



AUTODESK UNIVERSITY 2015

GEN9663

From AutoCAD I/O to View and Data

Fernando Malard
ofcdesk, LLC

Learning Objectives

- Learn how to use AutoCAD I/O API
- Learn how to use View & Data API
- Learn how to use both APIs together
- Learn how to empower your cloud solutions

Description

This class will explore the brand new Autodesk, Inc., web APIs. Attendees will explore AutoCAD I/O API and View and Data capabilities, learning how to use these technologies, but also how to use both together. We will explore the potential of these APIs to empower your cloud solutions by enriching them with visualization features, asynchronous processing, and zero-client products.

Your AU Experts

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 applications for civil engineering, architecture, interior design, and geographic information system (GIS) using ObjectARX technology, C++, Microsoft .NET, JavaScript, Cocoa, and databases. Fernando has had extensive experience teaching AutoCAD, Revit, C++, Microsoft Foundation Class, 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. Fernando has also worked with Autodesk User Group International (AUGI) communities, and he maintains a blog about ObjectARX technology. He holds a master's degree in structural engineering from the Federal University of Minas Gerais, Brazil.

Learning Objectives	1
Description	1
Your AU Experts	1
Introduction	3
Background technologies	3
Autodesk Developer Portal.....	5
AutoCAD I/O.....	5
Features and usage	6
Abstractions	6
View & Data	8
Features and usage	8
Abstractions and concepts	8
Storage Services	10
Using AutoCAD I/O and View & Data together.....	10
Conclusion.....	11



Introduction

During years Autodesk focused on desktop applications in several areas like AEC, MEP and Media & Entertainment. With the increasingly fast evolvement of web technologies there isn't any reason to not jump in and provide rich yet simple solutions to the web by exposing portions of Autodesk core technologies through APIs.

We have now several web services available from Autodesk. Some of them are indeed in beta stage but other services are being actively used by several users around the world. In the recent past most of users won't get too excited about this because the well known Internet instability, coverage and high costs were an important obstacle to these technologies. Once those obstacles were demolished by the rapid evolvement of Internet coupled with the amazingly quick spread of mobile devices with mobile connections the web became a stable and reliable channel to be used professionally.

It is hard to provide the same desktop user experience via web but if we reengineer some aspects of each technology considering the new user experience, non abundant storage and speed we can certainly obtain a reasonably good result. In other hand, while in desktop environments we are limited by its local resources and computing power. The web is a different story because in theory we have unlimited computing power. In a local environment, if you want to double your computing power you would need to acquire new hardware equipment, expand your infrastructure and probably hire more staff. In contrast, the web provides several cloud solutions where by a click of a button you can magically duplicate your computing power. The ownership cost is also a great advantage because you are dealing with companies created and specialized on handling this instead of a local IT department that relies on their abilities to respond these demands in a non 24-hour fashion.

Two of the current Autodesk web technologies stand out at this time. They can really play important roles in your current and future web solutions. By understanding their potential, you can strategically fit them into your cloud architecture to complete the solution solving the areas you are not specialized on. Both APIs rely on some background knowledge and the use of their core technologies which would leverage their potential and results.

This class will focus on **AutoCAD I/O** and **View&Data** Autodesk APIs. They play different roles on any solutions you would implement being AutoCAD I/O responsible to deal with DWG files through a command-prompt like version of AutoCAD while View&Data, through WebGL, can work as a browser based, zero-client viewer. We will introduce both APIs and then explain how they can work together to empower your Cloud based solutions.

