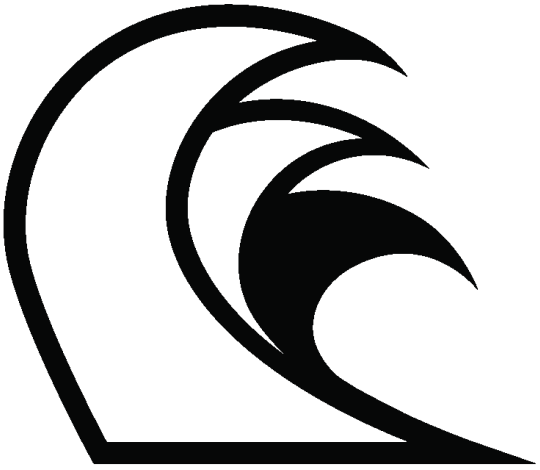
Programming in MotionBuilder || Focusing on Python
Autodesk MotionBuilder 2013

Autodesk Developer Network
May 2012
Module 1: MotionBuilder Programming Introduction



AUTODESK®
MOTIONBUILDER

Contents

Programming in MotionBuilder || Focusing on Python

Autodesk Developer Network

Module 1: MotionBuilder Programming Introduction

Agenda

- Introduction to MotionBuilder
- Why use Programming in MotionBuilder
- Python Introduction
- What distinguishes Python from the OpenReality SDK?
- How do they both fit in Autodesk MotionBuilder
- Help Documentation
- Code Samples
- Additional Learning Resources
- Let us know about Software Problems or Wish list Items

1.0 Introduction to Autodesk MotionBuilder

- Autodesk MotionBuilder is the industry-leading, real-time 3D character animation software for games, feature film, and television productions.
- It is designed to complement Autodesk Maya and Autodesk 3ds Max software, as well as other Autodesk DirectConnect applications that support the Autodesk FBX file format.
- With its core focus on interactive real-time workflows, Autodesk MotionBuilder enables creative and technical artists to take on demanding, animation-intensive projects.
- It is a professional package designed for 3D data acquisition, manipulation, and visualization, which makes it not only an animation productivity solution, but a tool that drives the iterative process of creativity.
- The software, available for Windows operating systems, natively supports the platform-independent Autodesk FBX 3D data interchange solution that allows Autodesk MotionBuilder software to integrate with applications in a production pipeline that supports FBX.
- In addition to powerful animation features and an intuitive interface, Autodesk MotionBuilder also includes dedicated tools for creating cameras, lights, shading, cartoon rendering, textures, shadows, and constraints.
- Autodesk MotionBuilder comes with two SDK toolkits one in C++ which we call Open Reality, OR SDK for short, and the other one is in the language Python, which we just refer to it as the Python SDK, or Python for short.

1.1 Why use Programming in MotionBuilder

- Automate repetitive, time-consuming tasks and extend features without leaving the Autodesk MotionBuilder environment.
- Support for the popular, easy-to-use Python scripting language allows production facilities to better integrate Autodesk MotionBuilder into their production pipeline.
- The Open Reality SDK can be used to create custom tools and features that plug directly into Autodesk MotionBuilder and extend its functionality.
- In-house developers can create project-specific functionality, which accommodates specific workflows and requirements, including custom file types.

1.2 Python Introduction

What is Python all about?

- Object Oriented
    - This means its class model supports advanced concepts such as polymorphism, operator overloading, and multiple inheritance.
    - Don't worry if you don't understand these terms you'll find they are much easier to learn with Python than with just about any other Object Oriented Programming language available.
- Open Source
    - Which means it is free; there are no restrictions on copying it, embedding it in your systems or shipping it with your products.
    - The beauty of free means tons of people are using it so there are lots and lots of forums, communities, extra tool being built, and lots of keen programmers that can help you when you get stuck.
- Mostly Interpreted
    - There is no need for an external compiler or debugger as you do not need to compile your code to run it.
- Used for Both Standalone programs and Scripting Applications
    - This multi-purpose language is great for Autodesk MotionBuilder because it allows you to use the standard Python language to do Windows type functionality and automation if you need too all in the same Autodesk MotionBuilder Python scripts.

Advantages

- Quicker Development Cycle
    - You can make changes on the fly, which means it is well suited to rapid prototyping, because you code, test and debug all within Autodesk MotionBuilder.
- It is Extremely Portable

    - Python is written in portable ANSI C, and compiles and runs on virtually every major platform in use today.
    - It is also backwards compatible.
- Large number of existing modules
    - Python is open-source which encourages users to create code and share it with others, so in this spirit there is tons of free modules and documentation available on the web.

