

M-eux Test

Test Automation on Windows CE 5

Getting Started Guide

Abstract

This Getting Started Guide describes how to install, configure and use M-eux Test for testing Windows CE 5 applications. This document is intended for Quality Assurance (QA) engineers and testers who wish to get acquainted with the functionalities of M-eux Test.

Part Number MT-GS-CE5
Revision 3772, March 2014



© Copyright 2013 Jamo Solutions N.V. No parts of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Jamo Solutions.

This document is provided for informational purposes only. Jamo Solutions makes no warranties as to the information in this document. The information contained herein is subject to change without notice.

All trademarks are the properties of their respective owners

Contents

Contents.....	3
Chapter 1: Introduction	4
1.1. M-eux Test overview.....	4
1.2. How to use this document	5
1.3. Where to go from here	5
Chapter 2: Get your PC ready	6
2.1. System Requirements	6
2.2. Prerequisites	6
2.2.1. Mobile Device Prerequisites	7
2.2.2. Required Privileges	7
2.2.3. Visual C++ 2010 Redistributable	7
2.2.4. .NET Framework 4.0.....	7
2.3. Installing M-eux Test.....	7
2.3.1. Preparation	7
2.4. License activation.....	11
2.5. Troubleshooting.....	15
2.5.1. The device is not able to (re)connect to the M-eux device manager.....	15
2.5.2. The device manager ports are already in use	16
Chapter 3: Getting your device ready.....	18
Chapter 4: Launching your application	20
4.1. NetCFAppLaunch Method.....	20
4.2. Redirect Method	21
Chapter 5: Your first script.....	23
5.1. Using QuickTest Professional.....	23
5.2. Using Visual Studio.....	27
5.3. Using Eclipse	33
Chapter 6: Summary	39

Chapter 1: Introduction

Welcome to the Test Automation on Windows CE 5 Getting Started guide. This document will assist you with installing, configuring and using M-eux Test. To make sure you can get started automating your test cases as soon as possible, we invite you to closely follow the instructions contained in this guide.

We have made every attempt possible in making the instructions in this guide as clear as possible. However, we recognize that we are unable to cover everything in a single guide. Should you require further assistance, please do not hesitate to visit our www.jamosolutions.com website or to contact our support team at support@jamosolutions.com.

After reading this document, you should be familiar with the following topics:

- Test automation on real Windows CE 5 devices
- Test automation on Windows CE 5 emulators
- Best practices on Windows CE 5 testing

1.1. M-eux Test overview

To test mobile applications using M-eux Test, you need three main components. The **Windows CE 5 device** or **Windows CE 5 emulator** hosts the application that you want to test. You use a **scripting environment** in which you write and execute your scripts. Finally, the **M-eux Test Device Manager** is the core of our application and acts as the gateway between the mobile devices and scripting environment.

M-eux Test currently supports the following products:

- **Windows CE 5 Devices:** Both Windows CE 5 emulators and real actual devices are supported. This connection can be established through USB via ActiveSync or Windows Mobile Device Center (Windows Vista), or the network connection of the device.

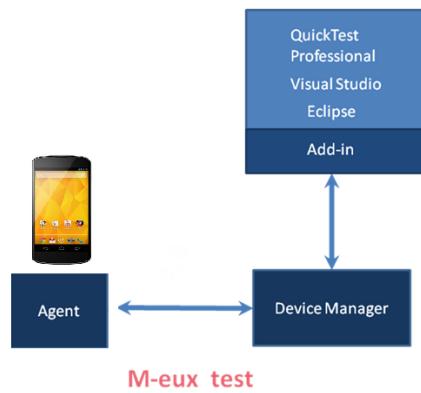


Figure 1: M-eux Test Architecture

1.2. How to use this document

This document will get you started with using M-eux Test for testing applications on Windows CE 5. To get started, here's what you need to do:

- **Get your PC ready.** You will first need to install M-eux Test on your PC.
- **Get your device or emulator ready.** Depending on your testing strategy, you may want to run your test on a physical device or an emulator. M-eux Test supports both.
- **Get your app ready.** M-eux Test captures information of your app through instrumentation. To make your app testable, you need to instrument your app.
- **Create your first script.** Once your pc, device and app are ready, you can start creating your first script.