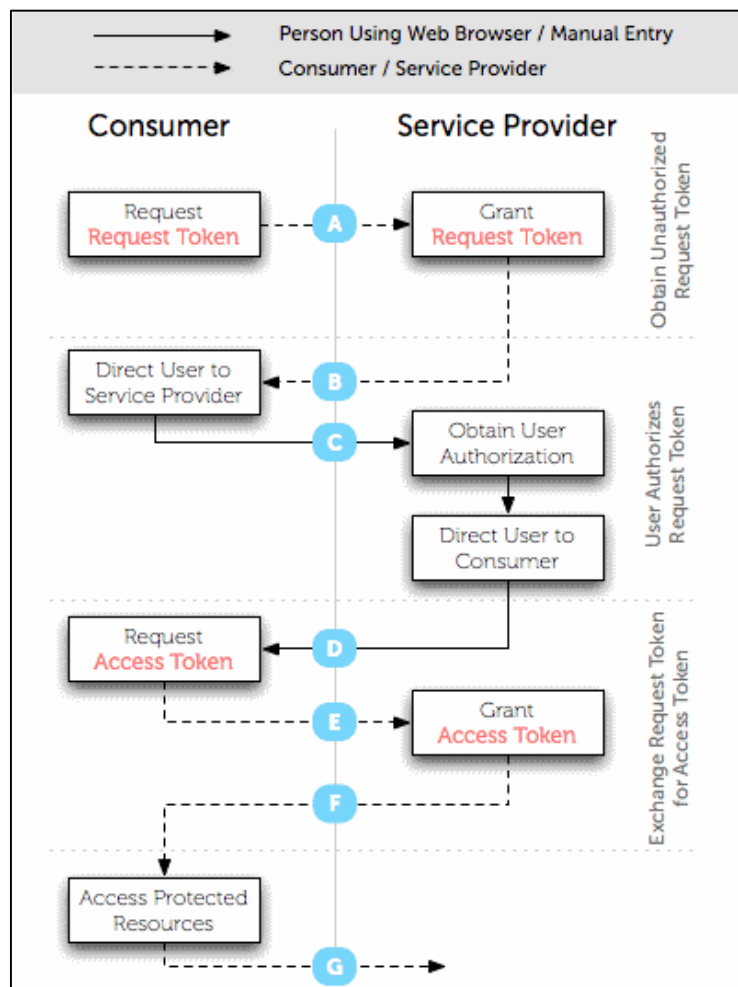
Background technologies

Both APIs have their particular characteristics and capabilities. As web services they share lots of common concepts and mechanisms to be created and consumed. Essentially they work based on some well know and widely used web technologies such as:

- **REST (Representational State Transfer)**: Is a software architectural style, based on resources, often used in Web Services. It is simple and used across the web for cloud solutions through web services calls like GET, PUT, POST and DELETE. It uses URIs as resource names and also work with HTTP responses.



- **RESTful:** Is a Web Service that implements REST architecture.
- **OData (Open Data Protocol):** It is a standard the defines the best practices for building and consuming RESTful APIs. It simplifies the business logic while building RESTful APIs allowing you to not worry about details of the RESTful procedures and protocols to consume the Web Services.
- **JavaScript:** It is a weakly typed language so variables are associated with the values they receive and not the types. It does support most of C language syntax (if, while, switch, do while, etc.) so developers fluent in C/C++ will better code in JavaScript. It is an object based language and those objects contain properties that can be accessed at runtime.
- **OAuth (Open Standard for Authorization):** It provides web client applications a secure delegated access to resources on behalf of a resource owner. It is based on access tokens issued to third-party clients by an authorization server with the approval of the resource owner.



Autodesk Developer Portal

The very first thing you will need to do is to Sign Up to Autodesk Developer Portal:

<http://developer.autodesk.com>

The image shows two parts of the Autodesk Developer Portal interface. On the left is the 'Sign Up' form, and on the right is the 'About this app' page for a newly created application.

Sign Up Form:

- Autodesk logo and 'Sign Up' heading.
- Input fields for: Name (Fernando), Surname (Malard), Email (au2015@au2015.com), Password (masked with dots), and Confirm Password (masked with dots).
- Checkbox: ☐ I would like to receive email communications from Autodesk, including information about new products and special promotions.
- Blue 'Sign Up' button.
- Text below button: By clicking on Sign Up, you agree to the Autodesk Web Service API Terms of Service and the Autodesk Privacy Statement.
- Footer: © Copyright 2015 Autodesk, Inc. All rights reserved.

