MS-Test using C#  
  
INTRODUCTION TO TESTING

## Objective

The objectives of this practical session are to get you started on unit testing with JUnit.

## Overview

In this lab, we are going to use JUnit to create unit tests.

You'll test a method that checks if userID and password parameters are valid by following a set of rules.

## Working with Security class

1. Open the **Labs** Console project
2. Add a new class to this project called **Security**
3. Add the following code to the Security class

public class Security

{

public class Security

{

public bool Logon(string userId, string password)

{

string errorMessage;

// userId must not be null or spaces. Must be 8 chars minimum.

// password must contain a digit and an upper case char

if (userId == null || userId.Trim() == "")

{

errorMessage = "UserID may not be null or empty";

return false;

}

else if (password == null || password.Trim() == "")

{

errorMessage = "Password may not be null or empty";

return false;

}

else if (!ValidatePassword(password))

{

errorMessage = "Password must be min 8 chars with an

uppercase and a number";

return false;

}

else

{

errorMessage = "";

return true;

}

}

public bool ValidatePassword(String password)

{

bool hasUpper = false, hasDigit = false;

if (password.Length < 8)

return false;

foreach (char c in password)

{

if (Char.IsUpper(c))

{

hasUpper = true;

// continue loop as no character can be both!

continue;

}

if (Char.IsDigit(c))

{

hasDigit = true;

}

}

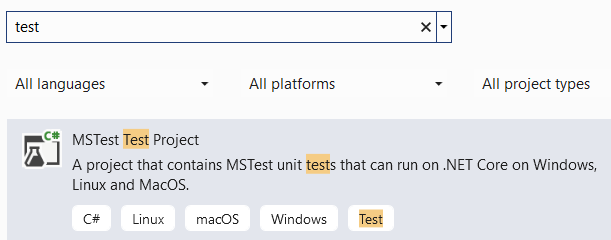
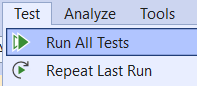
return hasUpper && hasDigit;

}

}

}

You will now test the above Logon method against a set of rules which are mentioned as comments in code.

1. First you will create a test case class in a MS-Test project called **LabsTests**.
2. Add a new MS-Test project to the Labs Solution by right-mouse clicking on the solution and then by selecting the below options:  
     
   
3. In the new MS-Test project, set a dependency (reference) to the Labs Console application.
4. Tip: Right-mouse click on the Dependencies node and select the Labs checkbox.
5. You need an example of what a test method looks like before producing your own test methods.
6. Please see the code on the next page for an example.
7. Please use this code to create a few more test methods with suitable names. Base your tests on the checks made in code. For example Password should not be blank or Null and have a minimum of 8 characters, etc.
8. Run your tests by selecting the Tests > Run All tests menus  
   
9. Replace the test case in this class with the following:

using Labs;

namespace LabsTests

{

[TestClass]

public class SecurityTests

{

[TestMethod]

public void TestLogonEmptyUserId()

{

String userId = "";

String password = "Freddy99";

Security s = new Security();

bool actual = s.logon(userId, password);

bool expected = false;

Assert.AreEqual(expected, actual);

// assertFalse(actual); // can also use this assert

}

}

}

**\*\* End \*\***