1.3. Where to go from here

This document will get you started creating your first script. If you want to further explore the features of M-eux Test, please visit our website at www.jamosolutions.com for further information on the features of M-eux Test.

Chapter 2: Get your PC ready

2.1. System Requirements

The supported systems and requirements:

Computer/processor	IBM-PC or compatible with a Pentium III or higher (Pentium IV or higher recommended) microprocessor.
Operating System	<p>Windows Desktop</p> <ul style="list-style-type: none">• Windows XP 32-Bit Edition—Service Pack 3,• Windows Vista (32-bit edition & 64-bit Edition),• Windows 7 (32-bit Edition & 64-bit Edition)• Windows 8 (32-bit edition & 64-bit edition)• Windows 8.1 (32-bit edition & 64-bit edition) <p>Windows Server</p> <ul style="list-style-type: none">• Windows Server 2003 32-bit Edition—Service Pack 1,• Windows Server 2003 R2 (32-bit x86),• Windows Server 2008 (32-bit & 64-bit edition)• Windows Server 2008 R2• Windows Server 2012• Windows Server 2012 R2
Memory	Minimum of 2 GB.
Color Settings	Minimum of High Color (16 bit).
Free disk space	<p>1 GB of free disk space for application files and folders and an additional 1 GB of free disk space on the system disk (the disk on which the operating system is installed).</p> <p>The free disk space requirements do not include disk space required for any prerequisites that may need to be installed before installing M-eux Test.</p> <p>After M-eux Test is installed, it is recommended to have at least 1 GB free disk space on the system disk for the operating system and M-eux Test to run correctly.</p>

2.2. Prerequisites

To install M-eux Test on your computer, you first need to install the following prerequisites:

- Windows Embedded CE 5 Emulator (if using Emulator, not required when using real device).
- On the emulator the .NET Compact Framework 2.0 needs to be installed.
- Optionally, OCR components

Once all prerequisites have been installed, you can continue with the installation of M-eux Test. You may be required to execute some post-configuration tasks to ensure M-eux Test will work correctly in your environment.

2.2.1. Mobile Device Prerequisites

In case of Windows Mobile devices and emulators, ActiveSync from Microsoft is needed in order to connect the PC to the mobile device for Windows XP and Windows 2003. Windows Vista and Windows 7 require the Windows Mobile Device Center.

In case of Android devices, the SDK of Android needs to be installed together with the USB drivers from Android in order to connect the mobile device with the PC using a USB cable.

In case of BlackBerry devices, the “BlackBerry Desktop Software” needs to be installed.

For iOS devices – you can download the latest iTunes from apple website.

2.2.2. Required Privileges

You need to have local system administration rights in order to install the product.

2.2.3. Visual C++ 2010 Redistributable

M-eux Test requires the Visual C++ 2010 Redistributable to be installed on your PC. You can download the Visual C++ 2010 Redistributable from <http://www.microsoft.com/en-us/download/details.aspx?id=8328> .

2.2.4. .NET Framework 4.0

M-eux Test requires version 4.0 of the .NET Framework to be installed on your PC. You can download the .NET Framework from <http://www.microsoft.com/en-us/download/details.aspx?id=17718> .

2.3. Installing M-eux Test

In order to test applications on Android devices or Android emulators, you need to install the M-eux Device manager. As shown in the architecture, the device manager is used to communicate between the device and the test tools.

You can download the set up on our website: M-eux Test Setup at <http://www.jamosolutions.com/memberArea/downloads.php>. You will need to have a credential to log in the member area network. If you do not have the credentials yet, please email us at info@jamosolutions.com

It is mandatory to install the application with admin rights, simply right click on the **MeuxSetup.msi** and **run as admin rights**.

