand.

Es kann also geprüft werden ob alle Varianten vorhanden sind? Fehlen noch Bauformen? Fehlen Parameter? Passt alles zusammen?

Generell wird eine neue iLogic Regel bei uns immer nach demselben Prinzip erstellt.

iLogic Regel Optionen

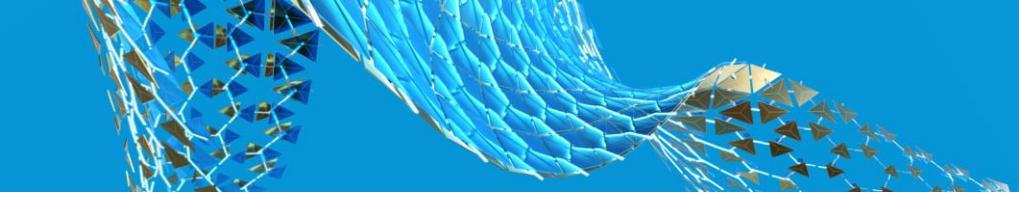
In den Optionen werden die Optionen: „Silent Operation“ und „Fire dependent rules immediately“



Anspringen der Regel bei Änderungen eines bestimmten Parameters

Die Regeln werden so erstellt, dass sie bei einer Änderung eines bestimmten Parameters anspringen und bestimmte Aufgaben erfüllen. Nicht wundern, ein genereller Start bei Parameteränderungen lässt sich auch über sogenannte iLogic Event Trigger behandeln z.B. „Any Model Parameter Change“, aber hier soll es spezifischer sein.

Das erreichen wir durch einen simplen Trick.



Da die Regeln eigentlich nur auf Parameteränderungen der Key Parameter, also der Parameter, die durch den Nutzer variiert werden können, reagieren sollen, werden diese zu Beginn der betreffenden Regel geschrieben nach diesem Muster:

```
dim_Key_Parameter1 = dim_Key_Parameter1  
dim_Key_Parameter2 = dim_Key_Parameter2  
...
```

Dadurch wird bei jeder Parameteränderung die Regel ausgeführt.

Allgemeiner Aufbau unserer iLogic Regeln

Der Rest ist dann einfache Syntax wie:

Wenn dim_Key_Parameter1 = WertX, dann tue dies etc.

Am Ende wird eine Regel immer mit dieser Zeile abgeschlossen:

```
iLogicVb.UpdateWhenDone = True
```

Diese Zeile bedeutet, dass Inventor das Dokument aktualisieren soll. Also alle Änderungen die vorgenommen wurden innerhalb der Regel auch wirklich im Modell ankommen.
Bekannt ist der Aufruf manuell durch das Klicken auf den üblichen Inventor Update Blitz.

Ob jetzt innerhalb der Regel Unterfunktionen etc. programmiert werden, entscheiden Sie als Nutzer ihren Fähigkeiten entsprechend.

Generell gilt, dass ein Großteil von dem was mit der Inventor API innerhalb von .net machbar ist, funktioniert auch innerhalb von iLogic.

Daher kann der Nutzer auch auf viele Quellen zugreifen in denen API Beispiele für VB.net angesprochen werden.

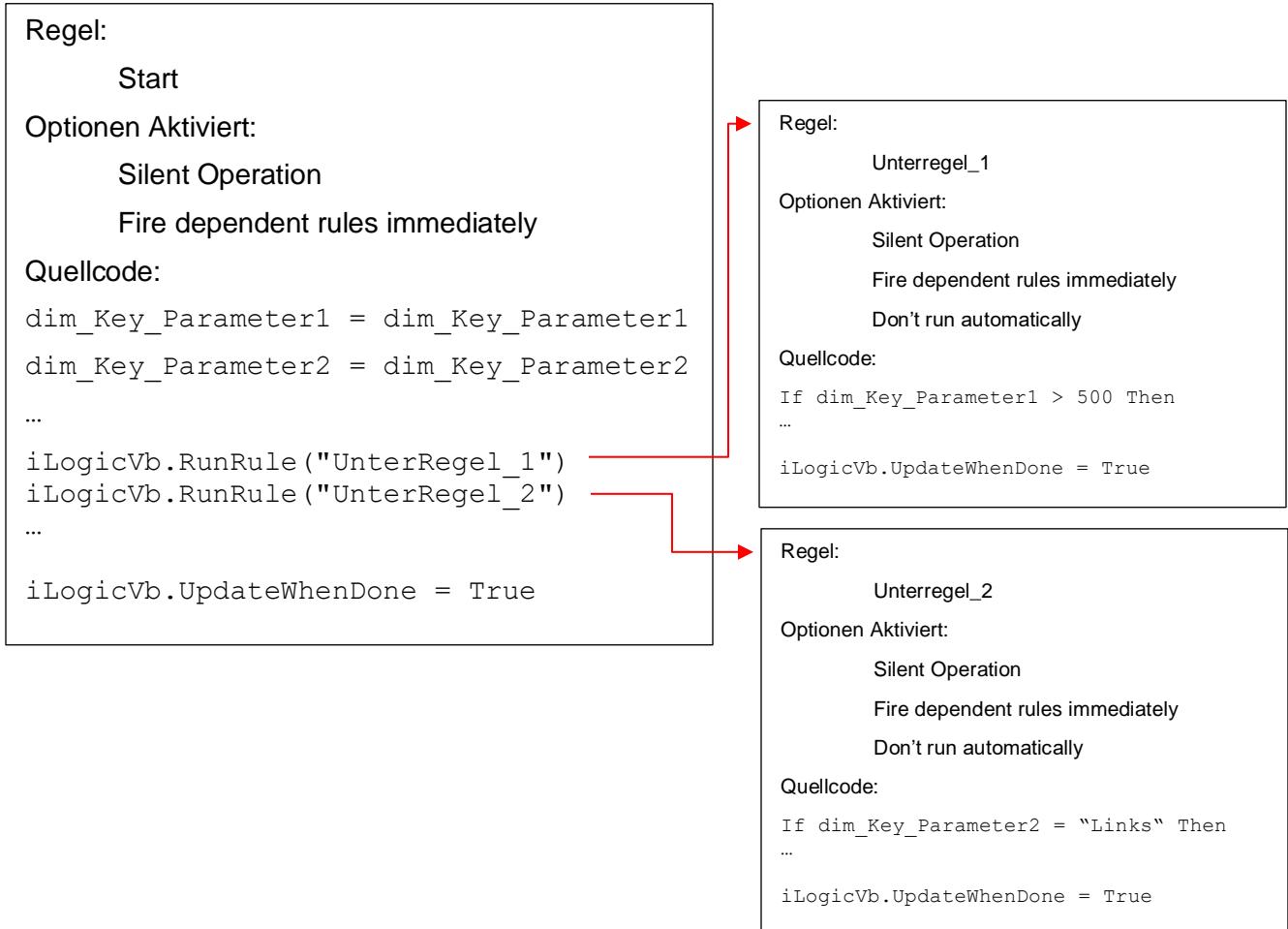
Alleine diese Erkenntnis eröffnet meistens schon die Tür zu ganz neuen Möglichkeiten.

