

Grasshopper for new Connection Rest API

Value Proposition

With the new version of the plugin, users are provided with enhanced features of the new API and an easier GH experience.

Old (0.1.#)	New (> 0.2.0)
• Hard to understand what is happening in the API when using components. • Lots of IO operations. • Long operations (like calculation) cause the script to freeze. • Lots of components for all objects. • Did not work with multiple connections in a project. • Hard to choose specific version of IDEA.	• Very tightly coupled to the actual working of the API. (easier transition to python or c#) • Projects are on server side until saved by the user. • API components are now async. Users can continue the script while components are calculating. • Automatic decomposing of API Objects. • Works natively with all connections in project. • API attacher / runner allows easy version specification.

Grasshopper Versioning

A new version of the plugin will be available as 0.2.0 on the Rhino Package Manager. **This will work with version of IDEA StatiCa > 24.1.0**

Old version 0.1.# will continue to be available and work with version < 24.1.2.

New Plugin Working

The new Grasshopper plugin provides a new experience more in-line with modern API Grasshopper interaction. Each component is tightly coupled to API interaction.

Installation of the plugin

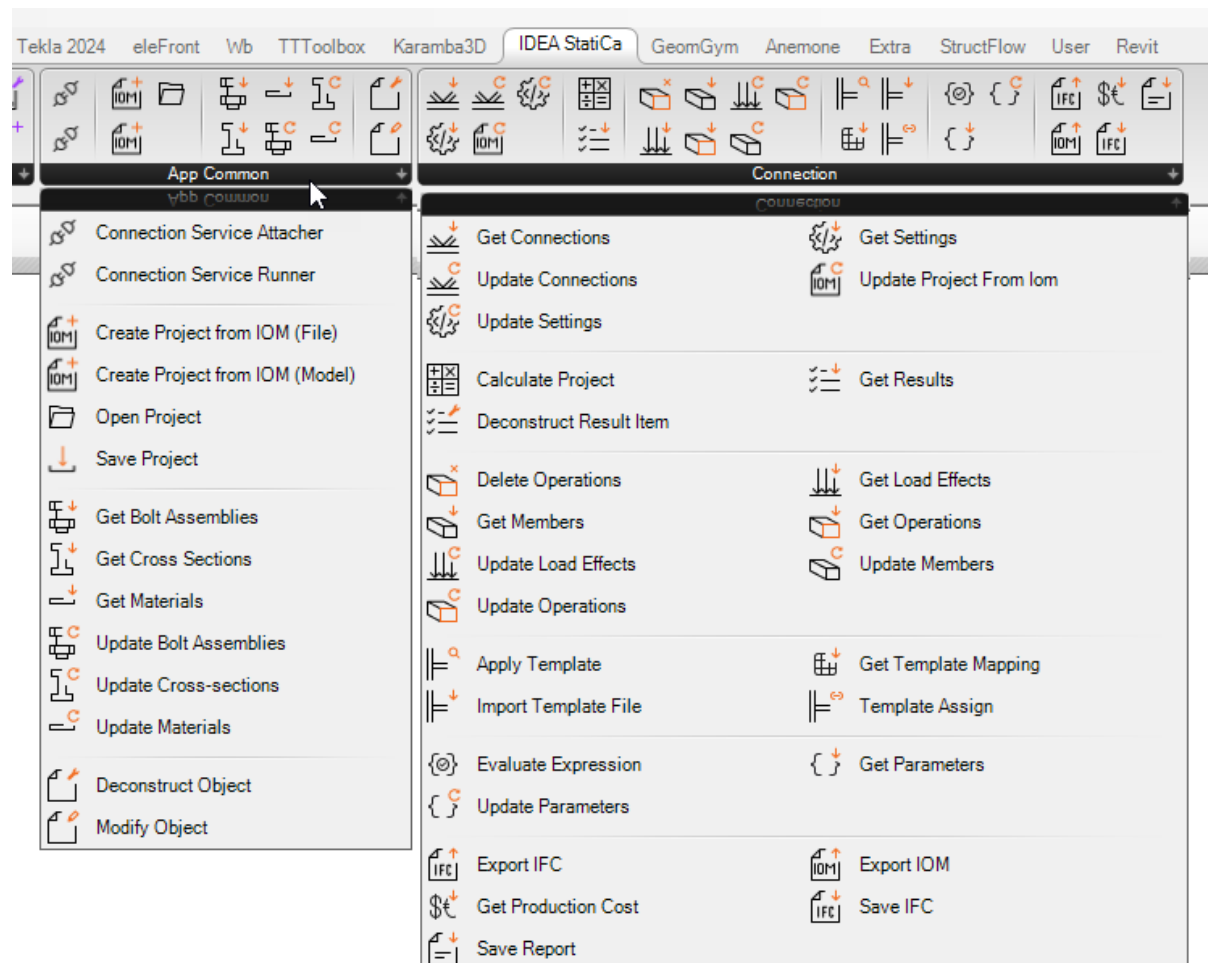
Plugin installation is like before. The plugin is installed through the Rhino Package Manager. Get to the Package Manager by typing *PackageManager* in the Rhino command prompt. Simply update or install the latest available version of *IDEA StatiCa*.

