

Introduction to Autodesk Inventor API and Customization for Dummies

Autodesk has exposed Inventor's API (Application Programming Interface) for programmers to extend the functionality of Inventor to custom requirements. It gives the power to developers to make tailor made Addins / Plugins for Inventor.

Before briefing about the API and how to customize Inventor, let us understand why we should go for Customization. Consider a situation where a CAD professional has to go through some 10 steps to achieve a task in Inventor and if the task is done repetitively, it could be programmed by using its API to do the repetitive tasks on its own, thus saving a lot of time and resources. Let us consider a simple example to understand the need for customization by taking a simple example of modeling a Cube, whose volume (length * breadth * height) is a constant value and you have a requirement of all variants of it for the following conditions. (This example is just for the sake of understanding, though you seldom come across these types of requirements).

- Volume of Cube = 1000 units.
- Length and Breadth of the cube are always equal and are integers (non-decimal values)
- Minimum length of each side = 5 units.
- Maximum length of each side = 40 units.

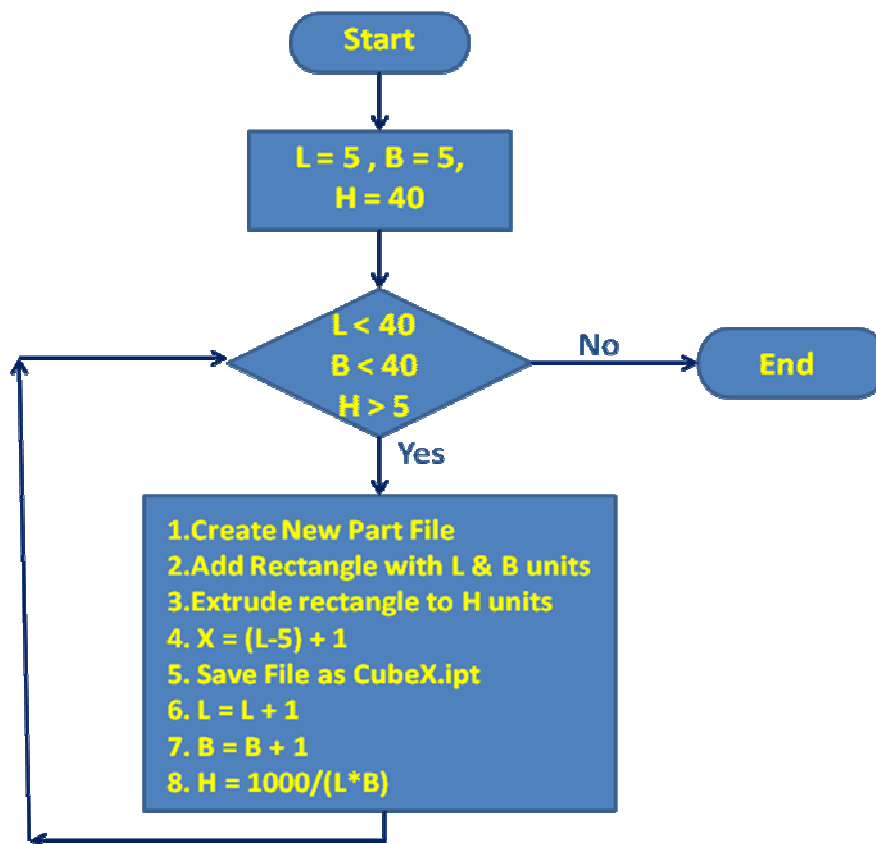
Sl No	Length	Breadth	Height	Volume
1	5	5	40	1000
2	6	6	27.77778	1000
3	7	7	20.40816	1000
4	8	8	15.625	1000
5	9	9	12.34568	1000
6	10	10	10	1000
7	11	11	8.264463	1000
8	12	12	6.944444	1000
9	13	13	5.91716	1000
10	14	14	5.102041	1000

The above table lists all the possible values of length, breadth and height of the cube which satisfy the given conditions. If a CAD professional were to do this the usual (manual) way, he/ she would most probably follow these steps:

1. Open a new part document
2. Draw a 2-Point Rectangle in Sketch Mode (with each side = 5 units)
3. Return to Part Model
4. Click on Extrude Command Button
5. Select the Rectangle, you created earlier
6. Enter the dimension of extrusion (40 units)
7. Saving the part document as Cube1.ipt

The person may take around 3-5 minutes to do the above mentioned 7 steps. He / She will now have to repeat the above steps for all the 10 different values of dimensions mentioned in the above table. In total, it would take around 30-50 minutes for the completion of all the variants of cube.