Wenn die Assets komplexer werden, empfiehlt es sich, mit mehreren Regeln zu arbeiten. Diese Unterregeln würden dann noch in den Optionen „Don't run automatically“ angewählt bekommen und würden dann aus einer Hauptregel, die wir meistens „Start“ nennen, gesteuert.

„Don't run automatically“ bedeutet, dass die Regel eben nicht automatisch anspringt, wenn sich ein Parameter verändert, welcher innerhalb der Regel verwendet wird.

So sieht dann das Schema aus:

Das kann dann selbstverständlich ganz nach Geschmack und Komplexität erweitert werden.

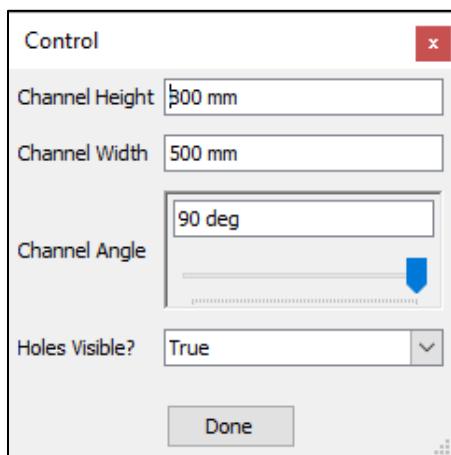


Wissenswertes zu iLogic z.B. Forms

In iLogic sind der Kreativität nahezu keine Grenzen gesetzt.

Vor allem durch das Nutzen von Bedingungen lassen sich schnell vielfältige Varianten darstellen.

Durch die Verwendung von Forms lassen sich auch Auswahldialoge erzeugen. Damit ist die Steuerung der Bauteile noch einfacher. Bereichsdefinitionen, oder Schrittgrößen um ein Bauteil in bestimmten Schritten zu verändern lassen sich so leicht umsetzen (z.B. 50mm Schritte bei der Länge)

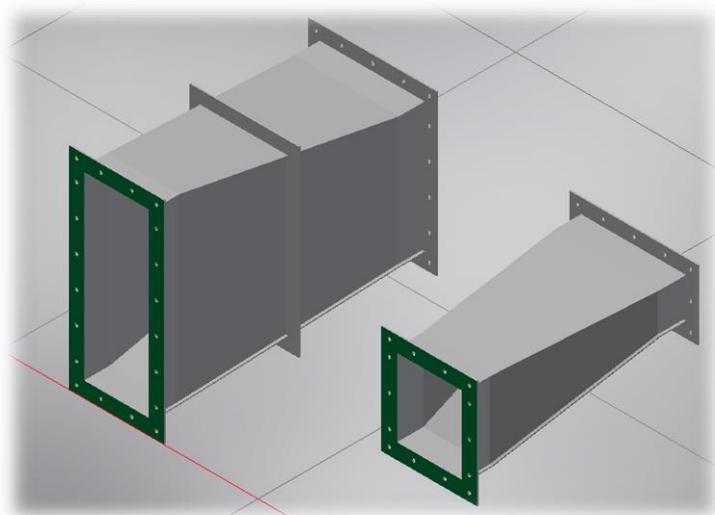


Je nachdem, ob ein Inventor Parameter als Multivalue Parameter angelegt wurde, wird dieser automatisch als Drop Down Liste in der Form dargestellt.

Zwar sind die Möglichkeiten die Form, also den Dialog, zu steuern nicht so vielfältig wie vielleicht von .net oder VBA gewohnt, aber für die meisten Anwendungen absolut ausreichend und dazu noch leicht per Drag'n Drop zu realisieren.

Die Ergebnisse lassen sich dann sehr leicht überprüfen.

Mit ein paar Änderungen in der Form lassen sich dann solche Ergebnisse erzielen wie hier gezeigt:



Durch iLogic lässt sich massiv an Zeit für Änderungen an Parametern sparen. Statt in die Parameterabelle zu gehen und dort Werte einzugeben und dann das Modell zu aktualisieren etc. kann dies mit einer Eingabe in der iLogic Form vorgenommen werden.

Das spart Mausklicks und Co. und damit wertvolle Zeit. Zudem lässt sich durch das Layout der Forms auch ganz gezielt der Nutzer steuern. Vor allem Nutzer, die nicht täglich mit einem Teil zu tun haben sehen nicht immer auf dem ersten Blick, welche Parameter welche Auswirkung hat und noch dazu nicht welche Parameter durch den Nutzer verändert werden sollten und welche nicht.

Hier der Vergleich der Parameterliste zur iLogic Form. In der Parameterliste sieht der Nutzer den Wald vor lauter Bäumen nicht und im Dialog ist alles wunderbar aufgeräumt:

Parameters										
Parameter Name	Consumed	Unit/Ty	Equation	Nominal Val	Driving Rule	Tol.	Model Value	Ke	Comment	
d113	Sketch6	mm	dim_x	30,000000		(Yellow)	30,000000	<input type="checkbox"/>		
d114	Sketch6	mm	dim_x	30,000000		(Yellow)	30,000000	<input type="checkbox"/>		
d115	UCS1	mm	-350,000000000 mm	-350,00...		(Yellow)	-350,00...	<input type="checkbox"/>		
d116	UCS1	mm	-50,000000000 mm	-50,000...		(Yellow)	-50,000...	<input type="checkbox"/>		
d117	UCS1	mm	0,000000000 mm	0,000000		(Yellow)	0,000000	<input type="checkbox"/>		
d118	UCS1	deg	0,000000000 deg	0,000000		(Yellow)	0,000000	<input type="checkbox"/>		
d119	UCS1	deg	0,000000000 deg	0,000000		(Yellow)	0,000000	<input type="checkbox"/>		
d120	UCS1	deg	90,000000000 deg	90,000000		(Yellow)	90,000000	<input type="checkbox"/>		
d121	UCS2	mm	50,000000000 mm	50,000000		(Yellow)	50,000000	<input type="checkbox"/>		
d122	UCS2	mm	350,000000000 mm	350,000...		(Yellow)	350,000...	<input type="checkbox"/>		
d123	UCS2	mm	-0,000000000 mm	0,000000		(Yellow)	0,000000	<input type="checkbox"/>		
d124	UCS2	deg	0,000000000 deg	0,000000		(Yellow)	0,000000	<input type="checkbox"/>		
d125	UCS2	deg	0,000000000 deg	0,000000		(Yellow)	0,000000	<input type="checkbox"/>		
d126	UCS2	deg	-0,000000000 deg	-0,000000		(Yellow)	-0,000000	<input type="checkbox"/>		
Reference Par...										
User Parameters										
dim_r	d107	mm	0,7 mm	0,700000	(Yellow)				Radiusfaktor des	
dim_Flange_t	d14, d11...	mm	6 mm	6,000000	(Yellow)					
dim_L1	d1	mm	44 mm	44,000000	(Yellow)					
dim_L2	d2	mm	44 mm	44,000000	(Yellow)					
dim_Flange...	d110, d109	mm	60 mm	60,000000	(Yellow)					
dim_H	d5	mm	800 mm	800,000...	Flange_Holes	(Yellow)				
dim_W	d107, d1...	mm	500 mm	500,000...	Flange_Holes	(Yellow)				
dim_Holes	True...	True		Holes						
dim_x	d114, d1...	mm	30 mm	30,000000	(Yellow)					
dim_dist_H...	d105, d1...	mm	120 mm	120,000...	(Yellow)					
dim_dia_Hole	d41, d33...	mm	14 mm	14,000000	(Yellow)					
dim_Versatz		mm	1,0 mm	1,000000	(Yellow)					
dim_Alpha	d8	deg	90 deg	90,000000	(Yellow)					
dim_t	d7	mm	4 mm	4,000000	(Yellow)					
dim_y	d32	mm	100 mm	100,000...	(Yellow)					
dim_y2	d34	mm	70 mm	70,000000	(Yellow)					