Component Overview

The new plugin is broken into two panels in the Grasshopper ribbon.

App Common: API components which are common across all IDEA StatiCa design API's.

Connection: Component which are specific to the connection API.

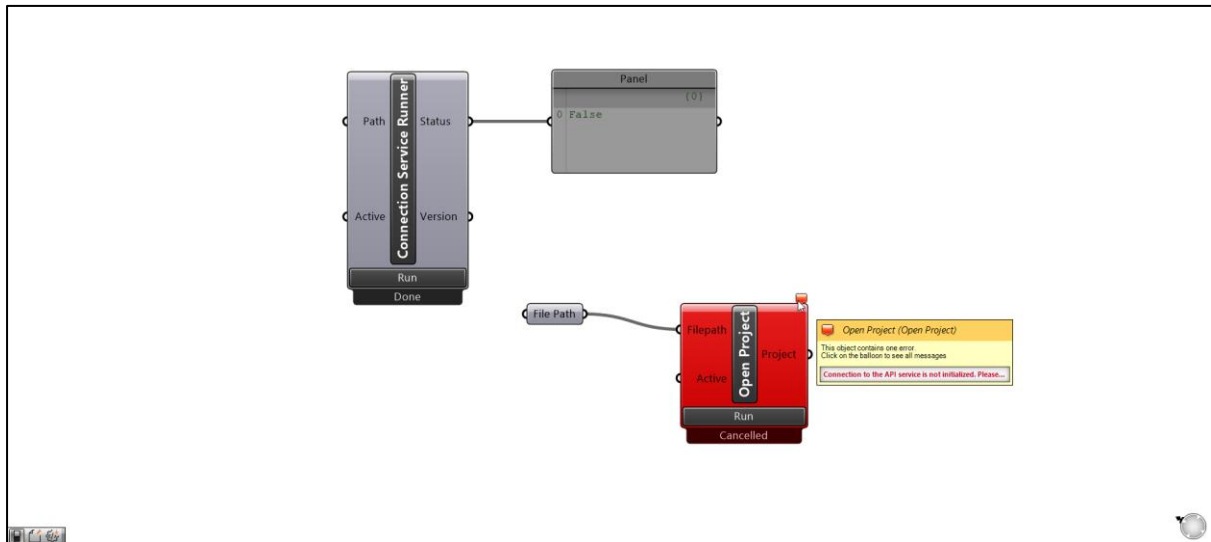


Connecting to the API through GH

As similar with API clients, you can connect the Grasshopper script to a runner API instance (Connection Service Attacher) or by starting the service within Grasshopper (Connection Service Runner).

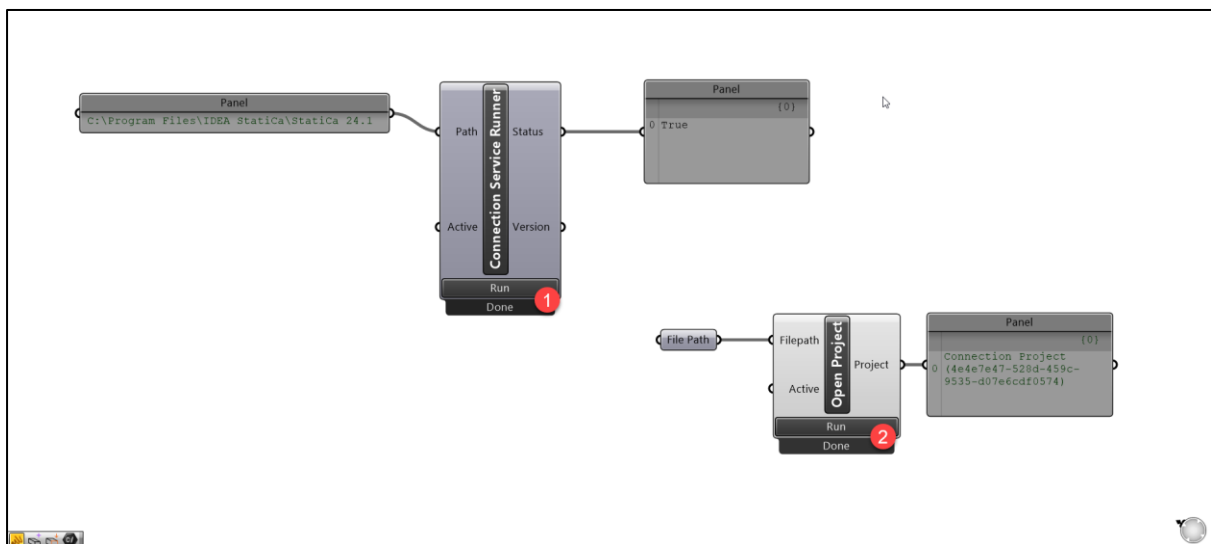
A user must select one of the above components and run them before any of the API components are able to run.

