# How to register Test controller with team project collection using TFS APIs

## Introduction

We can programmatically register the test controller with TFS using APIs.

## Building the Sample

Just build the solution with Visual Studio 2012.

## Running the Sample

Run the sample and provide the collection name.

Note: if you don't provide the collection name, you'll see a message

"Usage: RegisterTestController <collectionUrl>"

Select the option for registering the controller.

It will then register the controller.

## Using the Code

Following code snippet shows how to register test controller with TFS using APIs.

|  |
| --- |
| -Code block start-  --C# code snippet start--  static void Main(string[] args)  {  if (args.Length != 1)  {  Console.Error.WriteLine("Usage: FindTestControllers <collectionUrl>");  Environment.Exit(-1);  }  string tfsUri = args[0];  int i = 0;  try  {  listController = new string[256];  using (TfsTeamProjectCollection collection = new TfsTeamProjectCollection(new Uri(tfsUri)))  {  testManagementService = collection.GetService<ITestManagementService>();  testControllers = testManagementService.TestControllers.Query();  foreach (var testController in testControllers)  {  i = i + 1;  Console.Out.Write(i);  Console.Out.Write(" ");  Console.Out.Write(testController.Name);  Console.Out.WriteLine();  listController[i - 1] = testController.Name;  }  // Select the controller which you want to manipulate  // So from the list, select the number 1, 2, or..  Console.Out.WriteLine("Select the controller you want to manipulate properties for..(select the number above)");  selectedController = Int32.Parse(Console.ReadLine());  Console.Out.WriteLine(listController[selectedController - 1]);  Console.Out.WriteLine("Select any of the below for manipulating the selected controller");  Console.Out.WriteLine("1 Register");  Console.Out.WriteLine("2 Unregister");  Console.Out.WriteLine("3 Update");  int propertyManipulate = Int32.Parse(Console.ReadLine());  switch (propertyManipulate)  {  case 2:  Console.WriteLine("Sorry..Not in the scope of current sample, will be implemented later");  break;  case 3:  Console.WriteLine("Sorry..Not in the scope of current sample, will be implemented later");  break;  case 1:  string controllerName = Console.ReadLine();  ITestManagementService testManagementService1;  ITestController testControllers2;  using (TfsTeamProjectCollection collection1 = new TfsTeamProjectCollection(new Uri(tfsUri)))  {  testManagementService1 = collection1.GetService<ITestManagementService>();  //testControllers1 = testManagementService1.TestControllers.Query();  testControllers2 = testManagementService1.TestControllers.Create();  testControllers2.Name = controllerName;  List<ITestController> icollection = new List<ITestController>();  icollection.Add(testControllers2);  testManagementService1.TestControllers.Register(icollection);  }  break;  }  }  }  catch (Exception e)  {  Console.WriteLine("Error while performing the operation: " + e.Message);  }  }  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

## More Information

Managing Test Controllers and Test Agents with Visual Studio

<http://msdn.microsoft.com/en-us/library/dd695837.aspx>

**ITestManagementService** Interface

<http://msdn.microsoft.com/en-us/library/microsoft.teamfoundation.testmanagement.client.itestmanagementservice.aspx>

**TfsTeamProjectCollection** Class

<http://msdn.microsoft.com/en-us/library/microsoft.teamfoundation.client.tfsteamprojectcollection.aspx>