Disadvantages

- Possibly Slower language compared to C++
    - Python is slower than its sister, Open Reality SDK, because it is not a fully compiled language such as C++ since it is an interpreted language; however speed depends largely on the complexity of the program.
- Discloser
    - There may be occasions when you'd like to release some functionality to an outside party but not disclose exactly how it was implemented, since Python is both the source code as well as what is finally executed, there is no way to separate the two.

1.3 What distinguishes Python from the OpenReality SDK?

Advantages of OpenReality

- C++ plug-ins runs much faster than Python scripts in the majority of situations.
- More class and function access than Python, for example storing and retrieving custom data in FBX files and manipulators is only available in Open Reality SDK.
- Can derive from existing classes, e.g. FBBox, FBConstraint, FBDevice, FBShader, etc.
- In some functions or lists where the object returned is of a more general class, you can cast the item to an object of a more specialized class to access functions specific to that specialized class. In Python, certain functions or lists can automatically return an object of the more specialized type, but not all have this ability.

Advantages of Python

- Minimal resources needed to start developing. No need to obtain a compiler like Visual Studio (which can cost money and time to learn using it). All you need to do is use the Python Editor that is built into Autodesk MotionBuilder.

- No need to close and restart Autodesk MotionBuilder to fix code. C++ plug-ins is automatically loaded on start-up and cannot be unloaded so that its code can be edited. As a result, if functionality is available, then it is generally better to prototype in Python, only switching to C++ when speed is essential and the code is more or less locked down.
- No need to figure out if creating a certain object of an Autodesk MotionBuilder class can be done normally or as a pointer.

1.4 How do they both fit in Autodesk MotionBuilder

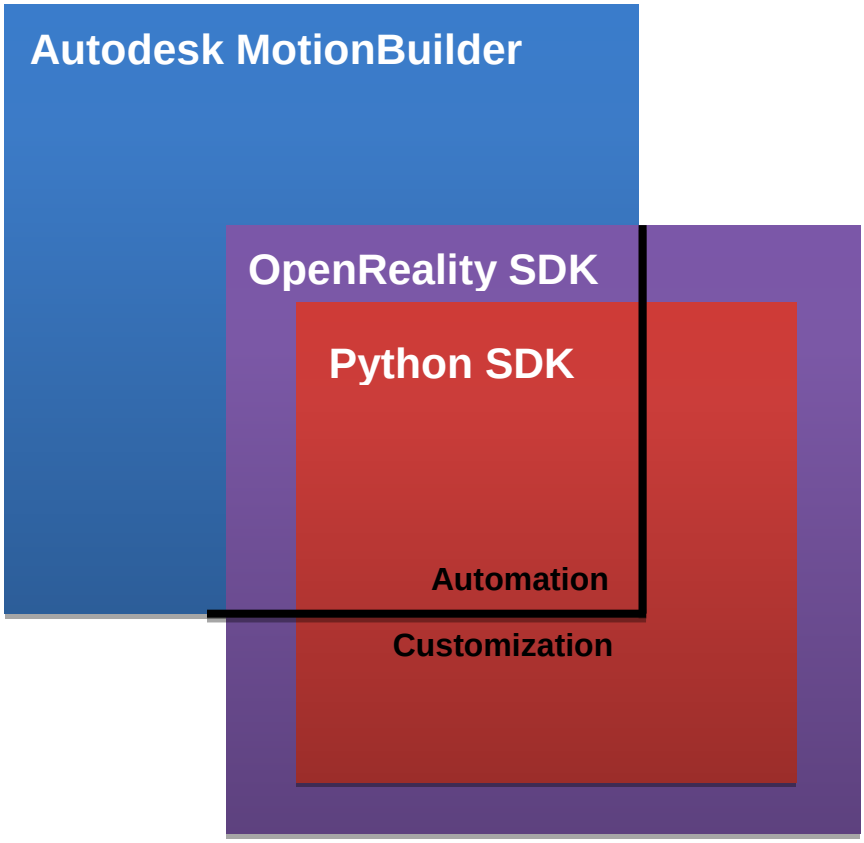


Figure 1 Displays the SDK functionality size compared to MotionBuilder

The way that Python fits into Autodesk MotionBuilder is it has a little bit smaller functionality then the OpenReality SDK, and then the OpenReality SDK has a little bit smaller functionality then what the Autodesk MotionBuilder Application can do, but both of them can extend MotionBuilder' functionalities in a way that's not possible without programming. So in other words everything Python does, the Open Reality SDK also does (Python is a wrapper to the C++ classes) but everything Open Reality SDK does Python SDK does not necessarily do (since it's has a smaller class and function set, not

by much anymore as there were huge changes made on the past few releases to Python).

