Salamander 3: Plug-In Creation Guide

Salamander 3 is designed to be easy to extend with extra functionality. New commands can be defined in an external assembly and are loaded dynamically at runtime. This guide will take you through the basics of writing your own plugins.

# Actions

Salamander commands and components are all defined as ‘Actions’. An Action is a single operation which has a set of input properties and a set of output properties. It is responsible for defining what those inputs and outputs are and the process which transforms one into the other. Actions do not define where the input data comes from or how the user enters it, however – Salamander resolves this automatically depending on the context. This allows exactly the same action to be run in different environments. When run in Rhino, for example, Salamander will prompt the user to enter the necessary information via the Rhino interface. When run in Grasshopper, a component with the required input parameters will be automatically generated.

Technically, an Action is a class which implements the IAction interface. Inputs are simply properties of the class tagged with an ‘ActionInput’ attribute and outputs are properties with an ‘ActionOutput’ attribute attached. An ‘Execute’ function must also be defined, which is called when the action is run.