2.3.1. Preparation

1. First step is to execute the **MeuxSetup.msi** file from the M-eux build you downloaded on your PC. The dialog boxes will guide you through the installation.
2. Click **Next**

M-eux Test: Windows CE 5 Getting Started Guide



Figure 2 Installing M-eux Test

3. Setup will check the components which are already installed on the pc and will indicate if any additional component is required. Review all warnings and errors and click **Next** to continue the installation process.

Note: The *Android SDK check will issue a warning if you have downloaded and copied the zipped version of the SDK. No warning will be displayed if you have used the installer to install the Android SDK. Please see the instructions [here](#) for more information.*



4. If you agree with the License Agreement, select **I accept the terms in the License Agreement** and click Next.

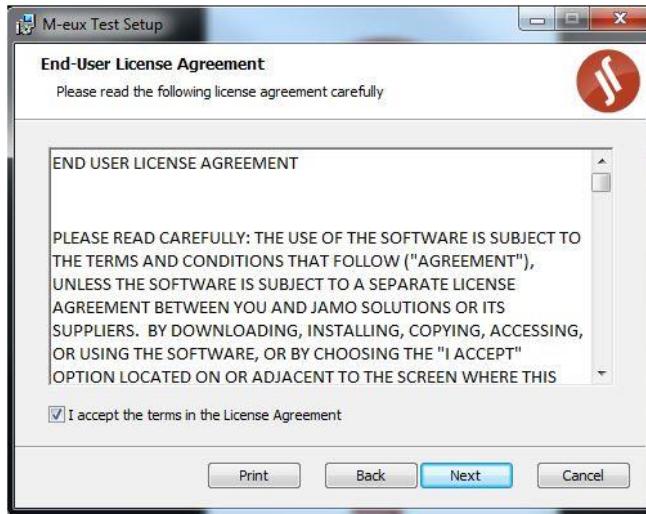


Figure 3 Accepting license

5. Select:
 - a. Unified Functional Testing (QuickTest Professional) if you want to use M-eux Test with QTP/UFT.
 - b. Select Visual Studio 2005 if you want to use M-eux Test with Visual Studio 2005 sp1.
 - c. Select Visual Studio 2008 if you want to use M-eux Test with Visual Studio 2008 sp1.
 - d. Select Visual Studio 2010 if you want to use M-eux Test with Visual Studio 2010.
 - e. Select Visual Studio 2012 if you want to use M-eux Test with Visual Studio 2012.
 - f. Select Visual Studio 2013 if you want to use M-eux Test with Visual Studio 2013.
 - g. Select Execution M-eux Visual Studio scripts if you want to execute only Visual Studio based scripts on this PC.
 - h. Select Eclipse Extension if you want to use M-eux Test with Eclipse

Note: You cannot extend Visual studio 2008 and Visual studio 2005 at the same time.