There are a few key classes that are not in Python such as creating devices or manipulators that won't likely get exposed in Python because Python is not optimized enough to perform at the real time speed Autodesk MotionBuilder needs for these tools, but you never know what can happen so don't hold me to this.

One more thing to add here, is that the beauty of Autodesk MotionBuilder or the opposite of beauty depending on what side of the fence you sit on is that the functionality in Open Reality SDK and Python closely mimics the functionality available in the UI, so when you are not sure of something or what a word means you can always look it up in the User Help Documentation and see the workflow or the definitions, and 9 times out of 10 figure out how to do it in Python or OpenReality (if it is exposed).

Unlike other 3D applications that you might be familiar with, Open Reality and Python generally have the same functionality, so it really comes down to the points listed in the advantages, disadvantages and your language skill for you to choose which SDK you would like to use.

1.5 Help Documentation

Most Current Version of Documentation

We will always ship Python/OR SDK documentation with each Major Release/Service Pack/Extension; however we also want to keep you up-to-date with the latest documentation even when there is no release, so we will be post documentation on the Autodesk website.

Download from here for the latest versions of the documentation:

http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=9693656

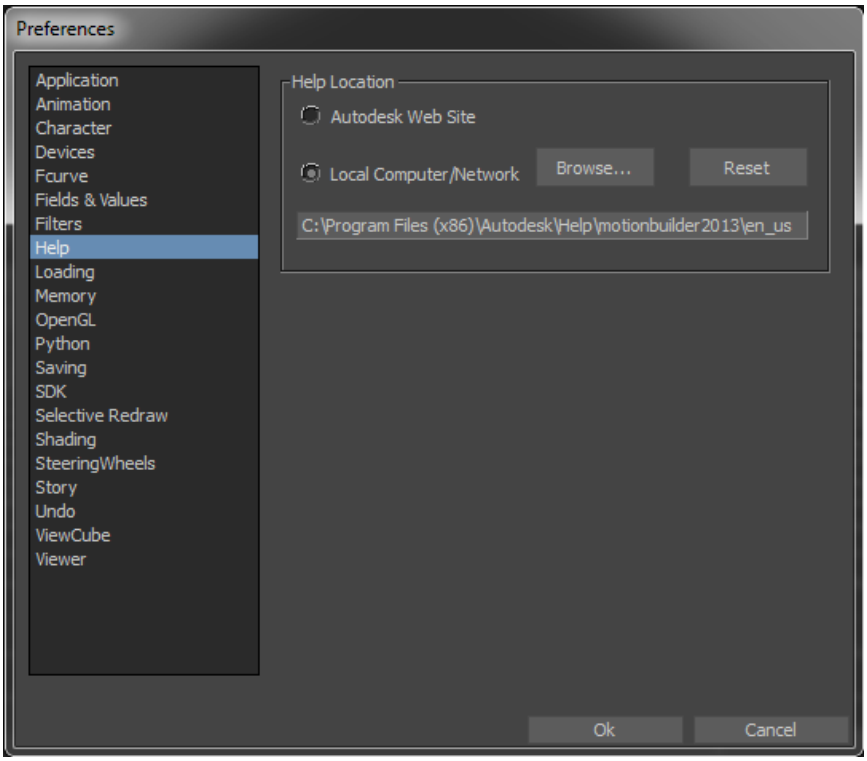
Starting MotionBuilder 2012, the MotionBuilder Help is being published to the Autodesk.com website. By default, MotionBuilder 2012 or MotionBuilder 2013 calls the Help from a web location to provide you with the latest documentation available. This shift to publishing directly to the web means we can provide regular updates and additions to content in an ongoing manner. This change also significantly reduces the footprint of locally installed data to your machine and makes the install and uninstall of MotionBuilder quicker.

