FDC322552-L

Learning Lab:
Modify files using Design Automation API

Fernando Malard
Ofcdesk, LLC

Learning Objectives

- Discover the basics of Forge
- Learn how to create your first app based on Forge
- Learn how to define bundle and activity
- Learn how to use prepare files and submit workitem

Description

AutoCAD software, Inventor software, Revit software, and 3ds Max software are now in the cloud, and you can use them to perform batch operations, such as creating, modifying, and migrating files or exporting data. This class will cover the basics of how to use app bundles, activities, and work items. Bring your own laptop or use the lab. This class requires basic knowledge of .NET, and previous experience creating desktop apps for AutoCAD, Inventor, Revit, or 3ds Max.

Speaker(s)

Fernando Malard is a civil engineer who has worked with AutoCAD software and ObjectARX technology since 1996 and with Revit software since 2009. He has also been an Autodesk Developer Network member since 1997. He has worked on several AutoCAD and Revit software applications for civil engineering, architecture, interior design, and geographic information system using ObjectARX technology, C++, Microsoft .NET, JavaScript, databases, and Forge Platform engines. Malard has had extensive experience teaching AutoCAD software, Revit software, C++, MFC, Microsoft .NET, and ObjectARX technology over the last years. Today he continues to apply his skills to the design and implementation of complex industry solutions in desktop, web, and cloud environments through Forge. Malard has also worked in partnership with Autodesk, user communities, and he maintains a blog about ObjectARX technology. He holds a master’s degree in structural engineering from UFMG-Brazil and he is ofcdesk's CTO since 1999.

Introduction

Through all the information technology history we always had transition moments where challenges and paradigm shifts were essential for a relevant and substantial evolution. Recently we are experiencing a huge wave of different technologies shared across the Internet, the so-called Cloud, being broadcasted and used on several professional areas. Maybe this wave wasn't so strong and devastating due infrastructure problems, lack of technical skills and the natural consolidation of companion technologies.

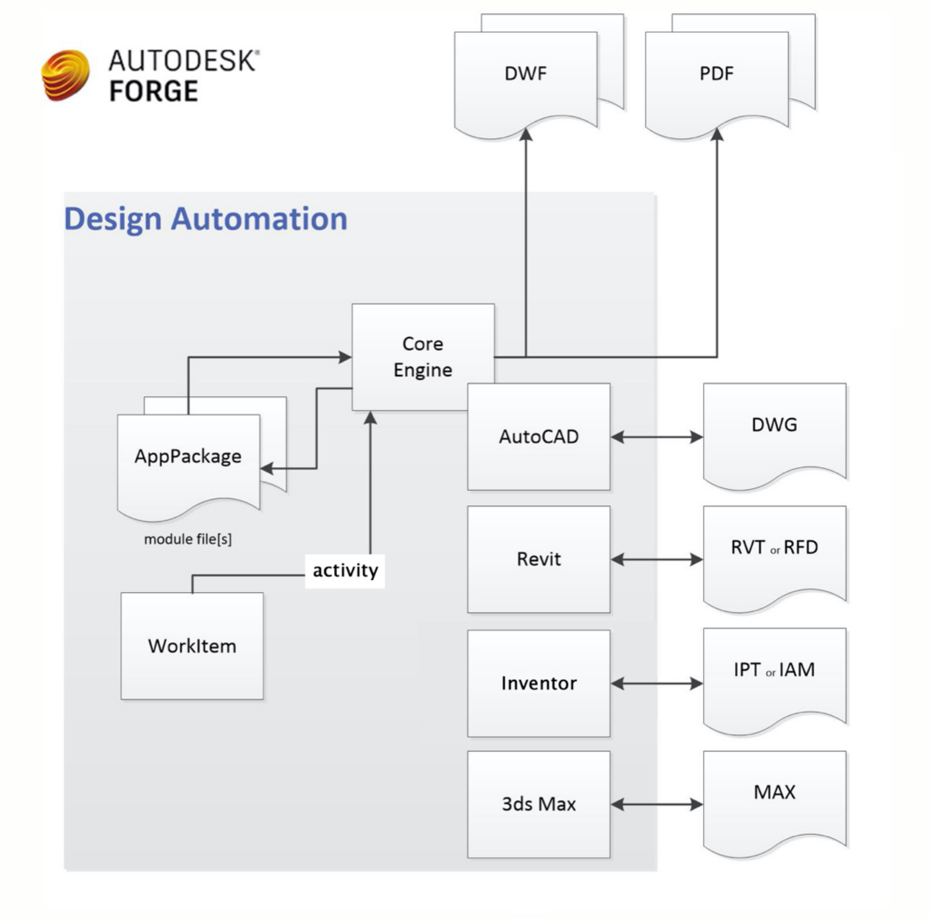
If we compare the Internet as it was years ago with the current desktop systems, it would be fairly easy determine that most of that software would never be able to be run and be used entirely in the web. The reasons were, among several, low data transfer speed, poor page processing speed, excessive information transfer, vulnerability, limited and expensive storage space, no mobile solutions or weak signal strength, remote blind spots and many others. Nowadays the reality is completely different, the Internet has evolved itself, but we also had a substantial evolvement of the infrastructure, desktops and mobile devices.

