

Inversion of Control (IoC)

Inversion of control can be described as "an abstract principle describing an aspect of some software architecture designs in which the flow of control of a system is inverted in comparison to procedural programming." [1] IoC is a pattern that naturally results from the proper organisation of dependencies in a body of code. It typically results in delegation of power over instantiation and wiring of dependencies to a framework, rather than writing manual code to sequentially achieve the task in a particular situation. This page provides resources to help you understand why and how the Infusion Framework uses Inversion of Control.

[Why Inversion of Control?](#)

[How to Use Infusion IoC](#)

[IoC Pipeline](#)

[Subcomponent Declaration](#)

[Contexts](#)

[Demand Resolution](#)

[Demands Specifications](#)

[Expansion of Default Component Options](#)

[Controlling The Timing of Subcomponent Creation](#)

[Invokers](#)

[Pseudoarguments](#)

[Event injection and boiling](#)