We will be able to save a lot of time and resources if we could code a program which does these repetitive tasks. The following flow chart would help you understand it.



The article does not cover any particular programming language(VBA, VB.NET, C# etc), but gives a brief understanding on the above flowchart. The languages would be covered in articles to follow.

In the above program flowchart, we start with assigning values to L,B and H variables and then check the conditions in the Decision Box. If the condition is true, we follow the steps listed in the bottom-most process and go back to Decision Box. This process (loop) continues until the Condition is returned falls (for the 11th time, when $H < 5$) and stops the program with End. Though the program looks fairly simple, it does save a lot of time. The only time spent would be to code this program and comparatively small duration of its execution.

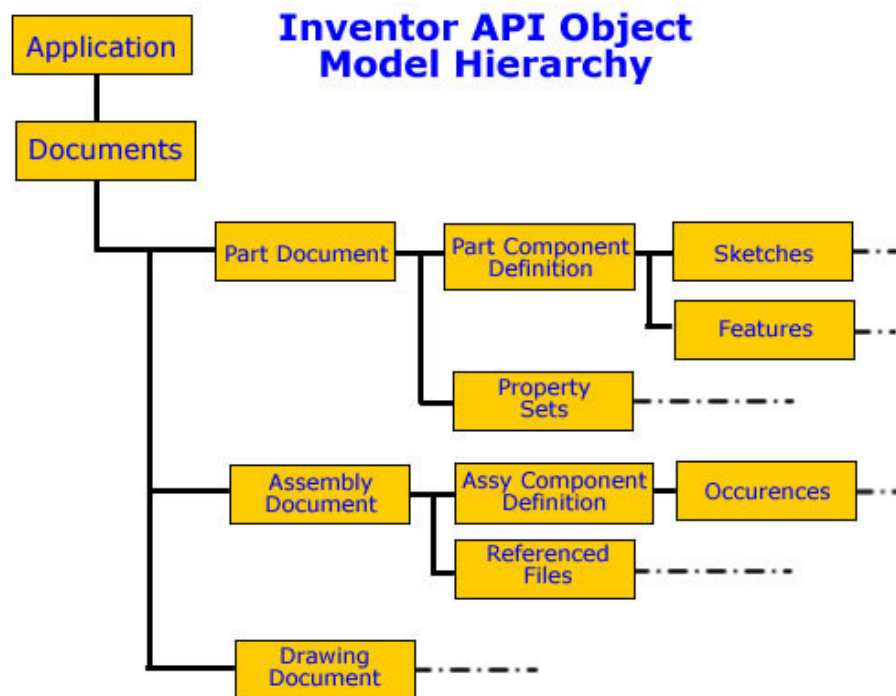
Now that you know and understand the logic of the program, we also need to make it talk to Inventor API and make it understand to execute the program. To achieve that, you would require knowledge of its API.

The above simple example holds good if there is a repetitive task at hand. The other main benefits of API are

- 1) Inventor by default, ships up with certain standard configurations. Industries which use Inventor, would have their own set of standards and configurations. By using its API, they can customize it to suit their own needs.
- 2) Many third party developers use the API to develop basic to advanced addins/ plugins to cater to different needs. Examples would be File exporters and importers into different CAD formats.

Inventor API Object Model

Inventor API is a COM (Component Object Model) based Object Oriented API. It is exposed through a set of objects and you can use valid methods and properties of these objects to create new objects and query and edit existing objects. To use its API, we should be familiar with all the various objects it supports and understand how we can gain access to a specific object. Main components of the object model are shown in the diagram below.