It was just a matter of time that traditional desktop solutions start to be available through Internet and hosted on robust servers with theoretical infinite storage, all this ubiquitously accessible. Facing this tech wave, the Cloud became an attractive place for existing businesses expansion, new businesses and user base expansion for companies traditionally focused on desktop solutions.

Autodesk couldn't play a different game. Traditionally focused on desktop solutions, Autodesk is creating Internet solutions for a while. The Forge platform emerged to respond to all mobility, portability, flexibility and expansion demands matching most of desktop product features. The Forge platform started with several services that can be consumed via web having the Internet browser as the only requirement. From the technology point of view, this platform brings several challenges but, in other hand, provides a powerful and revolutionary opportunity for existing companies regardless their size and revenue. In fact, the Cloud levels companies with different revenues and sizes while offering products and services to end users. During the following pages we will present the technology behind these platforms, challenges, impact and all benefits for migrating existing products and services or even create new ones on the Cloud.

Design Automation

The Design Automation API provides the ability to use the core API's of your favorite CAD engines, in the cloud, leveraging the scale of the Forge Platform to run automated jobs. These jobs could be highly repetitive, frequent or could be larger problems that need large-scale processing power. With the Design Automation API, you can offload that processing to the Forge Platform, which can process those jobs at a much greater scale and efficiency. Currently, the Design Automation API supports AutoCAD, 3dsMax, Inventor and Revit, with plans in the works to expand to other CAD engines.



