# Default Documentation

This file contains summary information about the deployment spike process.

The goal of this process to simply perform a number of common deployment scenarios and have those scenarios covered with thoroughly with logging, diagnostics and notifications.

The process works by launching and then scanning the assemblies in the executing folder. Assemblies can be specified in the configuration file but if nothing is configured then all assemblies will be scanned. Any classes that implement DeploymentSpike.Interfaces.ITask or DeploymentSpike.Interfaces.INotification will be loaded into the container. The tasks are then executed based on the task steps configuration. After all tasks have been executed the task diagnostic information would be generated. If any of the tasks throws and exception or fails then the process will halt and not continue.

## Standard Usage and Expectations of this Process

* This process would be run after tasks have been create and placed in the executing assembly folder. A task is a class that executes DeploymentSpike.Interfaces.ITask.
* Task order and details are controlled by the task steps configuration file.
* Tasks are run in a serial manner and if a task throws an exception the process is halted.
* Setting interfaces should be defined on the constructor so that the container can pass the correct values to the task.
* Task settings are expected to be an json serialized version of the DeploymentSpike.Interfaces.ISettings interface and should be located in the config folder underneath the application directory. A different location for the config file can be specified in the app.config file of the application.

## Configuration Setup and Precedence

There are two main places to inject settings into a task or notification. The default configuration would be the settings located in the component name in the configuration directory. A secondary configuration is available if a configuration file is specified in the task steps configuration item. If both files are present then the component settings would be loaded and then the task settings would be loaded and any collisions would use the task settings.

## Creation of a New Task

If a new task is desired then then the DeploymentSpike.Interfaces.ITask interface would need to be implemented and then the implementation would need to be copied into the executable folder. If there are configuration needs for the implementation then those would need to either be setup in the default component name configuration or the task configuration.

## Creation of a New Notification

If a new notification is desired then then the DeploymentSpike.Interfaces.INotification interface would need to be implemented and then the implementation would need to be copied into the executable folder. If there are configuration needs for the implementation then those would need to either be setup in the default component name configuration or the task notification configuration. A notification would be fired on the start of a task and on the completion of a task.

## Further Information

For detailed information see the files:

* ConfigurationSettings.txt
* TaskDescriptions.txt
* TaskSteps.txt
* NotificationDescriptions.txt