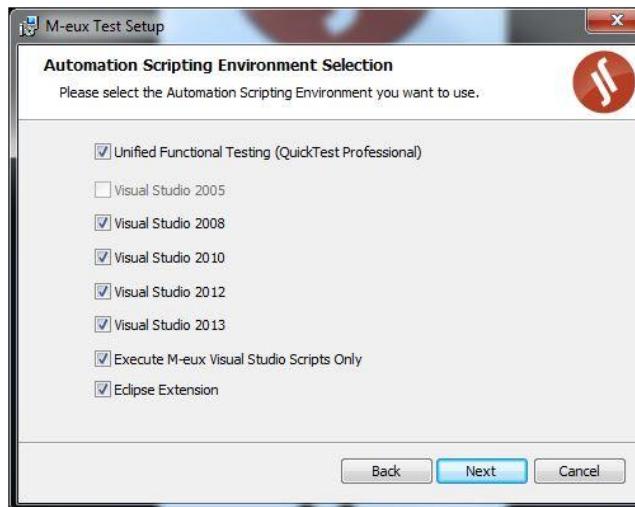
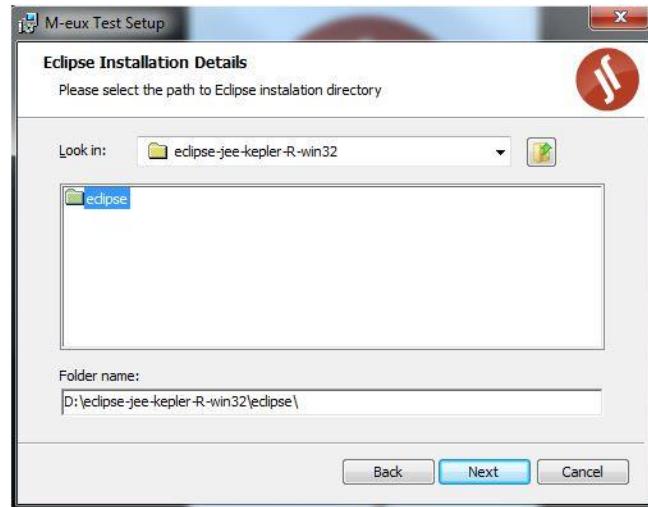
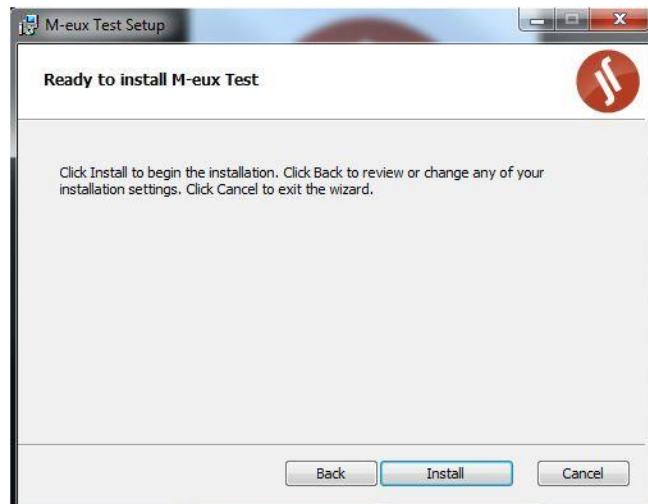


Figure 4 Choose Extensions

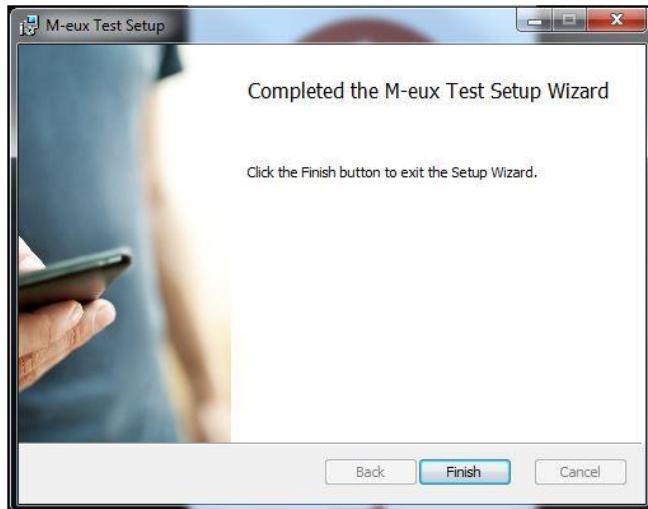
6. If you have selected the Eclipse Extension then setup will ask you to point to the directory where Eclipse is installed. Please make sure you point to a 32-bit version of Eclipse.



7. Click on "Install" to start the actual installation.



8. Wait until Setup has completed. Depending on your environment, this may take up to 10 minutes as setup is preparing your PC for smooth automation.



2.4. License activation

Start the device manager application on the PC.

When starting the first time the device manager, you get the device manager without possibility to connect devices or tools. You have to install a license first. Depending on the purchased license(s), you can choose between seat, site and token license.

Seat License activation

In case of a seat license, select the menu **Tools**, **License**, **New license**, **Seat**. Write down the locking id and send this id to Jamo Solutions: support@jamosolutions.com. Jamo solutions will return the file with the license keys. A path name to this file needs to be entered in the edit field linked to the Seat License Details section. You can write the path manually or use the 'Select button' to find and select your license key file.

