# How to use RunSettings and acceptancetests.runsettings

The values in RunSettings.cs file are used locally by default, unless you select Test->Test Settings->Select Test Settings File and choose some other file , e.g. acceptancetests.runsettings file (**NOTE: RunSettings.cs is found it the TestData folder)**

## (NOTE: not sure if this should be static or not. In latest Portal example it is not static)

## RunSettings

public static class RunSettings

{

public static string EnvironmentKey { get; set; } = TestContext.Parameters["EnvironmentKey"] ?? "POR QA 01";

public static string WebRoot { get; set; } = TestContext.Parameters["WebRoot"] ?? "http://porapp01qa/";

public static bool EnableChrome { get; set; } = Convert.ToBoolean(TestContext.Parameters["EnableChrome"] ?? "true");

public static bool EnableIe { get; set; } = Convert.ToBoolean(TestContext.Parameters["EnableIe"] ?? "false");

}

Here is latest version in Portal, just uses => instead of {get;set}

public string EnvironmentKey => (TestContext.Properties["EnvironmentKey"] ?? "POR QA 01").ToString();

public string WebRoot => (TestContext.Properties["WebRoot"] ?? "http://porapp01qa/").ToString();

public string ApiRoot => (TestContext.Properties["ApiRoot"] ?? "http://porapp01qa:81/api/").ToString();

public bool EnableChrome => Convert.ToBoolean(TestContext.Properties["EnableChrome"] ?? "true");

public bool EnableIe => Convert.ToBoolean(TestContext.Properties["EnableIe"] ?? "false");

On the server, the EnvironmentKey in acceptancetests.runsettings is set (as part of the build?) and is read into RunSettings.cs. Therefore the EnvironmentKey property is set for each environment. The same is true for WebRoot and the other variables.

Therefore, all being well, the values after the ‘??’ will be ignored on the server. However, if these properties are not set, the default values will be used

However, if RunSettings is used for local runs, EnvironmentKey and WebRoot etc. are NOT set, so the values after the ‘??’ WILL be used. So, if someone updates values in RunSettings file (e.g. changing WebRoot to be localhost), then this will affect your tests that are run locally.

## acceptancetests.runsettings

<?xml version='1.0' encoding='utf-8'?>

<RunSettings>

<TestRunParameters>

<Parameter name="EnvironmentKey" value="POR DV 01" />

<Parameter name="WebRoot" value="http://porapp01DV/" />

<Parameter name="EnableChrome" value="true" />

<Parameter name="EnableIe" value="true" />

</TestRunParameters>

</RunSettings>

This is primarily intended for use by the server. It is generally ignored for local runs (see below for exception to this.)

When running on the server, the RunSettings.cs file will look in acceptancetests.runsettings for a parameter value. If it does not find one, then it will take the default value given in RunSettings.cs.

The advantage that acceptancetests.runsettings has over App.Config is that it can be set dynamically by TFS, which can override the initial values given in the file. This avoids the need of having a config file per environment.

Exception to above rule.

If you select Test->Test Settings->Select Test Settings File and choose acceptancetests.runsettings file, then the values in this file will be used instead for LOCAL test run.

So for running locally in QA, you should have the following values:

<Parameter name="EnvironmentKey" value="POR QA 01" />

<Parameter name="WebRoot" value="http://porapp01QA/" />

**NOTE: this ONLY works with MS test** runner **in visual studio and NOT Resharper test runner (unless we manage to get it working in the future).**

**For Resharper test runner you still have to use the values after the ‘??’ in** RunSettings