WRONG



You will be provided with an error message on the component if the component has tried to run without a valid API connection.

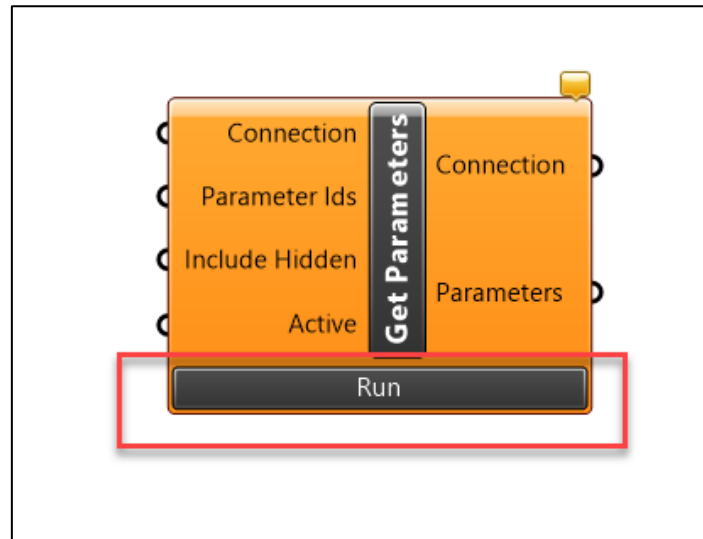
CORRECT



Once there is a valid API connection, the Service component should not need to be re-run until the script is closed or navigated away from. The connection is stored locally so that all components can access it.

API Components

Components which interact with the API are easily identified with the “Active” button below. Component can be run once by clicking the Run button or can be continually executed by using a true/false toggle in Grasshopper wired into the Active Component.



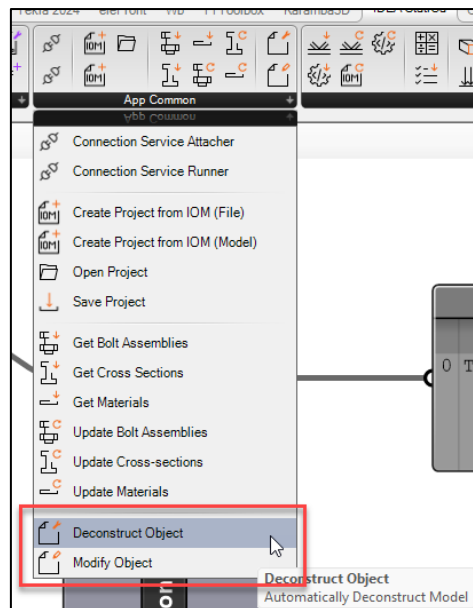
API components are closely related to Controllers in the Connection API. Controllers are either action based (Calculation, Export, Report) or object based (Load effect, Member, Parameters etc).

Object Based: Object based controllers will typically have a Get Component and an Update Component.

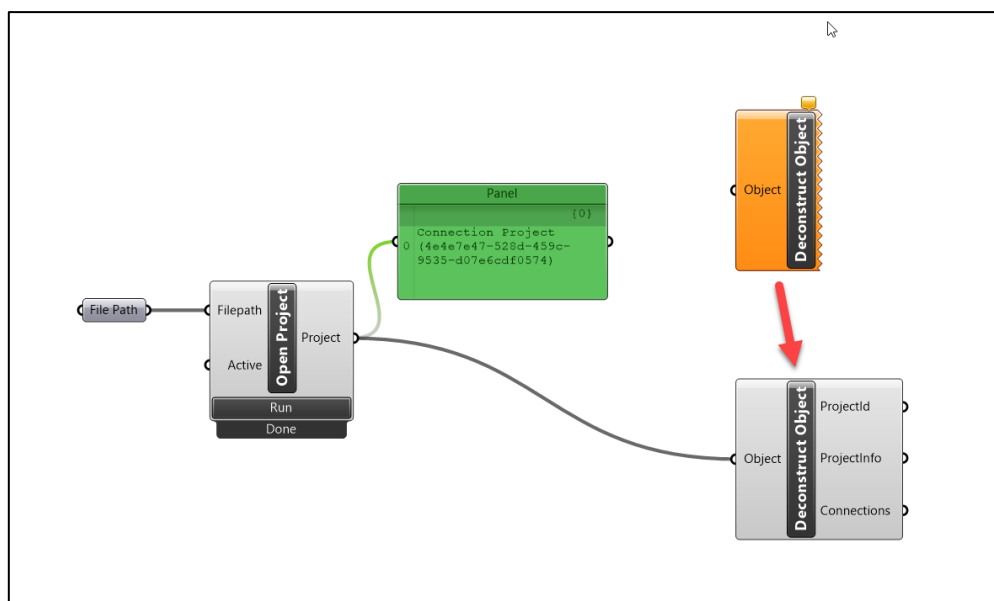
- **Get:** Retrieves all the objects of the controller type which are in the Connection and retrieve these specific objects which can be deconstructed or modified.
- **Update:** Updates provided objects that have been updated in the Grasshopper script. Objects are updated using the automatic Modify Object component.