Design Automation Engines

Design Automation common Terms

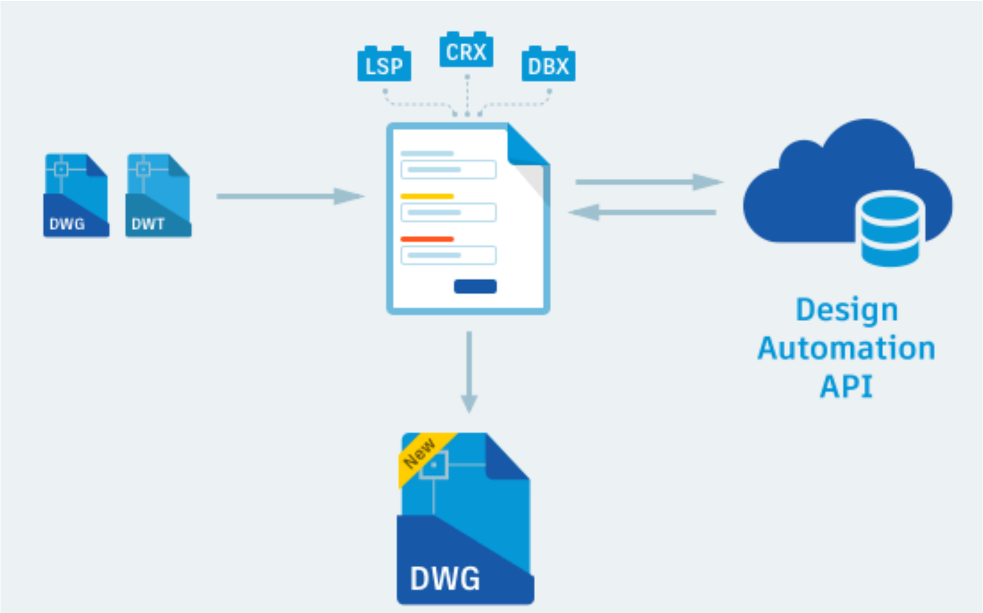
Term	Definition
Activity	The specification of an action that can be executed using a specified engine. For example, using AutoCAD to plot a supplied DWG to a PDF file. Another example: using AutoCAD to update a supplied DWG file to use a different set of CAD standards. An Activity specifies the number of input and output files, and the AppBundle and entry-point to use.
AppBundle	A module referenced by an Activity in order to perform specific functions. Typically, this is a DLL or some other form of custom code. For example, using the AutoCAD engine, this might be a custom AutoLISP routine that extracts Xdata attached to objects in a drawing or a script file that plots a DWG to a PDF file.
WorkItem	A job that executes a specified Activity, using specified input files and generating appropriate output files. The relationship between an Activity and WorkItem can be thought of as a “function definition” and “function call”, respectively. The Activity specifies the AppBundle(s) to use, which in turn specify the Engine to use. *Note: Once a WorkItem is created, it cannot be modified.*
Engine	The actual processing engine that runs the WorkItem job and processes the Activity. The following engine types are available: AutoCAD, 3dsMax, Inventor and Revit. There may be multiple versions of each engine.

Design Automation API for AutoCAD

Run scripts on AutoCAD DWG files

Imagine you have thousands of DWG files stored in the cloud, and you need them all converted to PDF files. Ordinarily, you would have to download all the files, run a script on them in the AutoCAD desktop software, and then potentially upload them all back to the cloud. Your efficiency would be bottlenecked by the processing power of your computer and your network bandwidth, and you would have to instrument logging and retry logic in your code to ensure that the entire job completed. Using Design Automation API for AutoCAD, you can offload all that processing to the Forge Platform, which can process those scripts at a much greater scale and efficiency. Current functionalities for DWG files include:

- creating new DWG files
- querying for information in existing DWG files
- purging drawings and saving them to other DWF file formats
- plotting DWG files to DWF and PDF
- translating text from one language to another



Design Automation – AutoCAD engine

Design Automation API for 3dsMax

Run Maxscript/Python and 3ds Max plug-ins with 3ds Max Batch

3ds Max offers rich capabilities for 3D data modeling, animation, rendering and simulation. Whether you are working with 3ds Max files or any other format, the 3ds Max engine can be used to load/import data from wide range of sources. Through Maxscript/Python and its extensive SDK, data can easily be processed for optimization, rendering and publishing. Design Automation API for 3ds Max provides access to the 3ds Max API running in 3ds Max Batch mode:

- Import and export data from any of the plug-ins available with 3ds Max
- Run scripts and modify the file as needed.
- Upload and use your own plug-ins C++ plug-ins and implement logic that connect to all parts of the 3ds Max architecture.
- Perform renderings and simulation and other compute intensive operations.