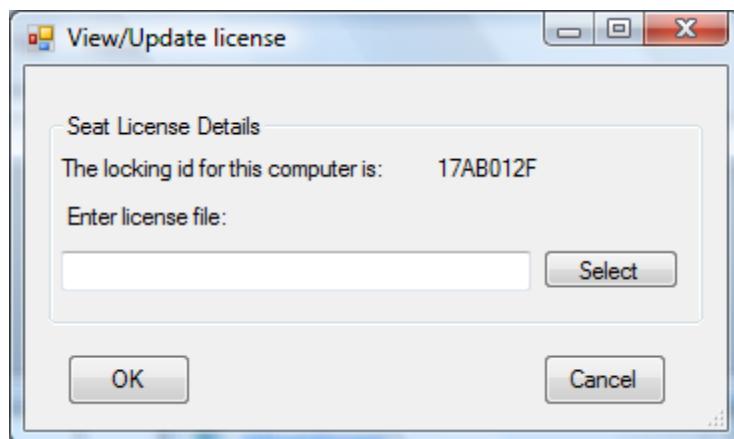


Figure 5 Seat license activation

After selecting "OK", the file will be read and the seat license will be installed.

Site license activation

In case of a site license, select the menu **Tools, License, New license, Site**. Enter the IP or hostname and port of the license server. The license server URL will be generated automatically.

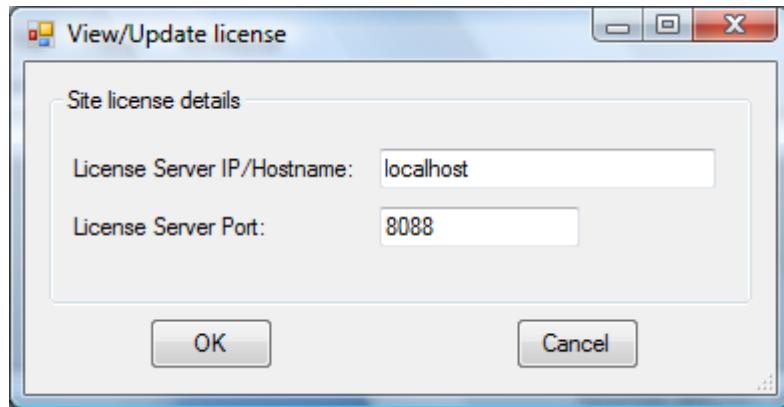


Figure 6 Site license activation

Note that prior to connecting to the license server machine, the license server needs to be installed on that machine. The licenses will be checked at run-time, so you need to be connected to the license server all the time while using the Device Manager.

If you have problems connecting to the license server, then please contact your product administrator who is responsible for the M-eux license server.

Note the site license will be released if you shut down your device manager. So if you want to transfer the site license to another tester in your company. You just need to shut down your Meux Device Manager.

Token license activation

Token licenses are distributed by your local license server. They allow you to use the tool while not being connected to the license server.

In case of a token license, select the menu **Tools, License, New license, Token**. In the details enter the IP or hostname and port of the license server. The license server URL will be generated automatically.

