



TRANSFORMING ENGINEERING

Referenced Files as a Service: A new way to provide data to Revit

Diane Christoforo
Senior Software Engineer

The Whole Talk, In Three Sentences

- Revit models contain data from many external sources.
- We built a framework that allows third-parties to register plugins to provide this data.
- Now we don't have to be responsible for handling every format someone will ever want.



Outline

- This slide
- **The Problem:** It's hard to parse data from many sources
- **The Idea:** We don't care where the data comes from
- **Framework Design & Implementation (plus demos!)**
- Q&A



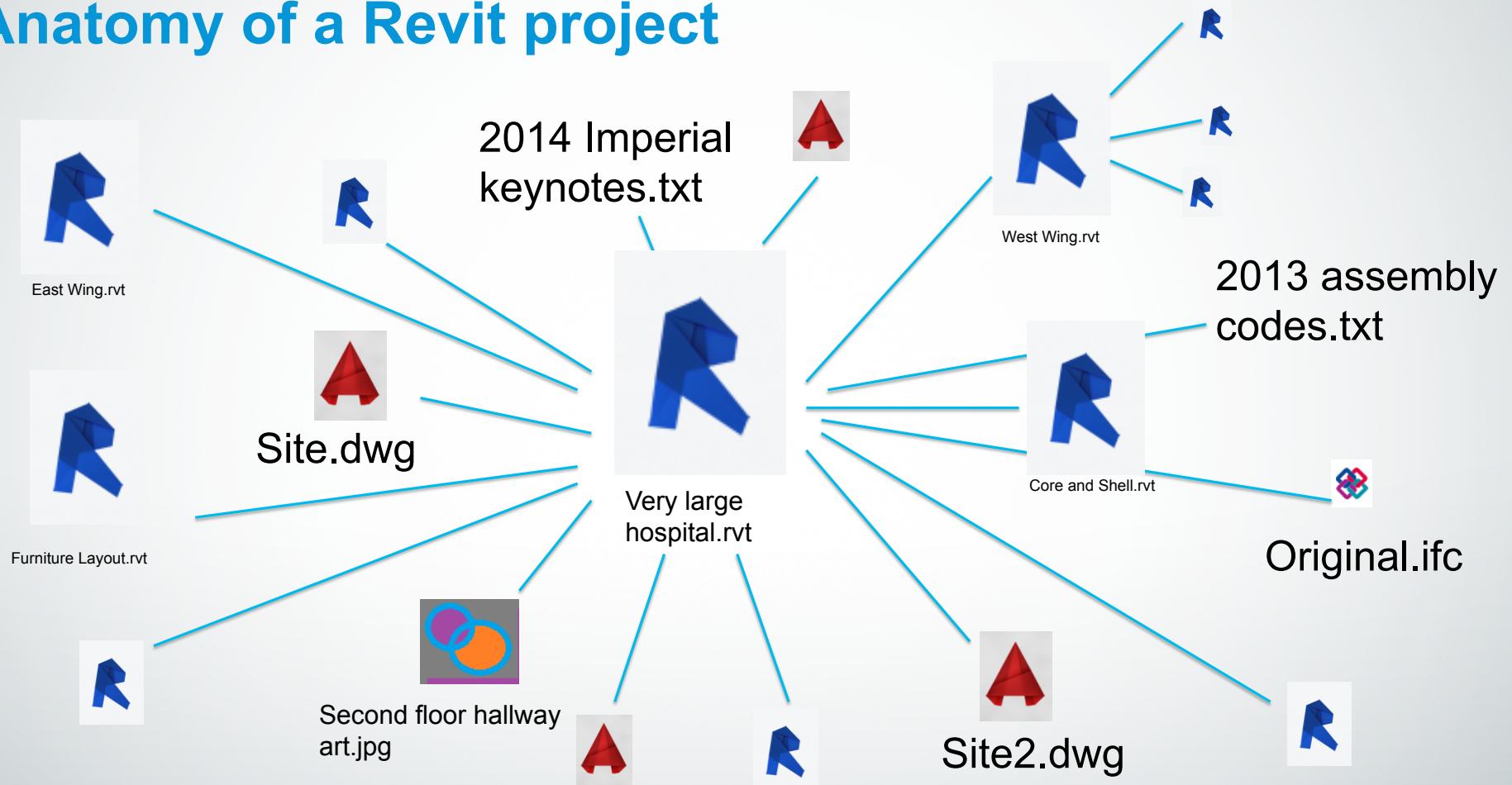
The Problem

Anatomy of a Revit project



Very large
hospital.rvt

Anatomy of a Revit project



Users say:



I'm going to download this model from Autodesk 360. You can automatically get all the links and set them up, right?