Design Automation API for Inventor

Run addins and iLogic with IAM/IPT/IDW files

Imagine you have customers continuously submitting orders for custom configurations of your product. Using Design Automation for Inventor plus your own custom code and iLogic rules, you can set up an automatic responder to receive these orders and prepare initial documents and models for engineers to fine-tune or validate. The automatic handler can modify parameters in part files and assembly files and update relevant drawing files, doing all the preliminary processing in the cloud, and delivering initial documents to engineers' desktops. Engineers can spend their time on the creative / unique / difficult aspects of each job, rather than wasting time on repetitive document-copying. Design Automation API for Inventor provides access to the full Inventor API ("server" only), including functionality such as:

- Modify parameters and features in IPT (part) files
- Configure components in IAM (assembly) files
- Update and fine-tune IDW/DWG (drawing) files
- Use iLogic rules or custom code
- Operate on common, static files, or per-job files
- Provide results in any Inventor-supported formats

Design Automation API for Revit

Run Revit addins or web applications to process or create RVT/RTE/RFA files

Design Automation API for Revit is Revit's engine running in the cloud as a Forge service. Build cloud-native apps and services that create, extract, and modify Revit data. By providing access to the full Revit DB API, without the requirement of installing Revit on the desktop, the following workflows are now possible anywhere:

- Create custom Revit family content
- Automate model creation
- Explore and analyze model data
- Extract and produce automated reports
- Modify existing models to maintain company standards
- Automatically create documentation

Typical Workflows

Using a Pre-Existing Activity

Use this workflow to execute a previously created Activity, supplying one or more documents as inputs. The Activity will already have had the necessary AppBundles associated with it. The Activity defines the expected input and output files.
This workflow applies to both your own Activities and Activities that are shared with you.

1. Upload all files to be processed and referenced. The files can be stored in Forge Data Management or in any other cloud storage provider.
2. Create WorkItem: Use the POST workitems endpoint to create a WorkItem that uses the pre-defined Activity. You specify the URLs of the input files and output files. The WorkItem downloads the input files and begins the execution of the custom programs or scripts defined in the AppBundles associated with the Activity passed to the WorkItem. When complete, it uploads the output files to the specified URLs.
3. Wait for completion: Poll GET workitems/:id to check the current status of the WorkItem. It might be “in progress” or “complete”. (There is also a non-polling mechanism.)
4. Get results: When the WorkItem is complete, the output file(s) are available at the specified cloud storage provider. A cloud service can utilize them from there, or you can download them.

Using a Custom Activity

This workflow is used when you need to create a new Activity and associate one or more AppBundles that define the custom programs or scripts to be executed on one or more documents.

1. Optionally, upload any custom programs or scripts to be assigned to an AppBundle and associated with a custom Activity to a specified cloud storage provider.
2. Use POST appbundles to create a new AppBundle.
3. Use POST activities to create a custom Activity.

(following steps are the same as for ‘Using a Pre-Existing Activity’)

4. Create WorkItem: Use the POST workitems endpoint to create a WorkItem that uses the pre-defined Activity. You specify the URLs of the input files and output files. The WorkItem downloads the input files and begins the execution of the custom programs or scripts defined in the AppBundles associated with the Activity passed to the WorkItem. When complete, it uploads the output files to the specified URLs.
5. Wait for completion: Poll GET workitems/:id to check the current status of the WorkItem. It might be “in progress” or “complete”. (There is also a non-polling mechanism.)
6. Get results: When the WorkItem is complete, the output file(s) are available at the specified cloud storage provider. A cloud service can utilize them from there, or you can download them.

Rate Limits and Quotas

Rate limits determine how frequently you can call an API. Rate limits are currently placed at a generous level and should not impact you negatively under normal circumstances. However, you may be impacted if your app generates an unusually high volume of API calls. When an app exceeds the rate limit for a given endpoint, the API returns a HTTP 429 “Too many requests” response code. Sometimes, the response may contain details of how long to wait before making a new request. We strongly recommend that you retry only after the waiting period has lapsed. If the next request also results in a HTTP 429 response code, retry after increasing the waiting period.

