STF

System Test Framework

STF - Plugin Developing Guide

Table of Contents

[2 Introduction 3](#_Toc434060425)

[2.1 Who is making STF – and why 3](#_Toc434060426)

[2.2 Logging 3](#_Toc434060427)

[2.3 Configuration 3](#_Toc434060428)

[2.4 Plugins 3](#_Toc434060429)

[2.5 Utils 3](#_Toc434060430)

[3 Components 4](#_Toc434060431)

[3.1 StfArchiver 4](#_Toc434060432)

[3.2 StfAssert 4](#_Toc434060433)

[3.3 StfConfiguration 4](#_Toc434060434)

[3.4 StfKernel 4](#_Toc434060435)

[3.5 StfLogger 5](#_Toc434060436)

[3.6 StfPluginLoader 5](#_Toc434060437)

[3.7 StfTestScriptBase 5](#_Toc434060438)

[3.8 StfUtils 5](#_Toc434060439)

[4 Samples 6](#_Toc434060440)

[5 Getting Started 7](#_Toc434060441)

[5.1 Start with a MsTest standard unit test 7](#_Toc434060442)

[5.2 Get the NuGet package (Mir.Stf) 7](#_Toc434060443)

[5.3 Get Stf into action 8](#_Toc434060444)

[5.4 The StfLog 9](#_Toc434060445)

[5.5 The Stf Asserter 11](#_Toc434060446)

[6 Sources and drops 12](#_Toc434060447)

[6.1 License 12](#_Toc434060448)

[6.2 Source code is on github 12](#_Toc434060449)

[6.3 NuGet package 12](#_Toc434060450)

# Introduction

## What is the role of a STF plugin

Could be a model or an adapter

## Model

A model is a test representation of an interaction towards a SUT (System Under Test).

A model has no SUT specifics like OpenWindow, Clicks or other actions that binds towards a UI target. The corresponding actions would probably Open, and StartAction.

Creating the model is a balance of using a paradigm (like a UI) without limiting the model towards other targets (like Database, Services etc…)

## Adapter

The Adapter is the connection towards the SUT from the test stack. Could be an interface towards an InternetExplorer, or an Oracle Database interface.

# Components

A STF plugin uses the following components:

* StfConfiguration
* StfLogger
* StfPluginLoader
* StfUtils

## StfConfiguration

All plugins uses the StfConfiguration.

A plugin will reserve a section in the overall StfConfiguration tree – a bit like namespaces in C#.

In that section the plugin will have its own configuration tree, and a matching configuration object.

## StfLogger

All plugin logs uses the StfLogger. This ensures that all logging gets to the same place.

## StfPluginLoader

The plugin needs to be setup, do the StfPluginLoader recognizes your plugin, loads it and initializes correctly.

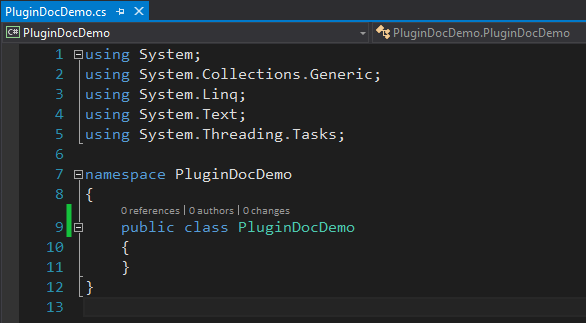
## StfUtils

Some utils that are usuable for scripting, and implementing test models.