We want to put our files on this website. You can get them from there, right?



We want Revit to build one keynote table from two files so we don't have to put it together by hand. You can do that, right?

Revit says:



The developers haven't taught me how to use A360.



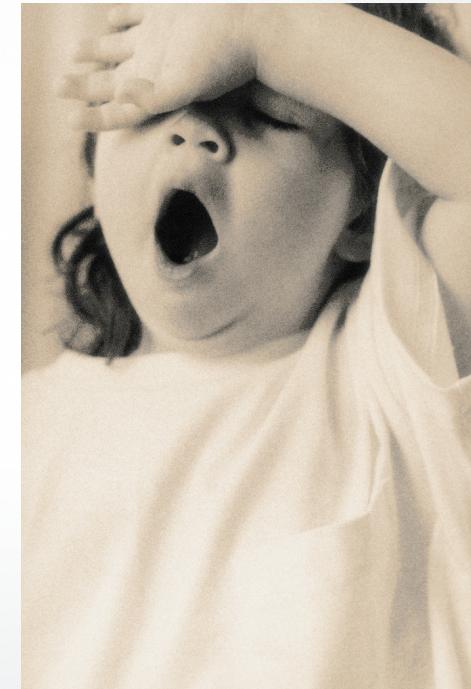
What's a website?



The developers haven't taught me how to do that.

Every time we want to get data from a new place, we have to:

- Add new UI
- Add new parsing code
- Understand and handle errors



Where we're going, we don't need files

The cloud doesn't have "files", so even if we did implement all the individual features people request, we still have a problem.



The Idea

The idea: move to a service model

1. Think of external data as a “resource”, not a “file”.
2. Revit no longer handles all data itself.
3. Developers register a plugin to provide resources.
4. Revit doesn’t care where the data started out.



**Now people can provide the data from
whatever source they like!**

Design and Implementation

Design Goals

1. Impose as few format restrictions on the user as possible.
2. Local files should also use the new framework. (“eat your own dog food”)



ExternalResourceService

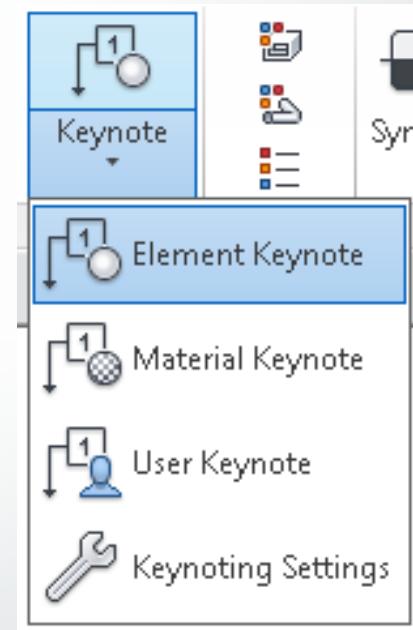
The plugins are built on top of Revit's existing ExternalServices framework. We added two interfaces:

- **IExternalResourceServer**
 - Tells Revit what resources it can provide
 - Loads resources when requested
- **IExternalResourceUIServer**
 - Provides feedback to the user after resources have been loaded

What we did in Revit 2015

You can use the External Resources framework to build a plugin supplying:

- Revit links
- Keynotes
- Assembly codes



Wait, what's a keynote?