Control

Channel Height: 300 mm

Channel Width: 500 mm

Channel Angle: 90 deg

Holes Visible?: True

Done

Add Numeric | Update | Purge Unused

Link | Immediate Update

Reset Tolerance: + ▲ ○ -

<< Less | Done

Automatische Namensvergabe der FDU Asset Ausprägungen

Bisher ist es so, dass jede Änderungen an den Parametern, ob manuell, per Form oder wie auch immer, jedes Mal das vorhandene IPT verändert. Vor einer Änderung müsste also immer zunächst eine Kopie erzeugt werden, und diese Kopie wird dann verändert, damit es mehrere Varianten gibt.

Auch hier hilft die FDU.

Denn die FDU übernimmt automatisch diesen Kopiervorgang. Das bedeutet jede Veränderung an dem eingefügten Asset bewirkt die Erzeugung einer Kopie und damit einer neuen Variante.

Wird eine Ausprägung gewählt, die bereits früher erstellt wurde, so wird automatisch diese bereits vorhandene Version ausgewählt, anstatt eine neue Variante zu erzeugen. (Bei der Nutzung von Vault würde dann vor dem Einchecken geprüft werden, ob bereits eine entsprechende Ausprägung existiert.)

Dazu muss das vorbereitete Teil nur als Asset abgespeichert und dann in einem neuen Factory Layout eingefügt werden. Dort kann es dann beliebig oft verändert werden.

Werden Änderungen am Asset vorgenommen und diese sollen später auf bereits eingesetzte Assets übernommen werden gibt es die Funktion Update Asset.

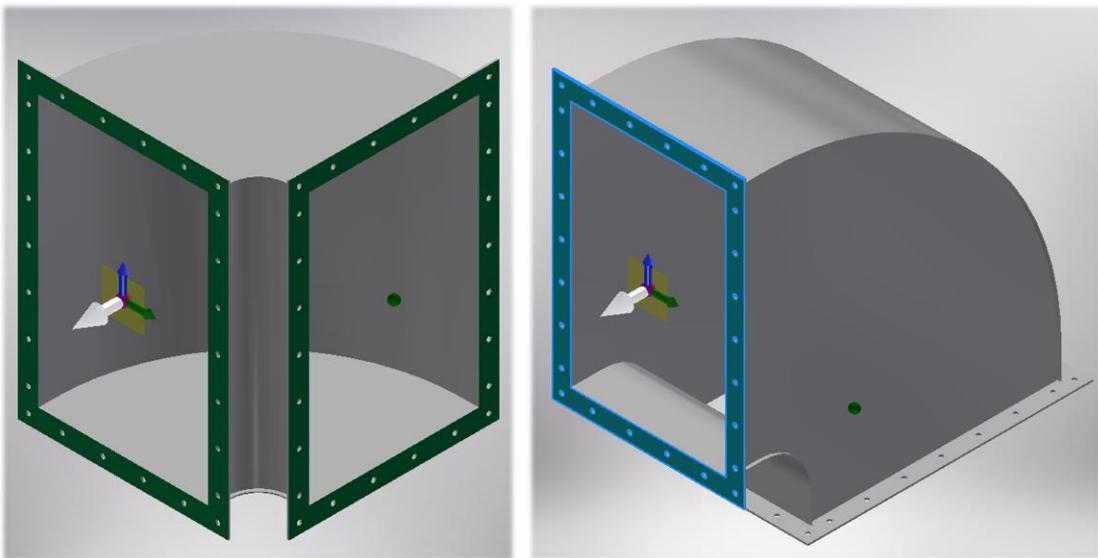
Dort lässt sich erkennen, ob bereits eingefügte Assets in der Asset Bibliothek eine Änderung erfahren haben die sich auf die vorhandenen Assets im Layout auswirken.
Parametereinstellungen etc. bleiben dabei erhalten.

 Injection Duct Straight_D800BD9612359F023561140581F5EA4A.ipt	22.01.2020 09:06
 Injection Duct Straight_D1641AB0153AB2986019099D8D6CC07B.ipt	30.01.2020 14:18
 Injection Duct Straight_E3D714AD0D22C42AB9F6701CAAB72F31.ipt	20.01.2020 14:11
 Injection Duct Straight_E5F071FC0906F8C773EC78927F3C937D.ipt	29.01.2020 13:16
 Injection Duct Straight_E45A951450F08F97A9C16140AE48761F.ipt	23.01.2020 10:49
 Injection Duct Straight_F08AAA45E353F758BCAD4D94D6159271.ipt	06.02.2020 12:50
 Injection Duct Straight_F064A06C327DBEF400A541F6F7CEF1F8.ipt	23.01.2020 10:50
 Injection Duct Straight_F549ECA70C1CF7C3C3590D87604B47B0.ipt	31.01.2020 11:28
 Injection Duct Straight_F81756AAF2821E59064F3F6D2E014910.ipt	13.01.2020 12:35
 Injection Duct Transition_0D0E3C6A8D3672F71BBBE72EB7D396A4.ipt	06.02.2020 14:43
 Injection Duct Transition_0E8A1D5A8ABC17381DA17DCB8271B7B5.ipt	23.01.2020 11:34
 Injection Duct Transition_0F36F5266E2237C9D96AEF87D7039A50.ipt	23.01.2020 10:49
 Injection Duct Transition_1E9719EC6C3414A0C35C73B2AFCFE197.ipt	03.02.2020 13:57
 Injection Duct Transition_2C7506C7957B2179E0EE28B6BE6ECE55.ipt	03.02.2020 14:25
 Injection Duct Transition_3EDE561172840E4B2EE38E7FFD06BE5F.ipt	06.02.2020 14:43
 Injection Duct Transition_4B7342F0B186E6AF4BB13AFC767AA07B.ipt	10.02.2020 09:15
 Injection Duct Transition_4BFEB502249257F0C6B5F74A245B9048.ipt	03.02.2020 13:58
 Injection Duct Transition_4D365F142477222BF52DBEC6F1A374C9.ipt	22.01.2020 09:04
 Injection Duct Transition_4DD0EE3AE1C66E8E2B545E5175163D49.ipt	25.09.2020 13:39
 Injection Duct Transition_5A3E484201977FD926A0078F351E1927.ipt	10.02.2020 09:10

FDU Connectoren

Einer der größten Vorteile bei der Nutzung der FDU ist die sogenannte Vererbung bei der Verwendung von Connectoren.