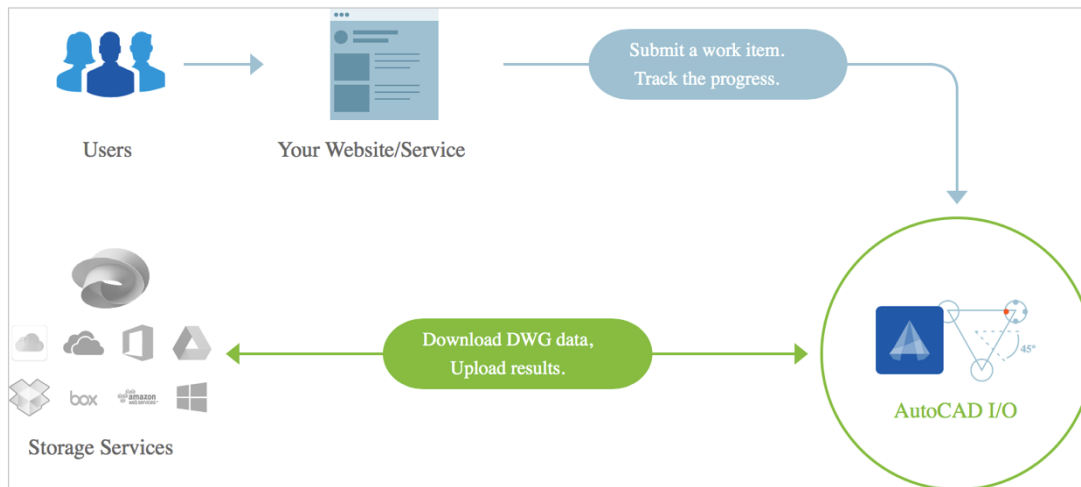
About this app (Created 12 Oct 2015):

- Icons for 'AutoCAD® I/O API' and 'View and Data API'.
- Consumer Key:** Yb5LS37ZYue5DDBVf0LSUCGomOLnlwP
- Consumer Secret:** VPX13jHfbae9CII. Links: [Hide](#), [Regenerate](#)
- App Name:** AU2015
- Description:** Autodesk University 2015 - Las Vegas
- CallBack URL:** <http://au2015.callback.mywebsite.com>
- Your Website URL:** <http://au2015.mywebsite.com>

Once you sign up, create a new application and select AutoCAD I/O and View & Data API. You will receive a **Consumer Key** and a **Consumer Secret**. These are your Developer Credentials that you need to keep private. When accessing these web services your credentials are going to be required and you need to provide them hidden from any user consuming your solution. Don't store this data into any visible file to the client like inside JavaScript code or any other HTML visible storage. You can create as many applications you want with one of these two services alone or both together as we are doing in this class.

AutoCAD I/O

The first API we will present and explore here is a service that allows you to process DWG files on the Cloud by accessing a prompt version of AutoCAD. It requires you to provide a script to run the process. The input file can be a web shared resource as the result files can be also be downloaded. You can have your own applications working together with the AutoCAD service thus enriching your processing potential and fine tuning the huge AutoCAD processing potential to your need. As a cloud based processing you can expand your capabilities by using the parallel processing native to Autodesk web services.



You cannot access any UI capability once AutoCAD I/O runs AutoCAD without any visible interface. In fact, it is almost like running AutoCAD in a prompt command. Exactly due that the applications you submit to work together with AutoCAD I/O must comply with **AutoCAD CRX and DBX applications** format and packaged in autoloader format. Currently it must run on both AutoCAD 2015 and 2016.

Features and usage

AutoCAD I/O can potentially do almost anything you want with your DWG files as long you consume a native AutoCAD feature or your own application features. Among several usages, here a some you can initially consider:

- Perform a variety of operations on DWG content.
- Create new DWG files.
- Query any information from an existing DWG file.
- Purge drawings, change drawings and save them to other formats.
- Plot drawings to DWF and PDF.
- Simplify the DWG manipulation process for ordinary users.
- Expand your processing capabilities by running several DWG files at once.
- Provide simple services running AutoCAD behind the curtains.