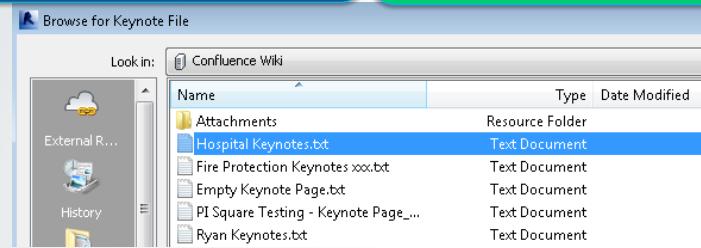
- Revit links – You can link one Revit model into another in a read-only fashion.
- Keynotes – Keynotes are a special kind of tag which indicates information about the purpose of an object. (For example – 1001010 “Fire Safety”.)
- Assembly codes – Assembly codes are a classification system for families. They’re based on Uniformat.

The screenshot shows a Confluence page titled "Hospital Keynotes". The page header includes the Autodesk logo, navigation links for Spaces, People, Calendars, Browse, and Create, and a search bar. The main content area displays a table with two columns: "Key" and "Text". The table lists various requirements, many of which have sub-items. The page was created by Emily Marcus and last modified by Diane Christoforo.

Key	Text
01100	Summary
01200	Price and Payment Procedures
01300	Administrative Requirements
01400	Quality Requirements
01500	Temporary Facilities and Controls
01600	Product Requirements
01700	Execution Requirements
01800	Facility Operation
01530.A1	Temporary Dustproof Partition
01530.A2	Temporary Protective Floor Cover
01530.A3	Temporary Protective Passageway
01530.A4	Temporary Protective Wall Cover
01530.A5	Temporary Shoring, Bracing And Support
01530.A6	Temporary Weatherproof Closure
02050	Basic Site Materials and Methods
02070	Geosynthetics
02100	Other

Powered by Atlassian Confluence 5.4.4, Team Collaboration Software · Report a bug · Atlassian News

1.) Revit asks the plugin for the list of resources it provides.



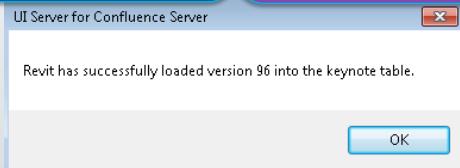
2.) User selects a resource

3.) Revit asks the plugin to load the resource

4.) The plugin gives Revit the resource

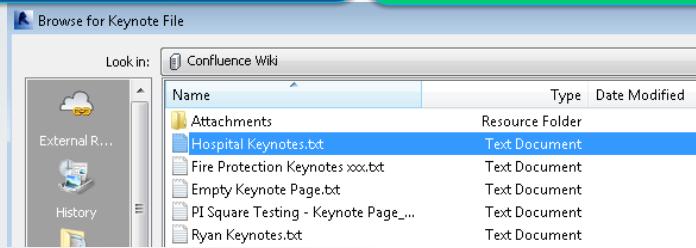
Key	Text
01100	Summary
01200	Price and Payment Procedures

5.) Revit asks the plugin to handle any errors



6.) The plugin handles errors or displays UI

1.) Revit asks the plugin for the list of resources it provides.