# Samples

# Getting Started

## Start with a standard class

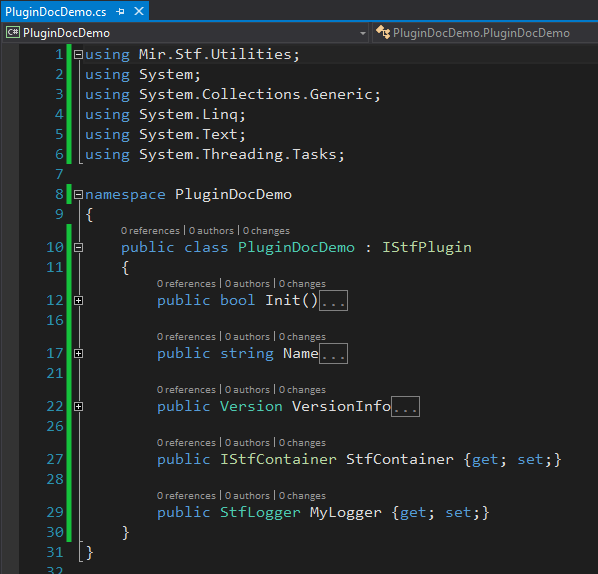


## Get the NuGet package (Mir.Stf)

Using the (Visual Studio) standard NuGet Package manager

## Get Stf into action

You do that by letting your plugin class implement the IStfPlugin.



## Methods for you to implement

### Init

Init will be called from the PluginLoader, when the plugin is loaded.

### Name

Name of your plugin for the STF kernel to use

### VersionInfo

The version of your plugin.

Other plugins (or test scripts) might depend on specific versions of it. This might come handy, when different test environments have different versions to be supported from your framework.

## Methods for you to not implement

Just give the methods a standard property {get; set;}.

The setter is called when your plugin is instantiated (Dependency Injected by Unity).

### StfContainer

The Unity container – leave it and love it. For advanced use, you can fiddle with it. For now it gives you a function Get<T>(), to get objects you need from the Stf system.

### MyLogger

This logger is a singleton logger. Shared among all current users of Stf.

# Configuration for the Plugin

## Plugin configuration interface

## Loading the config values

# Sources and drops

## License

Copyright governed by Artistic license as described here:

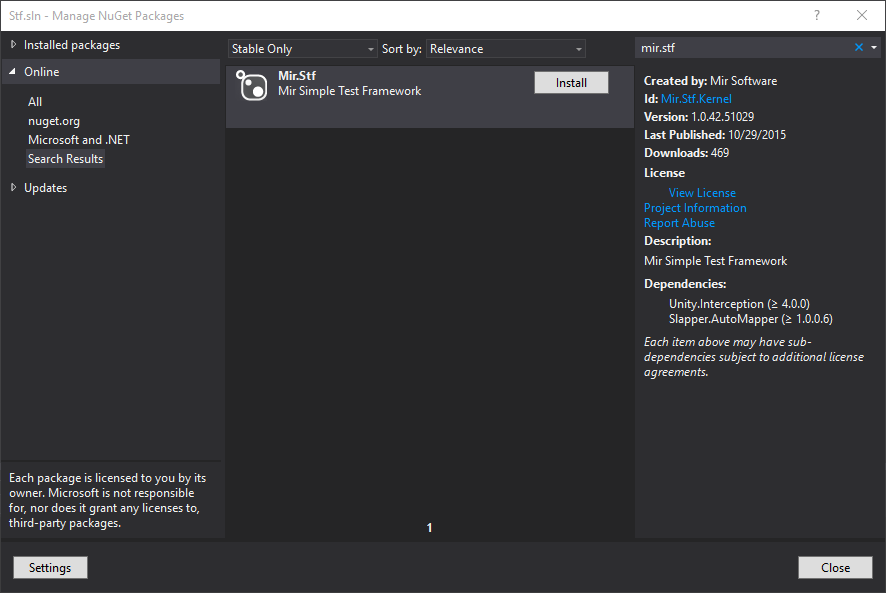
<http://www.perlfoundation.org/artistic_license_2_0>

## Source code is on github

<https://github.com/UlrichFreiberg/STF>

## NuGet package

Name is Mir.Stf



## The source of the demo is here:

STF\Demos\PluginDocDemo