Abstractions

The AutoCAD I/O API has basically 3 important abstractions. The first two, AppPackage and Activity are some kind of pre-configurations setup before you actually consume the service through the WorkItem. These abstractions are better described in details here:

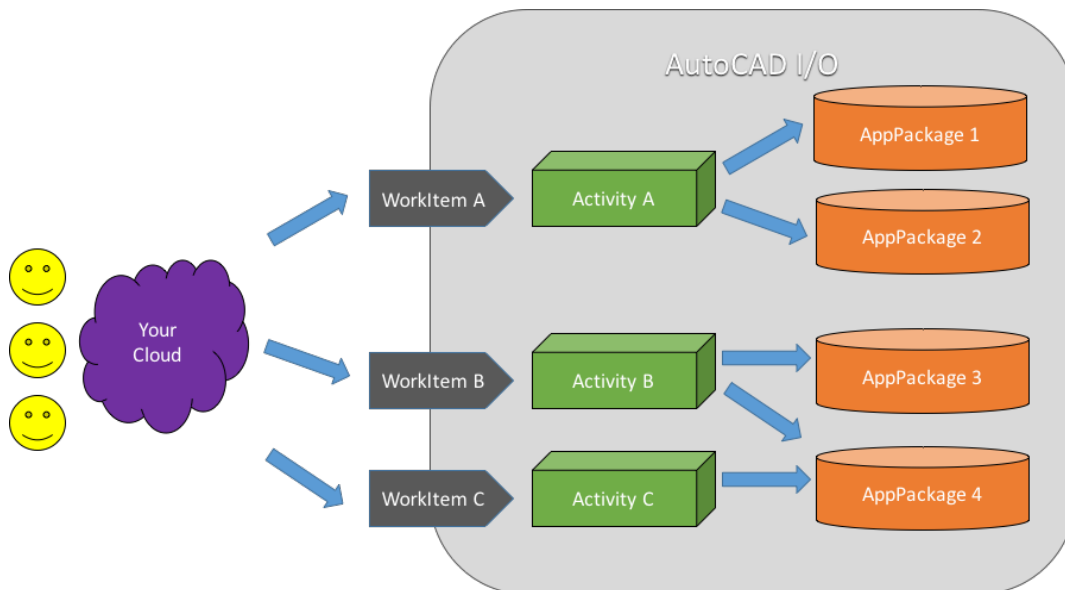
AppPackage: It is basically the application extensions you will run along your application. It can be composed by **DBX, CRX, LSP** or **DLL** modules, written in **AutoLISP, C++** or **.NET**, all compatible with **AcCoreConsole.exe** (the AutoCAD prompt application). This package can implement several commands



that will extend the native AutoCAD core capabilities and will be available to be called by your client web service calls. The application must be compliant with Autoloader format and packaged as a bundle. They also need to be compatible with **AutoCAD 2015 and 2016**. Each AppPackage will have its own ID and version so you can better control its usage across several versions of your service calls.

Activity: It is the **pre-configured action** you want to perform. This action is also identified by an ID, it can refer to one or more AppPackages and it will have a **script** (AutoCAD Script) which will drive the process. You can create several activities bound to your AppPackage each one calling a specific set of native AutoCAD commands or your own defined commands. If your Activity only calls AutoCAD native commands, you don't need to refer to any AppPackages. Each Activity will also require **input and output named parameters** which will need to be provided when you consume this activity. You can create activities in a Private or Shared fashion.

WorkItem: Finally, the WorkItem, represents each **execution** of your Activities (actions). The WorkItem will call the Activity providing its **input and output parameters** as defined when you created the Activity. Each WorkItem process can be tracked by requiring its **status** or through a callback URL. The WorkItem can point only to a single Activity through its ID. The WorkItem is the code which will be strictly bound to your application or website where the user will direct or indirectly provide the input/output arguments and watch the process execution.



It is important to understand the relationship among these abstractions will be totally dependent on the way you plan to provide your solution. You can create several application packages referred by several activities. The users will consume your services by submitting work items using high level interface like a website or application. Behind the scenes, the Autodesk servers will process these calls and it is your responsibility to track execution and provide appropriate feedback to the users.

View & Data

The second API, View & Data, allows you to create custom web applications that enable your customers to visualize and interact with **2D and 3D design data** in a web browser without installing any plugin. The visualization is done via **WebGL** which is basically supported by all major Browsers, desktop or mobile, and created specifically for the web. It supports over 60 file formats like **DWG, RVT, IPT, DWF, FBX** and many others. The **REST** and **JavaScript** APIs make it easy for you to quickly develop web applications to consume this API and enrich your cloud solutions.