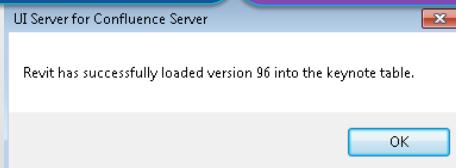
2.) User selects a resource

3.) Revit asks the plugin to load the resource

4.) The plugin gives Revit the resource

Key	Text
01100	Summary
01200	Price and Payment Procedures

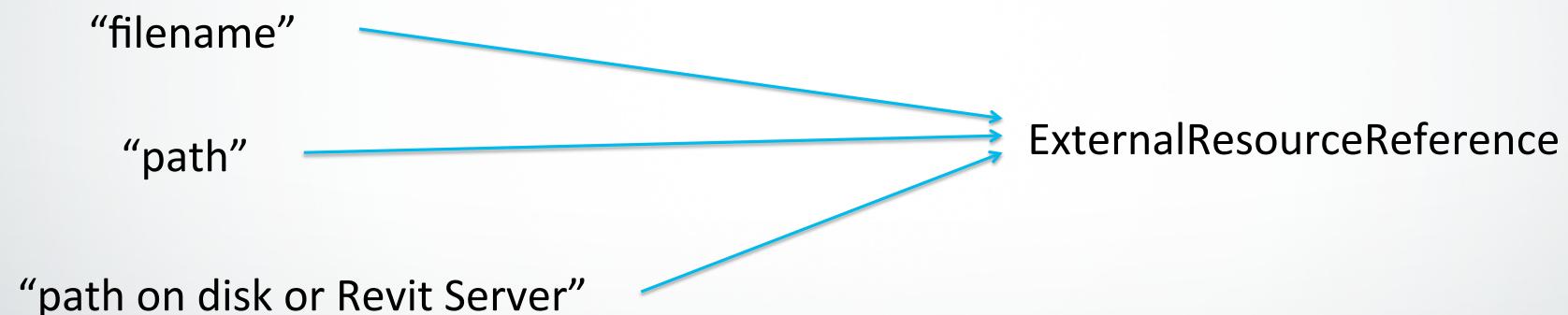
5.) Revit asks the plugin to handle any errors



6.) The plugin handles errors or displays UI

Describing a resource – ExternalResourceReference

Instead of storing file paths, external resources now store a new structure called an ExternalResourceReference.



Inside the ExternalResourceReference

- (Guid) serverId – Which plugin provides the resource.
- (String) version – The version of the resource.
- (String) inSessionPath – The “display name” of the resource – what the user will see.
- (IDictionary<String, String>) referenceInformation – Info which the plugin needs to identify the resource. The plugin defines the format.

Resource maps can be as simple as:

“resource” = “4”

or more complicated:

“name” = “hospital third floor restroom”

“user” = “dchris”

“language” = “en-us”

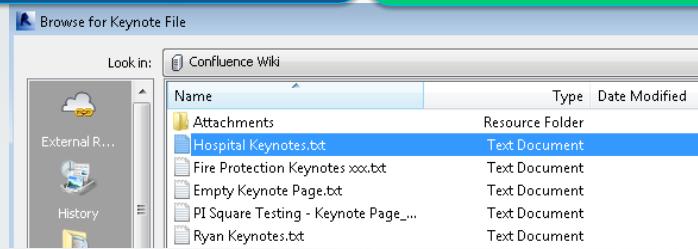
“format” = “txt”

“last-accessed” = “3-26-14”

The screenshot shows a web browser window with the Autodesk Confluence interface. The title bar displays the URL <http://wiki.autodesk.com/display/rnts/Hospital+Keynotes>. The page header includes the Autodesk logo, navigation links for Spaces, People, Calendars, Browse, and Create, and a search bar. The main content area has a breadcrumb trail: Revit Ninja Teams / Home / RFaaS Confluence Keynote Demo. The page title is "Hospital Keynotes". It shows a table of requirements with 15 rows, each containing a key and its corresponding text description. The table is styled with alternating row colors.

Key	Text
01100	Summary
01200	Price and Payment Procedures
01300	Administrative Requirements
01400	Quality Requirements
01500	Temporary Facilities and Controls
01600	Product Requirements
01700	Execution Requirements
01800	Facility Operation
01530.A1	Temporary Dustproof Partition
01530.A2	Temporary Protective Floor Cover
01530.A3	Temporary Protective Passageway
01530.A4	Temporary Protective Wall Cover
01530.A5	Temporary Shoring Bracing And Support

1.) Revit asks the plugin for the list of resources it provides.



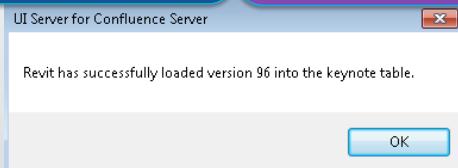
2.) User selects a resource

3.) Revit asks the plugin to load the resource

4.) The plugin gives Revit the resource



5.) Revit asks the plugin to handle any errors



6.) The plugin handles errors or displays UI

IExternalResourceServer.LoadResource()

```
void IExternalResourceServer.LoadResource(  
    Guid loadRequestId,  
    ExternalResourceType resourceType,  
    ExternalResourceReference desiredResource,  
    ExternalResourceLoadContext loadContext,  
    ExternalResourceLoadContent loadContent)
```

- **loadRequestId** – Revit sends a Guid with every load request so they can be uniquely identified.
- **resourceType** – Useful for plugins that handle multiple types of resource.
- **desiredResource** – The resource which Revit would like the plugin to load.
- **loadContext** – Information about the state of Revit at the time of the request.
- **loadResults** – **The results of loading the resource.** Each resource type has its own subclass.