Design Automation Limits and Quotas

Quota	Applies to	AutoCAD	Inventor	Revit	3ds Max	Units
Maximum number of downloads (inputs) per WorkItem	Activity, WorkItem	10	10	200	50	
Maximum number of uploads (outputs) per WorkItem	Activity, WorkItem	10	10	200	50	
Maximum total size of all downloads (inputs)	WorkItem	100	200	2000	5000	MB
Maximum total size of all uploads (outputs)	WorkItem	100	250	2000	5000	MB
Default duration of processing Note: Includes download and upload time	WorkItem	100	100	10800	3600	seconds
Maximum duration of processing when specifying limitProcessingTimeSec Note: Includes download and upload time	WorkItem	400	1200	43200	36000	seconds
Maximum uncompressed size	AppBundle	500	500	2000	5000	MB
Maximum number of versions, in total (for all AppBundles, and for all Activities, separately)	AppBundle, Activity	100	100	100	100	
Maximum number of aliases, in total (for all AppBundles, and for all Activities, separately)	AppBundle, Activity	100	100	100	100	
Max permitted size of an AppBundle upload	AppBundle	100	100	100	100	MB
Max permitted size for AppBundle/Activity/WorkItem json payload	AppBundle, Activity, WorkItem	16	16	16	16	KB
Maximum download time for a WorkItem	WorkItem	300	300	300	300	seconds
Maximum upload time for a WorkItem	WorkItem	180	300	300	180	seconds

Some limits are common to all Engines:

- WorkItem processing time cannot exceed one minute without a "heartbeat" to indicate that it is working correctly and not in an infinite loop. The simplest "heartbeat" is to send output to the "standard output" (or "trace output") stream from your code. However, if you invoke a single operation that is expected to take more than one minute (e.g., opening a large file), you can send the output asynchronously.
- The rate limit for Forge Design Automation as a whole is 100 calls per second.

Limiting monthly consumption

By default, Forge does not limit the time a Forge App takes to runs automations. If necessary, you can use the ServiceLimit API endpoint to set the *limitMonthlyProcessingTimeInHours* attribute on *frontendLimits* to limit the amount of time the app is allowed to consume each month. If the app reaches the consumption limit, the system rejects subsequent WokItems for the remainder of the calendar month.

AppBundle and Activity Aliases

In order to be able to use a version of an AppBundle or an Activity, you have to specify an Alias for it. You can view an Alias as a user-friendly name for a specific AppBundle and/or Activity version.There is no need for a 1-to-1 mapping between versions and Aliases. You can have many versions of an AppBundle and/or Activity and only some of these versions can have an Alias assigned to them.

Also, you can re-assign an existing alias to a different version at any time. This gives you the flexibility to swap, for example, a version of an activity used within your WorkItems without the need to update each and every WorkItem. An example of the use of Aliases is to have a "Beta" and a "Production" Alias for an AppBundle/Activity. The "Production" Alias is assigned to your stable version of an AppBundle/Activity and the "Beta" Alias is assigned to a newer version of the same AppBundle/Activity. When the "Beta" version becomes stable, you assign the "Production" Alias to this newer version of the AppBundle/Activity.

$LATEST alias

For every AppBundle and Activity, the system always creates a special Alias named "$LATEST". The main purpose of this Alias is for listing AppBundles and Activities. It can also be used in

- [GET] /appbundles/:id/aliases/$LATEST
- [GET] /activities/:id/aliases/$LATEST

calls. Note however, that it must not be used in other calls where an Alias (as part of fully qualified id, see later) is expected. You must use the real Alias that you created for the AppBundle or the Activity in those calls.

Callbacks

Design Automation allows the user to define an Activity and a WorkItem in a way that makes it possible to gain a better control over the execution flow of the WorkItem processing. This is done using a callback mechanism for three different types of events.
The callback events are:

OnDemand callback

OnDemand callback allows the user to provide additional input data to the WorkItem during the WorkItem execution, when needed. Using OnDemand callback requires it to be specified in both the Activity and the WorkItem. Also, your AppBundle has to support it: to invoke the onDemand callback, it has to write a specially formatted string to the output and it also has to handle the response in a specific way.

Note: Design Automation for AutoCAD supports also OnDemand callbacks for WorkItem's output.