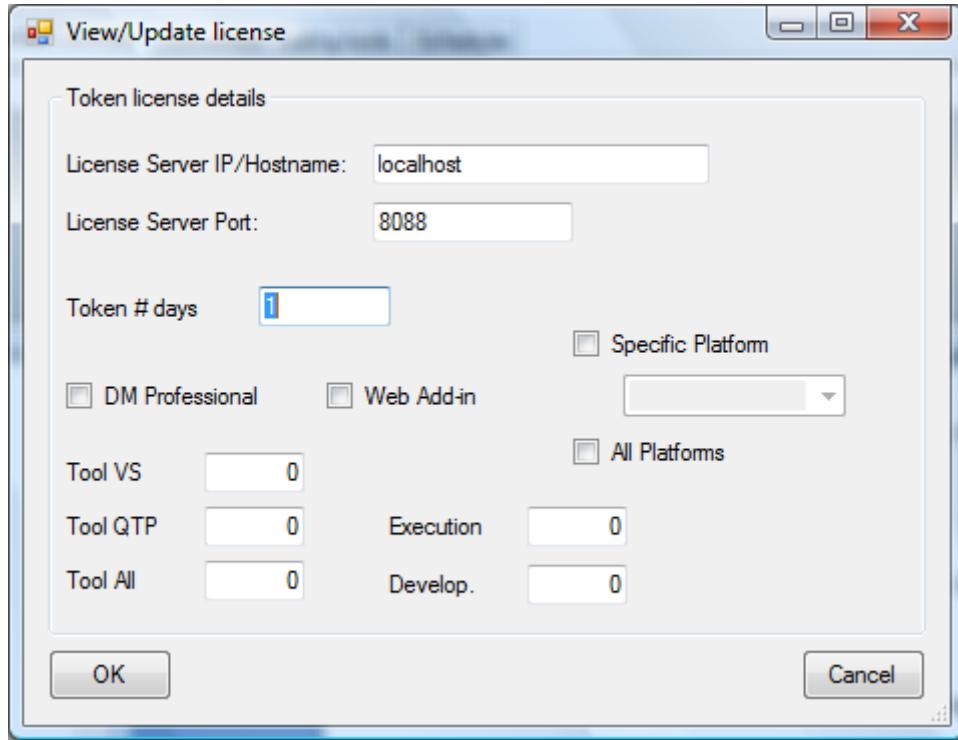


Figure 7 Token license activation

Note that prior to connecting to the license server machine, the license server needs to be installed activated and running on that machine.

- The checkbox **Web Add-in** consumes a web support for each tool demanded license.
- The **Token # days** edit field contains the number of days that the token will be used by the machine, i.e. the number of days that the token will be valid.
- The checkbox **DM Professional** denotes the license for the device manager professional edition. A license device manager professional activates the WAN Connector and local scheduler functionality.
- The checkboxes **Specific Platform** and **Platform All** denote the platform of your connected devices. Only one of these checkboxes can be checked.
- If you check **Specific platform** you have to select in the combo box under the platform you wish. If you select nothing, the Windows Mobile/CE platform will be levied. Selecting **Windows Mobile/CE** denotes that Mobile-based or Windows CE-based devices can connect to the Device Manager directly or indirectly through the WAN connector. Selecting **Android** denotes that android-based devices can connect to the Device Manager directly or indirectly through the WAN connector.
- Checkbox **Platform All** denotes that any supported mobile device can connect to the Device Manager directly or indirectly through the WAN connector.
- The other fields contain the number of token-licenses you want to have active for the specified time period indicating that all tokens have the same duration as specified in **Token #days**. Following table explains the tokens one can request:

Token Edit field	Description
Development	Denotes the number of development tokens. With one development token, the QA engineer can create, modify and execute a script. Normally this number is one since most of the time the QA engineer is working on one script at a time.
Execution	Denotes the number of execution tokens. With one execution token, you can execute a script. With N execution tokens, you can execute N scripts simultaneously.
Tool QTP (QuickTest Professional)	Denotes the number of QuickTest Professional tokens. This number is 0 or one since you can run only one QuickTest Professional instance at a time.
Tool ALL	Denotes the number of tool instances you can launch simultaneously. Currently Quick Test Professional and QuickTest Professional are supported.

After clicking OK, you will receive an overview of requested and received licenses. If all requested licenses are not granted, it means that they are in use by other users.

