Introduction to Inventor API Automation: Where Should You Start?

Nathan Bowser
Mechanical Designer
MESA, Inc.
Class summary

- Inventor software User
- Little or No programming knowledge
- Interested in learning to automate your design process using the Inventor API
Key learning objectives

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

- Recognize the importance of learning VB.NET, and know where to go to gain that knowledge
- Learn how to capitalize on the Inventor API Help to transfer VB.NET skills to the Inventor API
- Learn how to identify and utilize resources available for advancing your programming knowledge
- Learn how to debug programs directly within the iLogic interface using a debug viewer
Introduction

▪ Companies are doing more with less

▪ The need for automation has never been greater
  ▪ Reduce process time
  ▪ Eliminate opportunities for errors
Speaking the Language
Learning a programming language

- First, and likely largest, hurdle in becoming an Inventor API user
- Multiple language options available
  - VB.NET, VBA, C#, C++
- Which language should you learn?
VB.NET

- VB.NET is not the best programming language for every application
- VB.NET is the best language for new programmers who want to use the Inventor API
Why VB.NET?

- Simplicity
  - English-based Syntax (conversation like)
  - Compare the following samples:

VB.NET Sample

```vbnet
Dim i As Integer
For i = 0 To 100
    If i = 7 Then
        MessageBox.Show(i)
    End If
Next
```

C# Sample

```csharp
For (Int i = 0; i <= 100; i++)
{
    If (i == 7) {
        MessageBox.Show(i);
    }
}
```
Why VB.NET?

- Flexibility
  - iLogic is based on the VB.NET language
  - Learning VB.NET will allow you to program with both iLogic and the Inventor API
Where can I learn VB.NET

- Microsoft Virtual Academy
  - https://mva.microsoft.com/
  - Free, Online Training
  - By World-Class Experts

- “Visual Basic Fundamentals for Absolute Beginners”
  - By Bob Tabor
Beyond the Programming Language
Let’s Talk Semantics

- Common question beginners ask
  “What is the difference between programming in iLogic and programming in the API?”

- Let’s compare the two and clarify
What is the Inventor API

- Application Programming Interface
What is iLogic?

- **Inventor Add-In**
  - Included with Inventor
- Based on the VB.NET Language
  - Utilizes the Inventor API
- Additional functionality built-in
Additional iLogic Functionality

- Pre-existing snippets reduce barriers to entry
  - No need to connect to an Inventor session
  - Easily able to work with an open document
  - Quick access to parameters and iProperties
  - Simple methods for working with Excel
iLogic versus the API

- Application Programming Interface

Program: iLogic

Interface: API

Response

Request

Application: Inventor

Functions exposed to the API
Advantages of working outside of iLogic

- Visual Studio boasts greater functionality
  - IntelliSense capabilities
    - Makes programming faster and easier
  - Real time variable monitoring
    - See what values are currently contained
  - Debugging functionality
    - Examine exactly what the program is doing, line-by-line
Which one is Best for you?

- For new programmers, I recommend...
- iLogic
  - iLogic allows new users to begin building working Inventor programs right away
  - Program just what you need, when you need it
  - iLogic doesn’t have the additional overhead of Visual Studio
Transferring VB.NET Skills to Inventor
Help available from Autodesk

- Inventor Programming API Help Documentation
- Inventor API Object Model
Introduction to Using Inventor's Programming Interface

There are several resources provided to help you use Inventor’s Application Programming Interface (API). These resources are all part of Inventor's Software Development Kit (SDK). The various elements of the SDK and some additional external resources are described below.

API Help

The API Help is installed with Inventor and is accessed from the Help menu as shown below.

The help content consists of several parts:

- Introduction to the API, which is what you’re reading now.
- What’s new in this release of Inventor. This lists the changes that have been made in Inventor that may require some changes to any existing programs and lists the new objects, method, properties, and events that have been added for this release.
API Help Documentation

- Inventor API User’s Manual
  - Provides a look at Inventor ‘Behind the Curtain’
- Inventor API Reference Manual
  - Lists all Objects and Enumerators
- Sample Programs
  - Demonstrate how to implement
API Help Documentation

- **Index**
  - Lists the Objects and their Methods and Properties

- **Search**
  - Allows you to search the entirety of the Help file
API Help Documentation

- Object Name
- Description
- Methods
  - Behaviors
- Properties
- Data
- Samples
Help available from Autodesk

- Inventor Programming API Help Documentation
- Inventor API Object Model
Inventor API Object Model

oSHEET.DrawingNotes.Leadernotes.Item(1)
Additional Resources available

1. Manufacturing DevBlog
   ▪ http://adndevblog.typepad.com/manufacturing/inventor/

2. Mod the Machine
   ▪ http://modthemachine.typepad.com/

3. From the Trenches with Autodesk Inventor
   ▪ http://inventortrenches.blogspot.com/

4. Autodesk Knowledge Network
   ▪ https://knowledge.autodesk.com/

5. Autodesk Inventor Customization Forum
   ▪ http://forums.autodesk.com/...
Debugging iLogic
What is Debugging?

Debugging is the process of finding and resolving defects that prevent correct operation of computer software or a system.
Debugging iLogic

- No way to implement line-by-line debugging
- We can monitor critical paths and values
  - Using VB.NET’s Diagnostics Class

```csharp
System.Diagnostics.Trace.WriteLine("string")
```
Debugging iLogic

- Debug View
  - Monitoring tool available from Microsoft
  - [technet.microsoft.com](http://technet.microsoft.com)
## Debugging iLogic Demonstration

![DebugView on \SET3_M1 (local)](image.png)

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Debug Print</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Status' Process Initiated</td>
</tr>
<tr>
<td>12</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 0</td>
</tr>
<tr>
<td>13</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 1</td>
</tr>
<tr>
<td>14</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 2</td>
</tr>
<tr>
<td>15</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 3</td>
</tr>
<tr>
<td>16</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 4</td>
</tr>
<tr>
<td>17</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 5</td>
</tr>
<tr>
<td>18</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 6</td>
</tr>
<tr>
<td>19</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 7</td>
</tr>
<tr>
<td>20</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 8</td>
</tr>
<tr>
<td>21</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 9</td>
</tr>
<tr>
<td>22</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Info' i = 10</td>
</tr>
<tr>
<td>23</td>
<td>2:42:07 PM</td>
<td>iLogic: 'Status' Process Complete</td>
</tr>
</tbody>
</table>
Summary

- Learning to build Inventor Automations doesn’t happen overnight
  - Requires time and dedication
- With the help of these resources you can achieve success
How did I do?

- Your class feedback is critical. Fill out a class survey now.
- Use the AU mobile app or fill out a class survey online.
- Give feedback after each session.
- AU speakers will get feedback in real-time.
- Your feedback results in better classes and a better AU experience.
More Questions? Visit the AU Answer Bar

- Seek answers to all of your technical product questions by visiting the Answer Bar.

- Open daily from 8am-6pm Tuesday and Wednesday; 8am-4:30pm Thursday.

- Located outside Hall C, Level 2.

- Meet Autodesk developers, testers, & support engineers ready to help with your most challenging technical questions.