However, if for some reason, you still want to install a local hardcopy of MotionBuilder help, you can do so by going through the above website and downloading the installer for MotionBuilder help and running it. The help documents will be installed at:

C:\Program Files (x86)\Autodesk\Help\motionbuilder2013\eu_us -- for 64bit machine

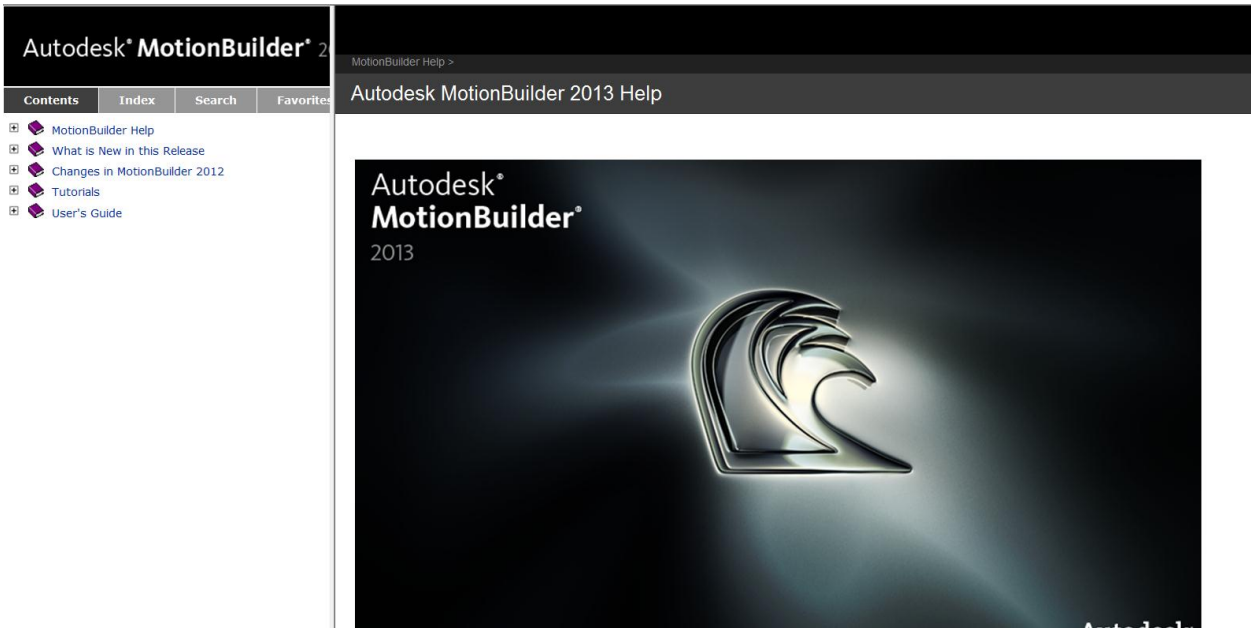
C:\Program Files\Autodesk\Help\motionbulider2013\en_us -- for 32bit machine

Then after you launch MotionBuilder, you can choose to use local hardcopy by going through menu “Settings”—“Preferences”—“Help”, and redirecting the help location to be your local hardcopy location, as shown in the screenshot.



Documentation Structure

The MotionBuilder 2013 Help Documentation consists of 5 parts, as demonstrated in the picture below:



1. MotionBuilder Help
   Provide general information about this release and how to access help documentation

2. What's New in this Release
   This document covers new feature and changes in MotionBuilder 2013

3. Changes in MotionBuilder 2012
   This document talks about new feature and changes in MotionBuilder 2012

4. Tutorials
   This chapter includes a set of nine Autodesk MotionBuilder tutorial demonstrating how to use this powerful keyframe and character animation software.

5. User's Guide
   It is designed to provide both reference and task-based information on all MotionBuilder interface elements: Window, editor, dialog box etc.

MotionBuilder 2013 SDK Documentation

Starting MotionBuilder 2012, the SDK Help Documentation is separated with MotionBuilder Help Documentation and is not included with the software installation. Instead, MotionBuilder calls the Help from a web location
To access the MotionBuilder SDK Help via the MotionBuilder software:

1. Launch MotionBuilder.

2. From the menu bar, select Help > MotionBuilder SDK Help.

The web-based MotionBuilder SDK Help launches.

To access the MotionBuilder SDK Help via the web:

Go to: http://www.autodesk.com/motionbuilder-sdkdoc-2013-enu.

The web-based MotionBuilder SDK Help launches.

To download the MotionBuilder SDK Help on your system and access the Help:

1. Go to: http://www.autodesk.com/me-sdk-docs.
2. Click the Download link next to the Autodesk MotionBuilder 2013 Open Reality SDK / Python Documentation.

   Note Extracting the Help in the MotionBuilder root directory may require administrator permission.
3. Save and extract the .zip file.
4. Double-click the folder you just created and then double-click the index.html file.

   The MotionBuilder SDK Help launches.

The MotionBuilder 2013 SDK Help encompasses the MotionBuilder SDK Programmer's Guide, the C++ Reference, and the Python Reference.