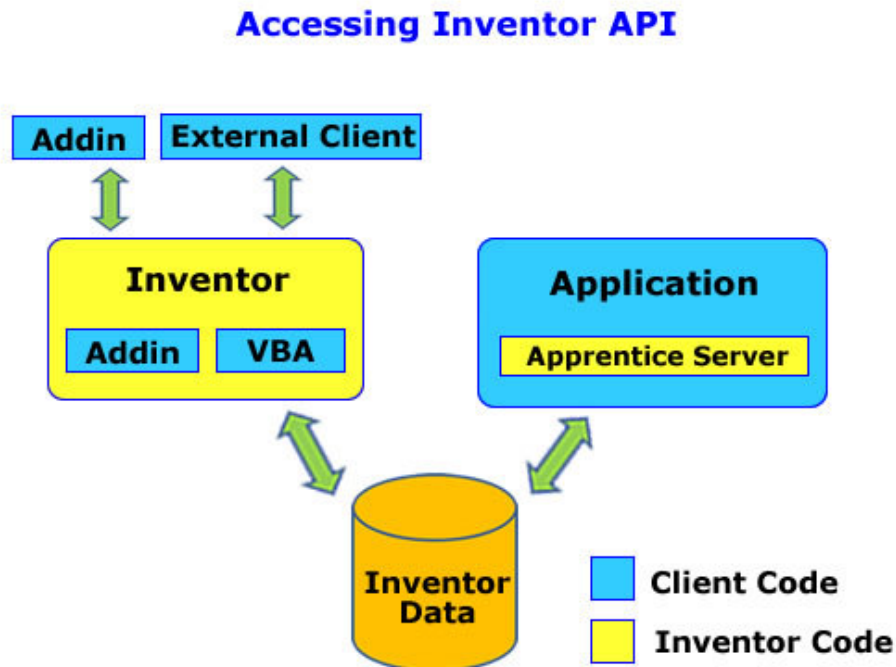
The top most object in the API Object Model is Inventor.Application object. Below it is Documents object, which is a collection of documents and is broadly classified into Part, Assembly and Drawing documents. Each of these document type have specific methods and properties which can be queried by the custom program you would write.

A simple API query would be **Inventor.Application.Documents.Count** . By doing this, we are navigating through the object model and calling the method “Count” in “Documents” object, which returns the number of documents that are opened in Inventor. We will have to understand the Object Model thoroughly to program moderate to complex addins. A detailed object model can be downloaded from

<http://www.autodesk.com/developinventor> and also browsing through Programming Help from Help Menu in Inventor.

Possible ways to Connect to API

The following diagram gives a brief introduction to the possible ways to connect to Inventor API.



Inventor data can be accessed broadly by Inventor and standalone Apprentice Server Applications. Again through Inventor, it can be either done internally (Through an Internal-Addin or VBA) or externally (through External-Addin or External Client).

VBA

Visual Basic for Applications, or VBA as it is commonly called, is a programming environment developed by Microsoft. Microsoft makes it available for companies to integrate into their applications. Inventor has an integrated VBA module, that gets shipped with it at no extra cost and is definitely the easiest way to begin Inventor Customization.

Addins

Addins are set of programs that can interact with Inventor API either as in-process with Inventor or as an external process to Inventor as a standalone program. The external process would still need Inventor process to be running for them to work. They are developed using one of the supported languages, which are Microsoft Visual C++®, VB, Delphi, and C#, VB.NET.

Apprentice Server

Apprentice servers are standalone applications which can access a somewhat restricted set of Object Model and can run even without Inventor running.

This is just the tip of iceberg. There are a lot of things you should learn to make very good programs to customize Inventor. The following resources would help you achieve it at a faster pace.

1. Dev TV Video Tutorial by Autodesk on “Introduction to Inventor Programming”. It can be viewed and downloaded from <http://www.autodesk.com/developinventor>
2. Browse through Brian Ekins’s blog on Inventor Customization at <http://modthemachine.typepad.com/> . Brian is the creator of Inventor API and has explained the basics and introduction to its API in his blog.
3. Beginners Guide to Inventor API, Autodesk University 2003. Though it may not seem to be of recent times, it has very good explanation on basics of API and Customization using VBA. It can be downloaded from http://www.smallguru.com/wordpress/wp-content/upload/AU2003_BeginnersGuideToTheAPI.zip
4. Browse Autodesk University (<http://au.autodesk.com/>), the registration for which is free and there are a lot of resources applicable to varied level of expertise in Customization.
5. Browse AUGI library (<http://www.augi.com//education/>) , it also has some suitable resource
6. Browse Inventor Customization forum (<http://discussion.autodesk.com/forum.jspa?forumID=120>) to get more insights, code examples and also problems faced by other developers and also an active Autodesk support team which answers those queries

Summary

In this article, we have discussed the necessity for Inventor Customization, brief introduction to Inventor API Object Model and different ways of connecting to it. In articles to follow, we would try to explore further on Inventor Customization.

About the Author

Rajeev Lochan is a CAD Developer and Enthusiast. At AR-CAD (<http://www.ar-cad.com>), he helps the core developer of freeCAD (free 3D CAD with Motion Simulation) in extending it further and also develop addins for various CAD software including Inventor. He also blogs about his learning experience of CAD and its customization at <http://www.smallguru.com> . He is also a member of AUGI Bangalore Local Chapter.