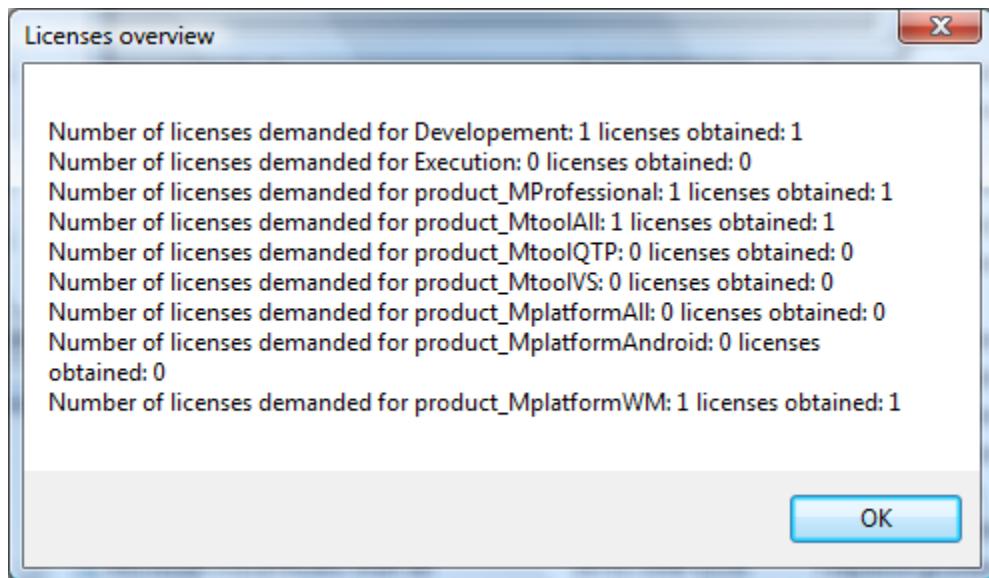


Figure 8 Token license overview

Note that for any license installation (install new license, switch to another license or release token license) all devices and tools have to be disconnected.

Handling the token license

All token licenses have an expiration date. If you want to release these tokens before the expiration date, you can do it by calling the menu item **Tools, License, Release Token License** or by installing a new license (seat, site). Before calling one of these menu items, you need to connect your machine to the network so that the license server is reachable. The menu item **Release Token License** will release all tokens and switch your license to site license connecting to the same license server address that was used to book and un-book the tokens.

Managing the professional license

The Device Manager will at startup time look to book a Professional license. You can release this license and use the Device Manager in Standard mode by selecting **Tools, Licenses, Release Device Manager Pro**. Performing this action will disconnect all connected WAN connectors and disable the scheduler.

If the Device Manager is in Standard mode, you can book a Professional license by selecting the menu **Tools, Licenses, Get Device Manager Pro**.

You can verify the mode the Device Manager is in by inspecting the menu entry in **Tools, Licenses**. The menu entry **Release Device Manager Pro** indicates that the Device Manager is in Professional mode. The menu entry **Get Device Manager Pro** indicates that the Device Manager is in Standard mode.

License overview

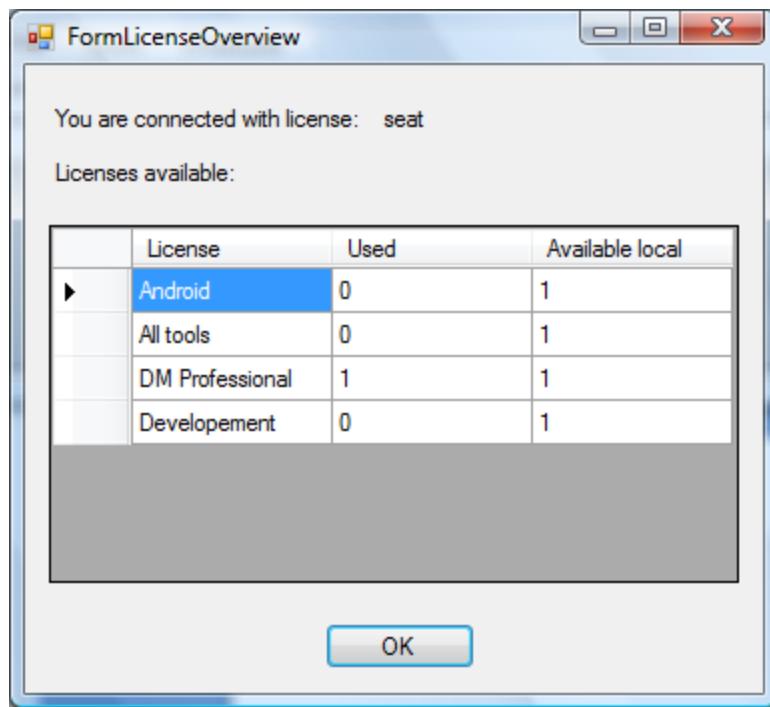


Figure 9 License overview

At each time you can check your available licenses in menu **Tools, Licenses, License Overview**. The displayed form shows the number of licenses in use and if using a seat or token license, the number of licenses available locally on your PC.