The SDK Programmer's Guide is targeted more at discussing what ways to work with the Python/OR SDK, currently this documentation focuses on some workflows; it is aimed at being written in structured English so you can learn more.

The C++ Reference Guide and Python Reference Guide are targeted at the technically savvy individuals who just want to look at the classes and see what parameters and return values they need to work within the SDK.

You will find that you will likely be working out of both Programmer's Guide and Reference Guides, since you will be accepting whatever information you can get your hands on since there is not a lot on the necessary workflows.

> ***TIP:*** Just because you are a Python Programmer don't be shy to the OR SDK documentation because you will notice there is a lot more information in it which is still relevant to Python, and Vice Versa if you are a C++ programmer using the OR SDK, don't forget to use Python as a resource as well because it is applicable to C++ mostly too.

1.6 Code Samples

The examples that come with MotionBuilder 2013 are one of the better areas to learn from, especially in this current release there was a big effort put into creating more Python examples.

The sample Python examples are located here (if you installed in the default location):

C:\Program Files\Autodesk\MotionBuilder 2013\bin\config\Scripts

The sample OR SDK examples are located here (if you installed in the default location):

C:\Program Files\Autodesk\MotionBuilder 2013\OpenRealitySDK\Samples

TIP: I am going to give you the same tip here as the documentation section, don't be shy looking for workflow information in the opposite language then you're working on because there are some samples only in OR SDK or only in Python. Just ignore the programming syntax and get an idea how to work with the classes and methods.

1.7 Additional Learning Resources

Autodesk MotionBuilder Developer Center

- Developer Center
  - *http://www.autodesk.com/developmotionbuilder*

Autodesk MotionBuilder Programming Forums

- The AREA || Python
  - *http://area.autodesk.com/forum/autodesk-motionbuilder/python/*
- The AREA || OpenReality SDK
  - *http://area.autodesk.com/forum/autodesk-motionbuilder/open-reality/*
- CG Society: Society of Digital Artists || Everything MotionBuilder
  - *http://forums.cgsociety.org/forumdisplay.php?f=85*
- David Lanier's 3D Community MotionBuilder
  - *http://dl3d.free.fr/phpBB2/viewforum.php?f=6&sid=491e7fa7c270eaa403764a750490c749*

MotionBuilder Blogs

- Stumbling Toward 'Awesomeness' by Christopher Evans
  - *http://www.chrisevans3d.com/pub_blog/?cat=12*
  - *http://www.chrisevans3d.com/pub_blog/?cat=3*
  - *http://www.chrisevans3d.com/pub_blog/?cat=5*
- Programming and Animation

    - *http://neill3d.com/en/category/motionbuilder*
- The Character Animator Toolkit for MotionBuilder
    - *http://motionbuildertraining.blogspot.com/*

MotionBuilder Python Tutorials

- Creating Interactive MotionBuilder User Interface Tools by Christopher Evans
    - *http://chrisevans3d.com/tutorials/mbui.htm*
- MotionBuilder Driven Key Importer by Jason Parks
    - *http://www.jason-parks.com/Pipeline_Tools/MotionBuilder_Scripts.html*
- MotionBuilder Tips and Tricks
    - *http://area.autodesk.com/tips/?word=&where=1&software=9&tutotips=&level=*

Web Resources for Python

- Python.org the Official Location for Python information
    - *www.python.org*
- Download Python 2.6.4 Software
    - *http://www.python.org/getit/releases/2.6.4/*

Books

- Learning Python by Mark Lutz and David Ascher
- Python Cookbook by Alex Martelli and David Ascher
- Python in a Nutshell (2nd Ed.) by Alex Martelli
- Programming Python by Mark Lutz
- Dive into Python by Mark Pilgrim
- Python.org gives you a whole list of good Python Books
    - *http://wiki.python.org/moin/IntroductoryBooks/*

1.8 Let us know about MotionBuilder Defects or Wish list Items

If you ever encounter something that doesn't work as it should in MotionBuilder or it causes the application to crash please let us know, we want to fix it, but we don't always know about every problem so you need to tell us.

Log all MotionBuilder Software defects here:

www.autodesk.com/motionbuilder-bugreport

Any items that you would like to see in MotionBuilder or are not exposed yet, log your wish list items here:

www.autodesk.com/motionbuilder-sug

TIP: Python is an exposed subset of the OR SDK and uses identical class hierarchy, and issues and limitations that apply to Python, will also apply to the ORSDK, so in other words if it's broken in the OR SDK it will be broken in Python.