Connectoren? Das sind Verbindungen, die vordefiniert werden. So lassen sich später einfach Teile verbinden, ohne aufwendig die Teile mit klassischen Inventor Abhängigkeiten oder Joints zu verbinden.



Diese Connectoren lassen sich nur in den bekannten 3 Achsen X, Y und Z definieren.

Wobei immer 2 Achsen festgelegt werden und die 3. entsprechend folgt. Das ist auch der Grund, warum es bei uns horizontale und vertikale Bögen gibt.

Die Hauptachse zeigt bei uns immer vom Bauteil weg und die zweite festzulegende Achse ist immer nach Oben gerichtet. Dieses Muster behalten wir bei allen Kanalteilen bei. Nur so ist gewährleistet, dass die Verbindungen immer korrekt ausgerichtet sind.

Parameter Vererbung mit FDU Connectoren

Connectoren sind alleine deswegen schon nützlich weil es deutlich einfacher ist, Teile miteinander zu verbinden, aber die Vererbung ist noch viel besser.

Angenommen, man hat eine Rohrleitung mit einem Durchmesser von 100mm

Soll nun die Leitung erweitert werden, dann würde normalerweise ein weiteres Segment der Baugruppe hinzugefügt werden und wird dann per Abhängigkeiten platziert. Nun muss dieses neu eingefügte Segment nicht zwangsweise auch 100mm Durchmesser besitzen. Das bedeutet, ein Editieren des neuen Segments ist notwendig.

Mit der Vererbung der Connectoren ist das ein Schritt. Segment dem vorhandenen Rohr hinzufügen und der Durchmesser Parameter würde auf das neue Segment übertragen werden und dieses automatisch anpassen.

Dasselbe funktioniert hier auch mit den Kanälen. Die Länge und Breite der Kanäle wird vererbt. Ob es sich bei dem neuen Segment nun um ein Übergangsstück, Rohrbogen oder Y-Verteiler handelt, alle passen sich automatisch an den vorhandenen Teil an.

Wichtig dabei ist nur zu wissen, dass bei der Verwendung vom „Connect“ Befehl immer zuerst der neue Teil angeklickt wird und dann der Vorhandene. Ist die Reihenfolge andersrum, so wird die Größe auf den vorhandenen Teil übertragen.

Die Steuerung dieser Vererbung erfolgt über sogenannte Connector Class Dateien.

Die Connectorclass Dateien sind XML Konfigurationsdateien. Es empfiehlt sich also auf einen XML Editor zurückzugreifen.

Connectorclass Dateien werden in diesem Standardverzeichnis abgelegt

```
C:\ProgramData\Autodesk\Factory Design <Version>\FactoryLibrary\ConnectorClasses\
```

Die ConnectorClass Dateien werden nicht so oft bearbeitet, sondern nur einmal zu Beginn festgelegt und immer wenn ein neues Asset dazukommt eventuell ergänzt.

Aus dem Grund kann ich nur empfehlen die Einstellungen zu dokumentieren damit spätere Änderungen leichter nachvollziehbar sind.

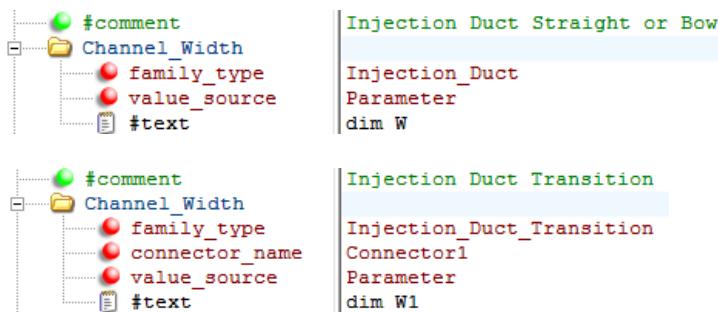
<pre> xml └── Connector ├── _schemaversion_ └── #comment └── Channel_Width ├── family_type ├── value_source └── #text └── Channel_Height ├── family_type ├── value_source └── #text └── #comment └── Channel_Width └── Channel_Height └── Channel_Width ├── family_type ├── connector_name ├── value_source └── #text └── Channel_Height ├── family_type ├── connector_name ├── value_source └── #text └── Channel_Width ├── family_type ├── connector_name ├── value_source └── #text └── Channel_Height ├── family_type ├── connector_name ├── value_source └── #text └── #comment </pre>	<pre> version="1.0" encoding="utf-8" 1.0000 Injection Duct Straight or Bow Injection_Duct Parameter dim_W Injection_Duct Parameter dim_H Injection Duct Y-Piece Injection Duct Transition Injection_Duct_Transition Connector1 Parameter dim_W1 Injection_Duct_Transition Connector1 Parameter dim_H1 Injection_Duct_Transition Connector2 Parameter dim_W2 Injection_Duct_Transition Connector2 Parameter dim_H2 </pre>
---	--

Im Bild sind 2 exemplarische Verbindungstypen definiert. Einmal die simple „Eine Größe gilt für den gesamten Kanal“ Verbindung: Injection Duct Straight or Bow (Das meint die Elemente wo sich die Einlaufgröße nicht von der Auslaufgröße unterscheidet.)

Und dann noch der Verbindungstyp, indem sich der Einlauf vom Auslauf in der Größe unterscheidet. Ein typisches Übergangsstück also.

Hier ist deutlich zu erkennen, dass es bei dem Connector mit nur einer Größe nur einen Parameter für Breite und Höhe gibt und bei dem Übergangsstück Connector 2 an der Zahl existieren H1 und H2 und B1 und B2.

Zudem gibt es spezifische Namen für die Connectoren: Connector1 und Connector2



Was bedeutet das im Detail?

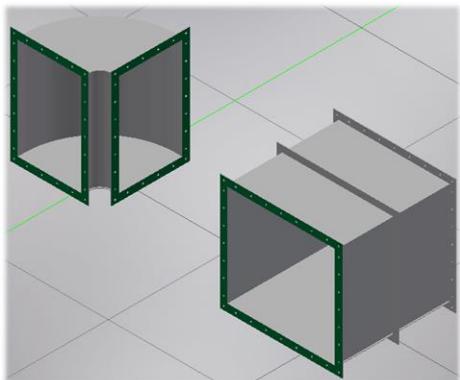
Es wird im Fall, das ein Connector vom Typ „Injection Duct“ an einen Zweiten vom Typ „Injection Duct“ angeschlossen wird, der Parameterwert vom Parameter dim_W auf das andere Asset und den Parameter dim_W übertragen, so dass am Ende beide Assets denselben Wert im Parameter dim_W stehen haben.

