# ApplicationLifeCycle library

A nuget package found in GamingSystemNuget.

* [Purpose](#ApplicationLifeCyclelibrary-Purpose)
* [Pros](#ApplicationLifeCyclelibrary-Pros)
* [Cons](#ApplicationLifeCyclelibrary-Cons)
* [Key Players](#ApplicationLifeCyclelibrary-KeyPlayers)
* [Gotchas](#ApplicationLifeCyclelibrary-Gotchas)

## Purpose

The ApplicationLifeCycle library centers are the SimpleInjector IOC container (<https://simpleinjector.org/>). It is used to make the implementation and registration of abstractions to implementations easy.

## Pros

Each library usually contains its own list of registrations. This makes it as easy as adding a project as a dependency to another project.

## Cons

Adding a project automatically adds the registrations. This can cause problems if the default registrations are not the desired dependencies. There are ways to overcome this but sometimes does cause headaches.

## Key Players

* CompositeRootBase
  + Each project gets one or more classes that inherits from this class.
  + This is an abstract class that forces the overriding of a method called “registerBindings”.
  + When the system is started, most dlls in the working folder are loaded and searched (there are a few hardcoded sections such as microsoft dlls, system.dlls, etc.) .
  + Any class that inherits from CompositeRootBase has its registerBindings() method called giving the class the ability to register dependencies with the GlobalContainer.
* IRequestLifeCycleStartupItem
  + Any class that is registered that implements the IRequestLifeCycleStartupItem interface, automatically has its “ExecuteAsync()” method called during the starting of the ApplicationLifecycle.
  + Each one of these implementations is given a priority (RequestLifeCycleStartupItemPriority). They are run in order allowing the developer to specify normal loading patterns.
  + These implementations have a short lifecycle.
* ISystemService
  + Implementations of this interface live throughout the lifetime of the application.
  + These are usually thread based services that allow for things such as backups, monitoring, heartbeat, etc.
* SystemServiceScheduledTask
  + Usually a property of an ISystemService implementation
  + Allows a callback to be set to be called at certain intervals.

## Gotchas

* Removal of a class library
  + If a class library is removed from a project but the dll is still left in the bin folder, it is possible that the library will be loaded and the compositeRoot registerBindings called yielding some really strange behavior. Fixing this is as simple as clearing the bin folder.
* As mentioned under “cons”, overriding behavior can become tricky. It is easier to remove the default registration and add the necessary registrations to dependencies that use it.