Features and usage

The View and Data API has Server and Client features. When used together those features can provide rich user experiences never seen on web based solutions. Some of these features are:

- Upload models via REST API and the web service creates the viewing stream.
- Access to model and model components data, using JavaScript, in your web page.
- 2D and 3D models into browser with textures, visualizations, navigation and design data.
- Browser compatibility with Chrome, Safari, Internet Explorer and Firefox.
- More than 60 design file formats in your web app.
- Zero plug-in install, no downloads.

Abstractions and concepts

The View & Data API is basically a translation service where you upload an input set of files and receive a WebGL data stream with graphical elements coupled with metadata generated from the input files. The process is based on the following abstractions and concepts:

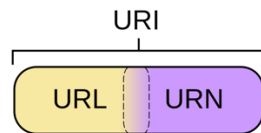
Bucket: It is basically a storage folder where you upload the input files. The input files can be sometimes just a single file, like a DWG, or even a set of files like an Inventor assembly composed by several parts. When you create a bucket at the server it can follow one of these retention policies:

- **Transient:** cache-like storage that persists for only 24 hours, ideal for intermittent objects.
- **Temporary:** storage that persists for 30 days. Good for data that is uploaded and accessed, but not needed later. This type of bucket storage will save your service money.
- **Persistent:** storage that persists until it's deleted. Items that have not been accessed for 2 years may be archived.

OSS (Object Storage Service): The OSS is responsible for storing and maintaining, secured by **OAuth 2.0**, the files you upload to View & Data service. Files are kept anonymous and stored accordingly to the bucket's policy. Each object stored into the bucket must have a unique key so the OSS can overwrite (update) this object when necessary. OSS also does not support versioning. To upload multiple files, you will also need to establish the **parent/child relationship**.

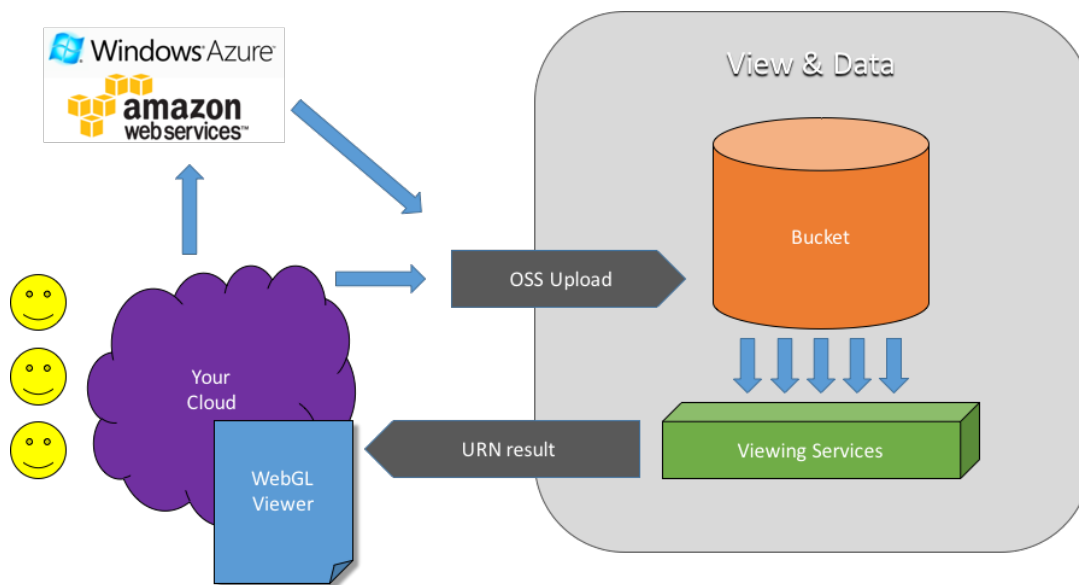


Viewing Services: Once you have your files uploaded you can request the translation service which will process the file set by the input **URN (Uniform Resource Name)** provided and, if succeeded, return a result URN to stream the resulting model. All URN must be **base64 encoded**. A URN, which is a **URI (Uniform Resource Identifier)**, functions like a person's name, while a **URL (Uniform Resource Locator)** resembles that person's street address. In other words: the URN defines an item's identity, while the URL provides a method for finding it.