So weit so gut.

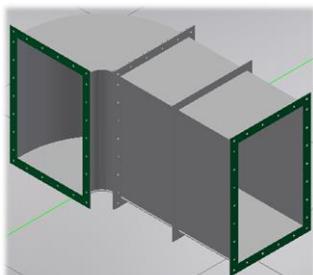
Im Fall das ein Connector vom Typ „Injection Duct“ an einen Connector vom Typ „Injection_Duct_Transition“ angeschlossen wird, wird zunächst einmal für den 2. Typ geschaut, welcher connector_name, z.B. „Connector1“, vorliegt. Das bedeutet es wird jetzt der Wert von dim_W1 vom vorhandenen Asset auf den Wert dim_W des neu eingefügten Assets übertragen.

Wie sieht das dann aus?

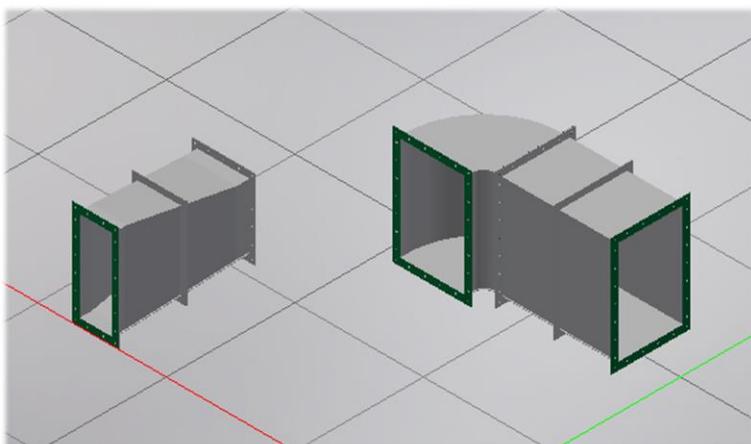
Im Folgenden ein kleines Beispiel wo insgesamt 3 Assets miteinander verbunden werden:



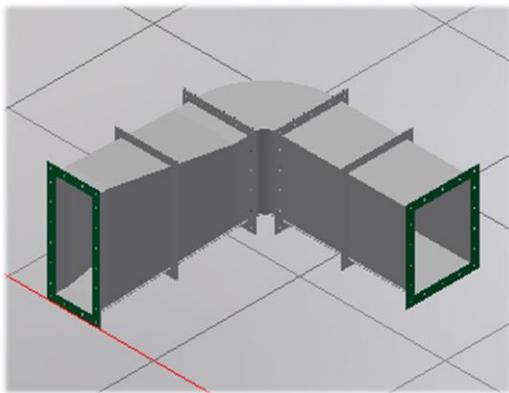
Gerades Stück wird an Bogen angeschlossen per Connector vom Typ „Injection Duct“



Das gerade Stück hat die Querschnittsmaße vom Bogen übernommen.



Nun wird der Bogen mit dem daran verbundenen geraden Stück an ein Übergangsstück verbunden.



Damit wird die Größe vom Querschnitt auf den Bogen übertragen und damit dann auch auf das daran angeschlossene gerade Stück.

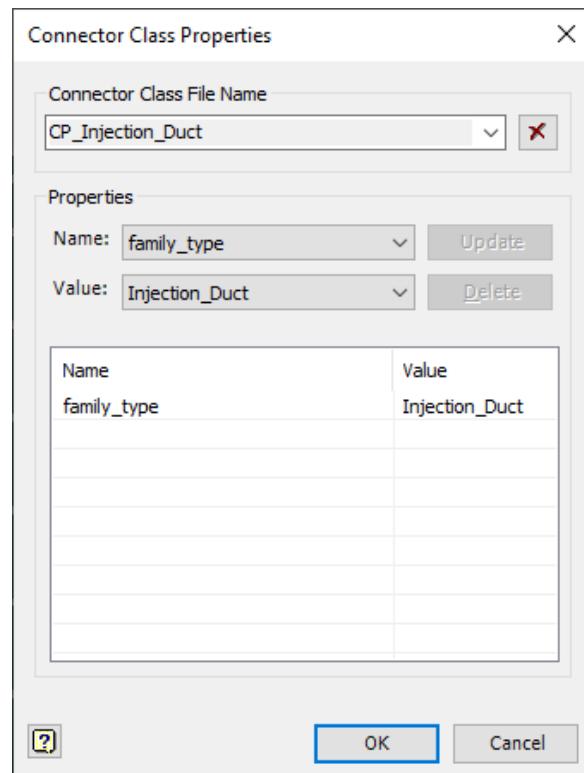
Ohne Betrachtung des Einfügens und der Auswahl der Assets sind insgesamt nur 6 Klicks notwendig gewesen um zu dem Ergebnis zu kommen das alle Teile aneinander angepasst sind.

Je nach Vererbungskomplexität kann so eine Übertragung der Parameter einen Augenblick dauern (es wird eine Kopie erzeugt, die Parameterwerte übertragen und dann die neue Version platziert), aber es ist kein Vergleich zu dem herkömmlichen Prozess in Inventor.

Auf meinem System lag zwischen dem ersten und dem letzten Bild gerade mal eine Zeitspanne von 20 Sek.

Und alles was dafür getan werden muss ist:

1. Eine Connectorclass Datei erzeugen
2. Einen Connector definieren
3. Einen Connectortyp zuweisen:



Beim Arbeiten über die Grenzen des eigenen Arbeitsplatzes hinaus, muss sichergestellt sein, dass die Connectorclass-Datei immer lokal bei dem jeweiligen Anwender liegt.

Ansonsten funktioniert das mit dem Vererben nicht.

Das wird im Normalfall aber leicht bemerkt. Funktioniert das nicht mit der Vererbung nicht, fehlt vermutlich die Datei oder ist nicht aktuell, ganz simpel.

Wir haben noch eine Besonderheit bei unseren Connectoren

Da die FDU bei den Connectoren nicht die Richtung unterscheidet (Bei Kanälen, Rohren mit Abzweigungen etc. nicht uninteressant), haben wir uns auf Farben der Anschlussflächen geeinigt.

Blau	=	Einlauf
Rot	=	Auslauf
Grün	=	Generelle Verbindungsfläche ohne bestimmte Richtung

Dadurch ist es leichter für den Anwender zu erkennen, welches Bauteil wie herum eingesetzt werden muss in einem fortlaufenden Kanal o.ä.

Fertigungsteile erstellen

Was haben wir nun alles erreicht.

Wir haben die benötigte Bibliothek an FDU Assets erstellt.

Das heißt eine Planung mit vereinfachten Teilen ist nun möglich und funktioniert auch ziemlich schnell.

Doch das Ziel ist es, fertigungsgerechte Modelle zu haben in der jeweils korrekten Ausprägung.

Nun wird also ein Modell aufgebaut, das in der Geometrie dem Asset entspricht, aber vom Anspruch her so modelliert wird, wie in Inventor normalerweise ein Fertigungsmodell aufgebaut wird.