Action Based: Action based controller do specific tasks or operations on a specific connection or project, these include calculation or exporting functionality

Most API components will take inputs and provide outputs of objects which can be deconstructed or modified. Providing components in this format reduces the overall number of components that we need to add to the Plugin. It also helps provide a better flow to the user as they always only need to find one component based on the action they are trying to achieve.



The most basic example is deconstructing a Project to get the available information within it, such as the connections available.

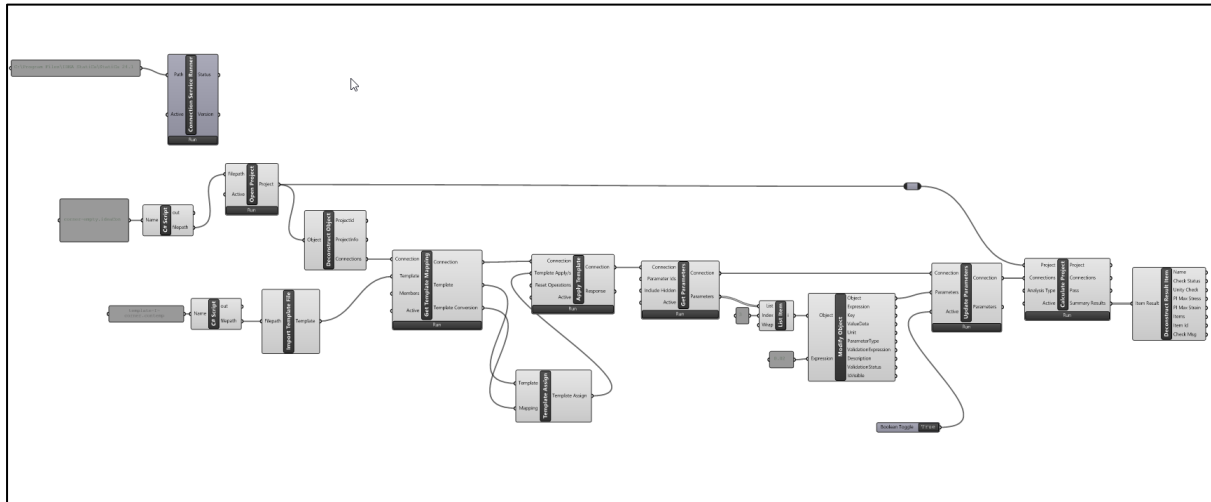


See the simple example below for how the modify component works.

In the future we are also looking to add a Create or Add component which will allow the user to create specific objects that can be added to the API.

Simple Example – Updating Connection Parameters

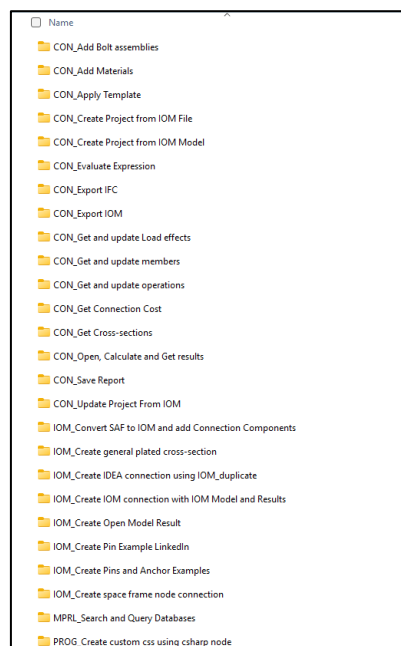
Below shows an example of using API and Automatic Object components in a simple example of applying a template and updating connection parameters.



Examples

A full list of examples is available to users that download the plugin:

%appdata%\McNeel\Rhinoceros\packages\8.0\IdeaStatiCaGrasshopper\[0.0.0]\Examples
0.2.0.zip



Functionality Improvements

- When creating a Connection project from IOM, multiple connections are now imported into one project.
- Export IFC directly in GH.
- Template application workflow is unified with the application.
- Pins and Anchors are now available to be defined in the IOM Components.
- Versioning upgrade and downgrade is now implemented.