The View & Data workflow is quite simple:

- Register an account and create an application.
- Use the authentication API to get an access token. The access token is required for all other API calls.
- Create a bucket in which to store your objects.
- Upload an object to the bucket and request that it be translated into a viewable object by the viewing service.
- At this point you have a uniform resource name (URN) for your object, which you pass to the viewing client.



Storage Services

Both APIs presented here rely on a stable Cloud front end client because they weren't meant to be accessed directly. Your Cloud solution should protect your credentials and hide anything relevant to the proper work of your services. If you can't afford (or don't want) to provide your own storage infrastructure it is highly recommended you adopt some stable and well known storage services. We have pretty good options around but Amazon seems to be the most flexible, wide-available and cost effective solution. Of course this will depend on your existing web structure and current service provider. Autodesk itself uses Amazon a lot so you can also take an extra advantage to be sitting aside their own servers thus providing an even faster communication with AutoCAD I/O and View & Data servers.

If you plan to implement advanced Cloud solutions integrated to already existing services, you may end up being forced to adopt some specific storage solution but you can certainly adapt it to Autodesk services and consume the APIs with the same overall performance.

Using AutoCAD I/O and View & Data together

After introducing AutoCAD I/O and View & Data APIs it is time to make them work together so you can leverage your solutions with those powerful web services. The idea is to provide for an ordinary user, not being skilled in AutoCAD or any other CAD system, a simple yet powerful way to create complex drawings and view them online, through a web browser, by consuming these services embedded into your own website.

To do that we will use 3 of the Autodesk Github samples located at:

<https://github.com/Developer-Autodesk/library-dotnet-autocad.io>

<https://github.com/Developer-Autodesk/workflow-aspdotnet-autocad.io>

<https://github.com/Developer-Autodesk/workflow-winform-autocad.io>

For your convenience the 3 projects were packed into a single VS2012 solution and migrated to AutoCAD I/O V2 (with a custom build) thus emphasizing the changes from AutoCAD I/O V1 to V2. The NuGet packages were also unified and shared across these two projects. JavaScript libraries were updated as well.

The main sample, “**workflow-aspdotnet-autocad.io**” implements a simple ASP.NET website with a simple input form to build a custom furniture. The web input form collects information about dimension and some other features like drawers and send this information to AutoCAD I/O where a custom AppPackage will run and generate a new DWG (on the fly) with the resulting geometry. This dwg is then saved into an Amazon S3 storage bucket and then forwarded as an input data to View & Data service. The DWG is translated and the result URN is sent back to the website where the JavaScript viewer will display the result.

The beauty of this sample is to completely hide the complexity from the end user. In fact, the user accessing the website will be running AutoCAD behind the scenes without any expertise. Further,



also without any specific knowledge, the end result will magically appear at the same web page after a couple of minutes. The user will then intuitively spin his mouse and see the 3D model orbiting right in front of his eyes without having to install anything.

Conclusion

This is indeed just a basic demonstration of the huge capabilities of the two Autodesk Cloud services being used alone or together. The AutoCAD I/O empowers your Cloud to process DWG files without any user interaction and without having to acquire a license, to teach users how to use AutoCAD but achieve extremely efficient results with native AutoCAD features or with enriched applications you provide to attend specific business rules.

In other hand, View & Data completes the historical lack for viewing capabilities on the web. By taking advantage of the WebGL processing and mobility, you can offer to users a easy yet powerful tool to visualize more than 50 file formats without imposing the labor intensive work of Plugin installation at the client's machine. The data layer behind the actual 3D/2D model leverages your Cloud capabilities by extracting valuable underlying data that could enrich the user experience by coupling them to visual elements.

We can expect to see much more during the next years and also see those services evolving really fast. As the Cloud solutions become more abundant, cost effective and considered as unlimited computing power, these APIs will certainly work quite well allowing you to concentrate your skills and investment in your core business without suffer from the technology side effects.

If we add the benefits of being totally mobile friendly, with the stable increase of 4G networks in terms of speed, stability and range, your Cloud solutions will be much more easily adopted and efficient once the zero-client approach applies on these technologies. The maintenance costs are also dramatically reduced and the support will be also outstandingly improved.

The web was never so powerful and capable of addressing complex problems like it is today!