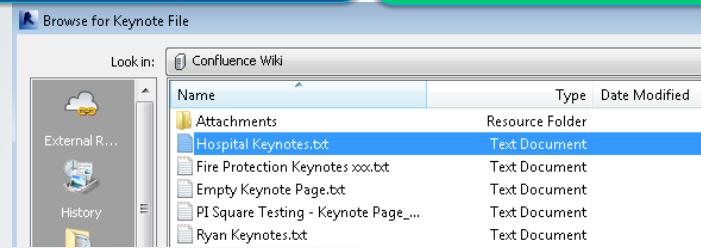
ExternalResourceLoadContent

- Each resource type has its own subclass.
- Keynotes and assembly codes **can be built directly.**
 - The plugin builds a table of key/value pairs and returns it to Revit.
 - No files needed anywhere!
- Revit links have to return a locally cached path.
 - Revit's still the only thing that can build a Revit model.
 - The user doesn't need to know or care that there's a local copy of the model.

IExternalResourceServer – other functions

- IsResourceWellFormed – We don't know what format the server is using, so it has to answer this question for us.
- AreSameResources – Again, the server needs to tell us if two resources are the same.
- GetResourceVersionStatus – Revit skips loading if we already have the most recent version of a resource.
- SetUpBrowserData – This is how the server tells Revit what resources it can provide. Revit then lists the resources in the UI.
- SupportsExternalResourceType – Servers can provide any or all of keynotes, assembly codes, and Revit links.

1.) Revit asks the plugin for the list of resources it provides.



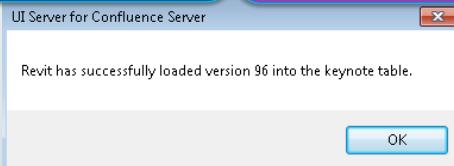
2.) User selects a resource

3.) Revit asks the plugin to load the resource

4.) The plugin gives Revit the resource

5.) Revit asks the plugin to handle any errors

6.) The plugin handles errors or displays UI



We don't know what will go wrong

How can Revit respond to errors like these?

Revit model “MyServer://
myhouse.rvt is missing from the
server.

The network connection is very
slow, so the operation was
cancelled. Try again?

You aren't logged into the
website. Log in:

Your subscription to “Keynotes
for Everybody” has expired.
Would you like to renew?

Your computer is full of bees.
Please reboot.

We couldn't reverse the polarity
of the neutron flow.

IExternalResourceUIServer.HandleLoadResourceResults()

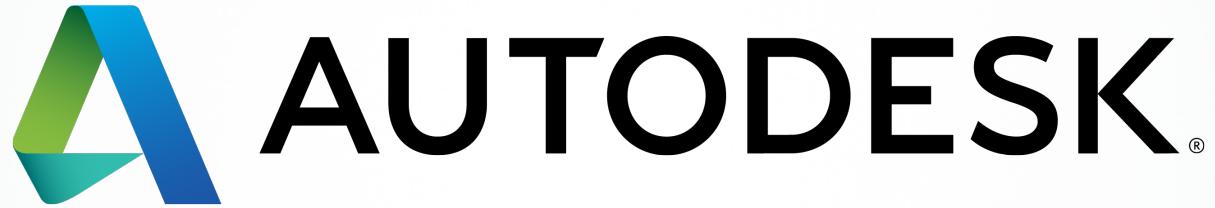
```
void  
IExternalResourceUIServer.HandleLoadResourceResults (  
    Document document,  
    IList<ExternalResourceLoadData> loadData)
```

- **document** – The Revit model which the resource is loaded into.
- **loadData** – A list of load results. The plugin can see any Revit-internal errors which occurred, and decide what UI, if any, to display.

Where to learn more

You can get a link to the Revit 2015 SDK (including our sample) at:

<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975>



Autodesk is a registered trademark of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document. © 2014 Autodesk, Inc. All rights reserved.