

Let's Take It From The Top: iLogic Best Practices for Success

Thomas Fitzgerald

Principal Implementation Consultant





About the speaker

Thomas Fitzgerald

- Manufacturing industry experience since 1997
- CAD user since 1998, Inventor and Vault expert since 2007, iLogic and Inventor API expert since 2010
- Microsoft Certified Systems Administrator (MCSA) 2004
- Joined Autodesk in 2011
- WWII historian, UAV Pilot, US Army Veteran

What is iLogic?

Dec 8, 2008

- “iLogic enables designers to capture and embed engineering and product knowledge directly into Inventor models, paving the way for rules-driven design. iLogic creates "smart parts" that define product behaviors and can embed higher levels of design intelligence directly into an Inventor model through a user interface designed for users with little or no programming experience.”

iLogic was created to fill a need for mid ranged manufacturing CAD software, allowing designers to integrate intelligence to 3D models and drawings. Rules Based Design, when developed properly, increases consistency and quality while improving throughput and performance in engineering departments. Decisions, conditions, and criteria can be written in code and applied to designs alleviating particular repetitive tasks from designers and engineers, allowing them to focus on more important processes.

What is Design Automation and Why?

Design automation enables you to turn around proposals quickly, design and manufacture efficiently, and deliver on your promises consistently, all whilst maintaining a healthy profit. Companies offering custom products are often putting their engineers under pressure. They're squeezed to create proposal documents and drawings as fast as possible. This leads to "best guess" costings, for orders that aren't guaranteed. Engineering departments are wasted on tasks that could be automated. They have less time to re-engineer existing designs, update drawings, and carefully check every detail. Jobs are then left open to errors, rework, backlogs, and delays. All of this can affect profit margins and damage reputation.

By automating time-sensitive and often repetitive upfront activities, you will reap the **benefits**. You have more time to **innovate** and add **value** to your products, improving your position in the market. You increase **throughput** and improve **quality**, enabling you to **win** more business.

Helpful Skills to Learn

- **INVENTOR API**

Leverage the Inventor API for writing your logic. The iLogic Rule Editor has a number of valuable snippets to use, but it doesn't encompass all the capabilities needed for comprehensive Design Automation. Understanding and adopting common code writing practices and fully understanding the Inventor API architecture will allow for more effective and thorough Automation.

- **VB.NET**

iLogic uses the Visual Basic language as its foundation. Becoming familiar with common code writing practices will ensure you can understand how to construct rules that are efficient and easy to maintain not only by yourself but by others that may need to edit your code. There are many books and online resources available at little or no cost for anyone with the ambition and desire to learn writing VB.Net in a short amount of time.

10 iLogic Fundamentals for Success

1. Configuring Inventor for iLogic
2. Internal and External Rules
3. Event Triggers
4. Parameters and Properties
5. Declaring Variables, Typecasting, and Shared Variables
6. Conditional Expressions and Loops
7. Routines and Functions
8. Rule Structure
9. Error Handling and Commenting
10. Forms and User Interfaces

```
Sub Main
    Dim oDrawDoc As DrawingDocument = ThisApplication.ActiveDocument
    Dim oActiveSheet As Sheet = oDrawDoc.ActiveSheet
    Dim oDrawingView As DrawingView = oActiveSheet.DrawingViews.Item(1)
    Dim oAssemblyDoc As AssemblyDocument = oDrawingView.ReferencedDocumentDescriptor.ReferencedDocument
    Dim oTG As TransientGeometry = ThisApplication.TransientGeometry

    oDrawingView.ViewStyle = 32257

    *** Iterate through Assembly to find Parts
    Dim oOCCs As ComponentOccurrences = oAssemblyDoc.ComponentDefinition.Occurrences
    For Each oOcc As ComponentOccurrence In oOCCs
        If oOcc.DefinitionDocumentType = 12291 Then
            Call TraverseSubAssy(oActiveSheet, oDrawingView, oTG, oOcc.SubOccurrences)
        Else
            Call CreateBalloon(oActiveSheet, oDrawingView, oTG, oOcc)
        End If
    Next

    oDrawingView.ViewStyle = 32258

    iLogicVb.RunRule("Add Dimensions")
End Sub

Private Sub TraverseSubAssy(oActiveSheet As Sheet, oDrawingView As DrawingView, oTG As TransientGeometry, oOCCs As ComponentOccurrences)
    For Each oOcc As ComponentOccurrence In oOCCs
        If oOcc.DefinitionDocumentType = 12291 Then
            Call TraverseSubAssy(oActiveSheet, oDrawingView, oTG, oOcc.SubOccurrences)
        Else
            Call CreateBalloon(oActiveSheet, oDrawingView, oTG, oOcc)
        End If
    Next
End Sub

Public Function CreateBalloon(oActiveSheet As Sheet, oDrawingView As DrawingView, oTG As TransientGeometry, oOcc As ComponentOccurrence)
    Try
        Dim occParent As ComponentOccurrence = oOcc.ParentOccurrence
        Dim parentName As String = occParent.Name
        If Not Right(parentName, 1) = "1" Then
            Exit Function
        End If
    Catch
    End Try
End Function
```

Configuring Inventor for iLogic

Internal and External Rules

Parameters and Properties

Declaring Variables, Typecasting, and Shared Variables

Conditional Expressions and Loops

Routines and Functions

Rule Structure

Error Handling and Commenting

Forms and User Interfaces

Questions from you
Answers from me



Make anythingTM

Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2019 Autodesk. All rights reserved.