OnProgress callback

OnProgress callback allows the user to check for the status of WorkItem execution on a periodic base. When defined, Design Automation calls the OnProgress callback approximatelly every 30 seconds. The callback body includes the id of the WorkItem for which it has been sent. Note that Design Automation starts calling the OnProgress callback only after the WorkItem is taken from the SQS queue and the actual processing of it is started. There is also the possibility to invoke OnProgress callback directly from user AppBundle's code by writing a specially formatted string to trace output.

The user can use the callback not only to monitor the status of the execution, but also to cancel the job if needed. To do so, just send an HTTP 205 Reset Content as a response to the received OnProgress callback. User should not use DELETE workitems/:id to cancel a WorkItem that uses OnProgress callback once this WorkItem is picked from SQS queue for execution.

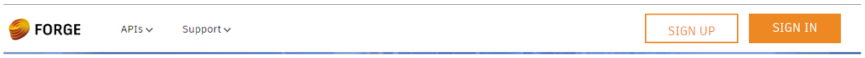
OnComplete callback

OnComplete callback lets the user know when the WorkItem execution has ended without the need to actively check for the WorkItem status periodically. The callback body includes the id of the WorkItem for which it has been sent, together with the WorkItem status and the url for the WorkItem report. Also, time and other data are included in the stats section, same as in the response to GET workitems/:idcall.

TUTORIAL

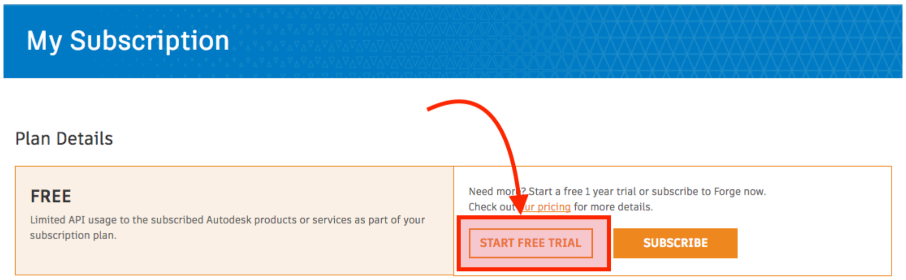
Creating your Forge account

Go to **Forge Developer Portal** (https://developer.autodesk.com/), click the “SIGN UP” button to create an account or "SIGN IN" to use an existing account. If you create a new account, be sure to click the link in the verification email that will be sent to you.



Activate subscription

Before using any of the paid APIs, like **Model Derivative**, you need to activate your trial. On the top-right, you'll see your name. Click to expand the menu and go to **My Subscription**. On the page that opens, click on **START FREE TRIAL**. That's it.



Limited API usage to the subscribed Autodesk products or services as part of your subscription plan.

Need more? Start a free 1 year trial or subscribe to Forge now.
Check out our pricing for more details.

Create an app

On the top-right, you'll see your name. Click to expand the menu and go to **My Apps**. Click the “CREATE APP” button. Select APIs you are going to use (you can safely select all for now). Enter your application name and description, then enter a callback URL:

http://localhost:3000/api/forge/callback/oauth
(this tutorial will not use this callback, but that's the URL used on other Autodesk Forge samples)

Once you set up an application, you will see a Client ID and Client Secret in your newly created app page. You will need these in all other OAuth flows and, by extension, to complete every other tutorial on this site!

DO NOT share your Client Secret, this should be kept confidential.

Modify your models

This tutorial will guide you through creating a webapp that can upload an input file, change its width and heightparameter and save the output file. The UI will let you define AppBundle & Activity (as an initial configuration) and execute Workitems, with an output result.

Both input and output files are saved in OSS Buckets, you can use "View Models" tutorial to view them.

To modify your models, you need the following steps:

1. **Prepare a plugin**
2. **Define an activity**
3. **Execute workitem**

Step by step instructions are available here:

https://learnforge.autodesk.io/#/tutorials/modifymodels