Das heißt Blechteile z.B. werden auch als Blechteile erzeugt um davon Abwicklungen zu erstellen.

iLogic in Baugruppen

Zunächst einmal wird diese Baugruppe für eine spezifische Baugröße und Ausprägung modelliert.

Ist diese Variante fertig und sauber modelliert, wird damit begonnen die iLogic Regeln zu erstellen.

Wir haben immer in der obersten Baugruppe eine Regel mit dem Namen „Dimensions“ also Bemaßungen oder Größen.

Diese Regel startet dann die Übertragung der Parameter an die notwendigen Stellen.

Das heißt die Regel wird aktiviert im Falle einer Parameteränderung der Key Parameter. In diesem Fall die Parameter die später von dem Asset an das detaillierte Modell übertragen werden sollen.

Die Baugruppen Regel überträgt dabei Parameterwerte an die untergeordneten Teile oder Unterbaugruppen, die dann wieder ihrerseits auf die Parameter reagieren. Damit hat jedes untergeordnete Element für sich die Kontrolle über die darin enthaltenden Parameter.

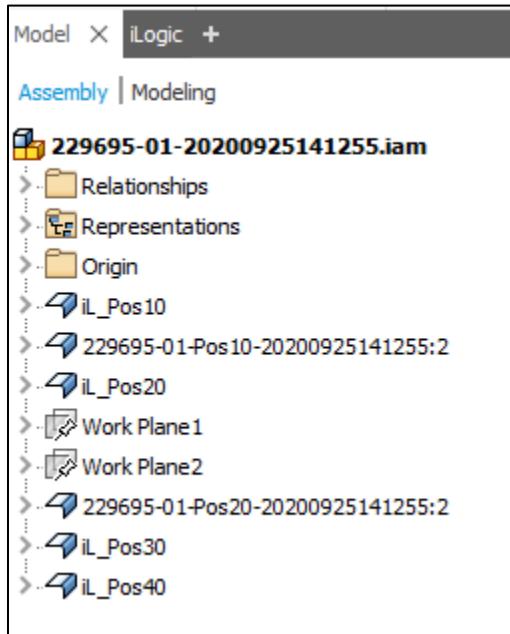
Also es wird nicht von oben herab ein Teil gesteuert, sondern nur etwas übertragen und dann im jeweiligen Unterelement wird darauf reagiert.

Da in der Baugruppe die Bauteile später automatisch neue Namen bekommen müssen, also auch alle untergeordneten Teile und Baugruppen, ist es wichtig, dass die Vererbung der Informationen innerhalb von Inventor rund läuft.

Also mit anderen Worten, wenn man in iLogic ein untergeordnetes Element über den Dateinamen oder ursprünglichen, veränderlichen Element Namen anspricht, dann funktioniert die iLogic Regel nur ein einziges Mal und dann nicht mehr, weil sich der Name verändert.

Aus dem Grund haben wir alle Elemente die im iLogic angesprochen werden mit einem vorangestellten iL benannt.

Die anderen Elemente sind egal, weil sie z.B. doppelt vorkommen etc.



Hier ein Beispiel für die Baugruppen Regel:

Dort ist dann zu sehen, wie die Übermittlung der Parameter an die Unterelemente abläuft.

```
dim_W = dim_W
dim_H = dim_H
dim_Alpha = dim_Alpha
dim_r = dim_r
dim_L1 = dim_L1
dim_L2 = dim_L2
```

Segment damit die Regel beim Ändern der benannten Parameter startet

```
Parameter("iL_Pos10", "dim_W") = dim_W
Parameter("iL_Pos10", "dim_H") = dim_H

Parameter("iL_Pos20", "dim_W") = dim_W
Parameter("iL_Pos20", "dim_Angle") = dim_Alpha
Parameter("iL_Pos20", "dim_r_Factor") = dim_r

Parameter("iL_Pos30", "dim_W") = dim_W
Parameter("iL_Pos30", "dim_H") = dim_H
Parameter("iL_Pos30", "dim_Angle") = dim_Alpha
Parameter("iL_Pos30", "dim_r_Factor") = dim_r

Parameter("iL_Pos40", "dim_W") = dim_W
Parameter("iL_Pos40", "dim_H") = dim_H
Parameter("iL_Pos40", "dim_Angle") = dim_Alpha
Parameter("iL_Pos40", "dim_r_Factor") = dim_r

Parameter("iL_Pos20", "dim_L1") = dim_L1
Parameter("iL_Pos30", "dim_L1") = dim_L1
Parameter("iL_Pos40", "dim_L1") = dim_L1

Parameter("iL_Pos20", "dim_L2") = dim_L2
Parameter("iL_Pos30", "dim_L2") = dim_L2
Parameter("iL_Pos40", "dim_L2") = dim_L2
```

Segment für die Übertragung der Parameter an die Unterelemente

```
iLogicVb.UpdateWhenDone = True
```

Ende und Aktualisierung des Inventor Dokumentes

In dem jeweiligen Unterelement z.B. Flansch gibt es dann eine Regel die wiederum reagiert auf die Parameter die sich ändern. Und ihrerseits das Teil anpassen.

Das war es dann auch schon.

Es ist in dem Falle egal, ob also eine Parameteränderung in der übergeordneten Baugruppe von Hand durchgeführt wird, oder aber ob von außen ein Parameter per Software übertragen wird.

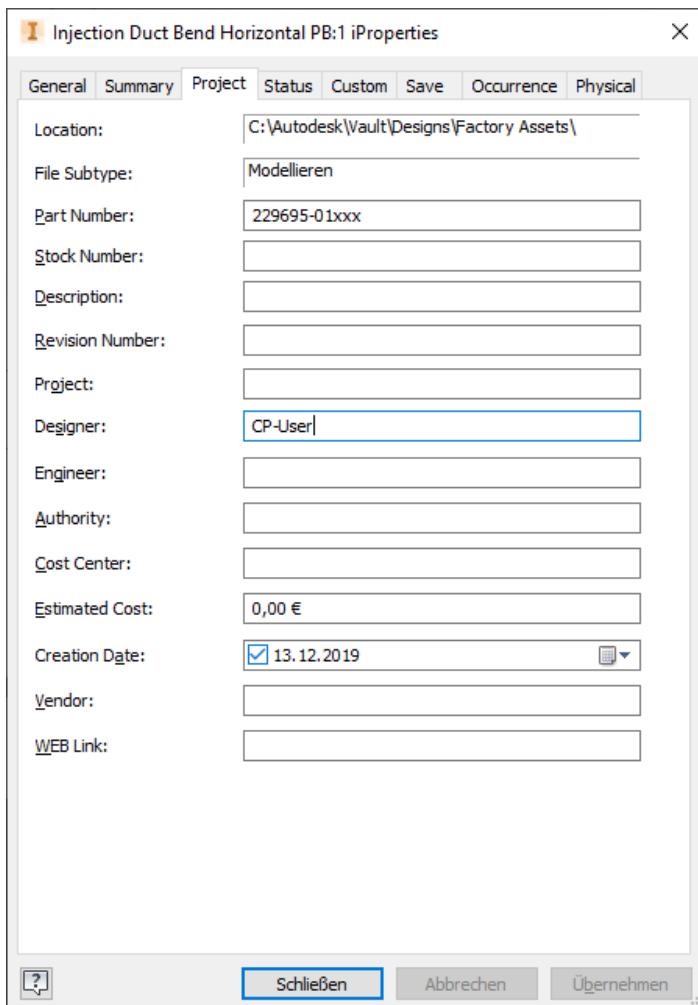
Und jetzt kommen wir zur nächsten Überlegung. Die Parameter sollen aus dem Asset übertragen werden. Aber woher soll das Asset wissen, welches Fertigungsteil dazugehört.

Es soll später möglichst automatisiert erfolgen.

Alle unsere Bauteile erhalten TeileNr. Diese Nummern sind eindeutig. Also ein Bauteil hat eine TeileNr. hat der Nutzer die TeileNr. hat er auch das Bauteil.

Im Asset wird bei uns in den iProperties ein Hinweis auf das Fertigungsmodell gesetzt:

(Im Feld „Part Number“)



Das xxx steht nur als Identifikation, dass es sich bei der eingetragenen PartNo wirklich um ein Asset Hinweis auf das Fertigungsmodell handelt.

Wir setzen bei uns Vault Professional ein und das machen wir uns an dieser Stelle zu nutze.

Hier sei erwähnt dass man es auch ohne Vault weiterführen kann, dort dann die Suche nicht im Vault sondern im Dateisystem vorgenommen wird.

Die Fertigungsbaugruppe wird im Vault unter ihrer TeileNr. gespeichert. Ist somit also leicht zu finden unter dem Suchbegriff der entsprechenden TeileNr.

Es gibt zu dem jeweiligen Fertigungsmodell als Baugruppe auch immer eine Zeichnung. Beide tragen die identische Nummer.

Mit der Kenntnis über die Zeichnungsnummer findet man per Abfrage im Vault auch sehr leicht alle dazugehörigen Dateien.

Der Vorgang in Kürze:

1. Selektieren eines Assets
2. Auslesen der PartNo. für die Suche des Fertigungsmodells im Vault
3. Suche im Vault
4. Herunterladen aller dazugehörigen Dateien aus dem Vault
5. Erstellen einer lokalen Kopie der Dateien von 4.
6. Übertragen der Parameter Werte vom Asset auf die Kopie (5.)
7. Ab hier übernimmt dann das Fertigungsmodell und reagiert entsprechend auf die übertragenden Parameter (6.)

Schritt 1, 2, 5, 6, 7 lassen sich auch mit iLogic umsetzen ohne vb.net, die Schritte 3 und 4 nur mit einem gewissen Aufwand, daher bietet sich hier die Umsetzung in .net an.

Hier ist die Hauptfunktion dargestellt. Damit sie einfacher zu lesen ist, wurden die Objekte und Funktionen in Deutsch geschrieben.

Das ist das Selektieren vom Asset:

```
Dim Quelle As Inventor.ComponentOccurrence
Quelle =
InventorApp.CommandManager.Pick(SelectionFilterEnum.kAssemblyOccurrenceFilter
, "Bitte das FDU Asset selektieren.")
```

Hier wird die Fertigungsmodell TeileNr. ausgelesen:

```
Dim Eigenschaft As PropertySet =
Quelle.Definition.Document.Propertysets.Item("Design Tracking Properties")
Dim Nummer As String = Eigenschaft.Item("Part Number").Value
```

Das ist nur die Überprüfung ob es sich bei der Nummer um eine Asset Fertigungsmodellnummer handelt:

```
If Nummer.Contains("xxx") Then
    Nummer = Nummer.Replace("xxx", "")
Else
    MsgBox("Dokument konnte nicht gefunden werden: " & Nummer & ".dwg")
    Exit Sub
End If
```

Das hier ist die Suche nach einer Zeichnung mit der entsprechenden TeileNr:

```
Vault_Funktionen.Suche(Nummer & ".dwg")
```

Hier werden in einem Schritt Dateien aus dem Vault abgerufen und lokal kopiert:

```
Dim NeuerName As String =
Kopier_Funktionen.Erstelle_Kopie(Vault_Funktionen.Get_Vault_File(Nummer &
".dwg"), Nummer, Nummer, True)
```

Hier wird die neu kopierte Baugruppe geöffnet:

```
Dim Ziel As Inventor.Document =
Inventor1.Documents.Open(System.IO.Path.GetFullPath(NeuerName).Replace(System
.IO.Path.GetExtension(NeuerName), ".iam"), True)
```

Und dann entsprechend aus dem Asset die Parameter übertragen an die eben geöffnete Baugruppe:

```
TransferParameter(Quelle.Definition.Document, Ziel)
```

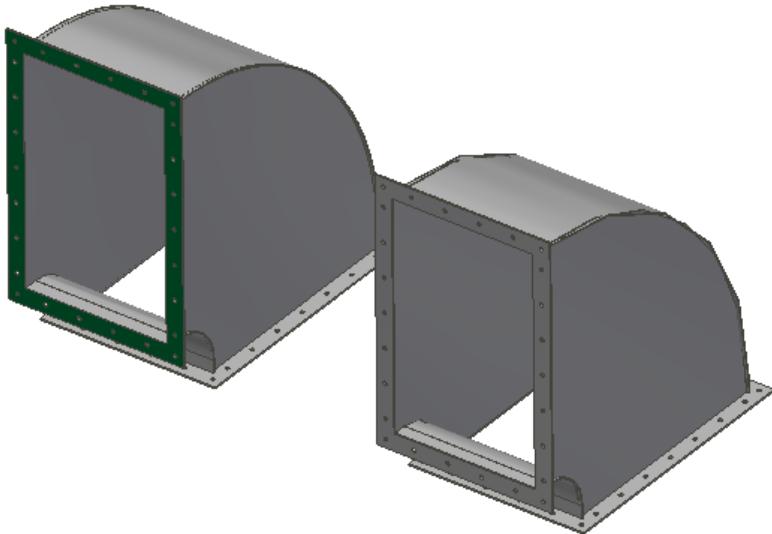
Zuletzt wird noch die dazugehörige Zeichnung geöffnet und eine Meldung auf dem Bildschirm angezeigt, dass der Vorgang erfolgreich war:

```
Inventor1.Documents.Open(NeuerName, True)
MsgBox("Done")
```

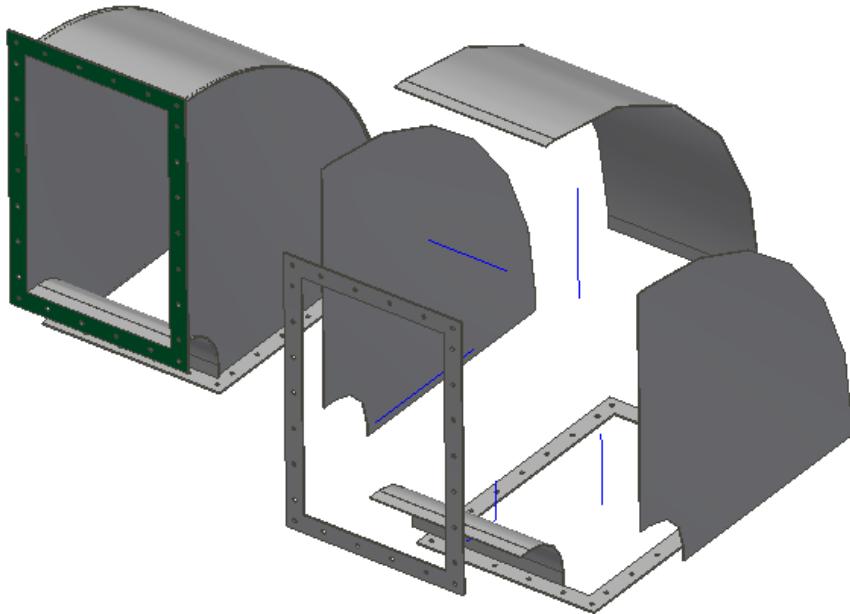
Die mit gelb unterlegten Funktionen sind neu erstellt worden und nicht im Standard enthalten.
Alle anderen Funktionen schon.

Das Ergebnis sieht nun so aus:

(Links Asset und Rechts das Fertigungsmodell)



Hier der „Beweis“:

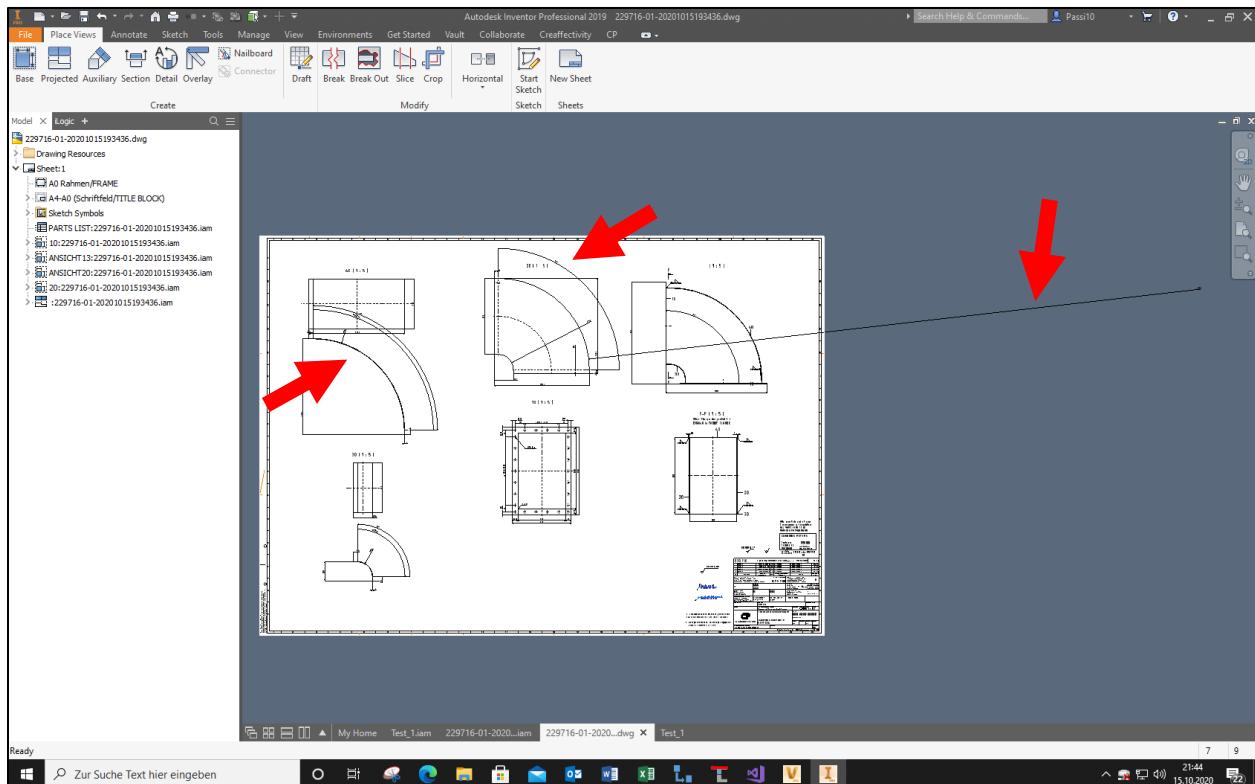


Wird ein Asset ausgewählt dann wird automatisch ein Fertigungsmodell erzeugt. Das kann immer je nach Komplexität des Fertigungsmodells einen Augenblick dauern, aber ist alles korrekt konfiguriert und parametrisiert ist dann ein 1:1 Modell entstanden.

In unserem Fall haben wir dann zu jedem Fertigungsmodell noch eine Basis Zeichnung erstellt. Diese wird automatisch mit kopiert und stellt dann das zum Asset gehörige Fertigungsmodell als 2D Zeichnung dar.

Je nach Aufwand bei der Erstellung einer Zeichnung in Inventor kann auch hier die Nacharbeit minimiert werden. In unserem Fall waren die Zeichnungen bis auf einige Maße, die noch zurecht gerückt werden müssen, oder ein paar Positionsnummern und Schweißangaben schon ziemlich gut.

Hier ist gut zu sehen was ich meine:



Wie geschrieben, je nach Aufwand, kann die Nacharbeit noch minimiert werden.

Abschluss

Ich hoffe, dass mein Beispiel ein wenig gezeigt hat, was mit ein paar Handgriffen möglich ist.

Ob nun Maschinen zusammengebaut werden oder Anlagen etc. das spielt keinerlei Rolle. Der Fantasie sind eh keine Grenzen gesetzt und iLogic, die FDU und .net lassen sich gezielt einsetzen um kleine und große Ideen umzusetzen.

Die Erfahrung hat mir leider gezeigt dass viele Anwender noch ein wenig scheu sind was den Umgang mit ihrer Software betrifft. Daher kann ich nur empfehlen, beschäftigt Euch, beschäftigen Sie sich mit ihren Werkzeugen und lernen sie kennen, was damit alles möglich ist.

Ich weiß das vielen die Zeit fehlt um sich tief einzugraben, aber je mehr Zeit gewonnen wird durch die effektive Nutzung aller Optionen, umso mehr Zeit wird gewonnen, um noch effektivere Lösungen zu finden.

Wichtig ist für mich der Hinweis an alle:

Auch die Community die es weltweit gibt und sich mit den Autodesk Produkten beschäftigt hilft gerne bei allen Ideen und Umsetzungen.

Eine simple Frage wie „Wie kann ich Vorgang XY beschleunigen“ reicht meistens schon aus.

Inventor, Vault und die FDU sind eine ideale Basis für eine effektive Umsetzung von Ideen.