2.5. Troubleshooting

2.5.1. The device is not able to (re)connect to the M-eux device manager

The communication between the PC and the mobile device is using the TCP/IP network protocol. A local firewall installed on the PC may block the communication between the PC and the mobile device. The TCP/IP protocol uses **ports** and **IP addresses** to establish communications.

We recommend you allow communication on the following IP-address blocks:

- 169.254.*.* (autoconfiguration IP addresses)
- 192.168.*.* (private IP range)
- All other IP ranges that are used for establishing connections between the mobile device and the desktop (e.g. Wi-Fi, Bluetooth, ...)

And on the following ports:

- 4444, 555, 5580-5590

2.5.2. The device manager ports are already in use

Since the communication between the Device Manager and the script (and vice versa) is using .Net Remoting technology, ports are needed for communication. The ports that are used by the M-eux Test tool can be modified in “MeuxDeviceManager.exe.config” (see the bin folder of the installation directory).

After launching Device Manager or Wan Connector you get message(s) that a port is already used on this machine

You have to go to the configuration file of Device Manager (Wan Connector) and change the specified port there. For details see the User guide, section Wan Connector configuration.

1. Make sure your local firewall does not block the communication of the device manager, you can enable the communication by: Control panel—Firewall—Allowed programs and make sure you have device manager listed in there and it can communicate through firewall in both home and public network.

M-eux Test: Windows CE 5 Getting Started Guide

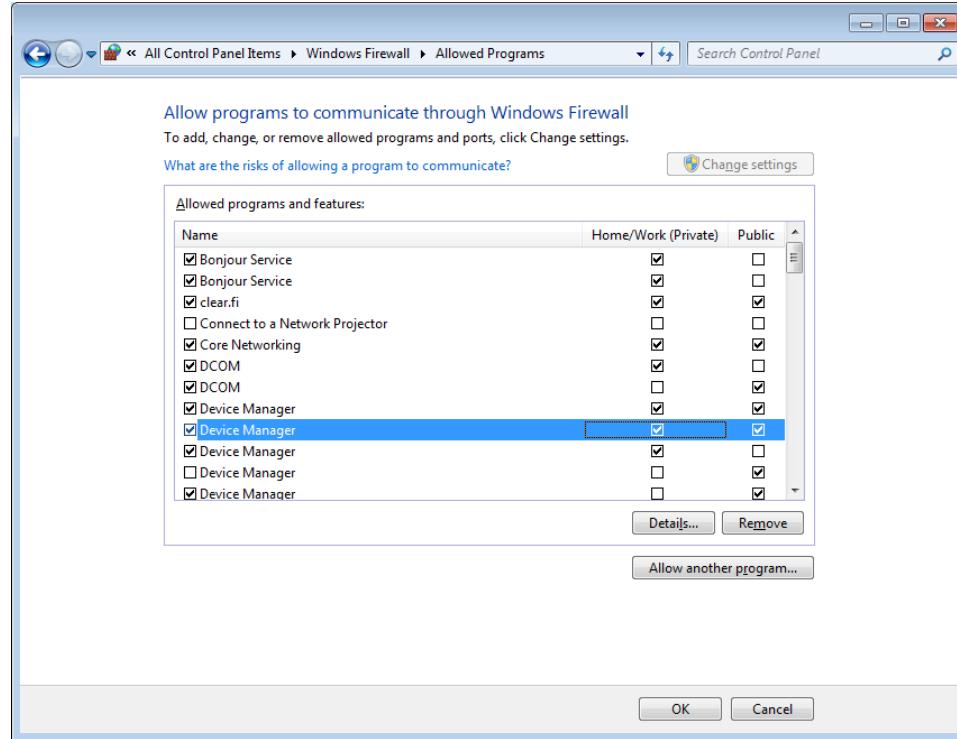
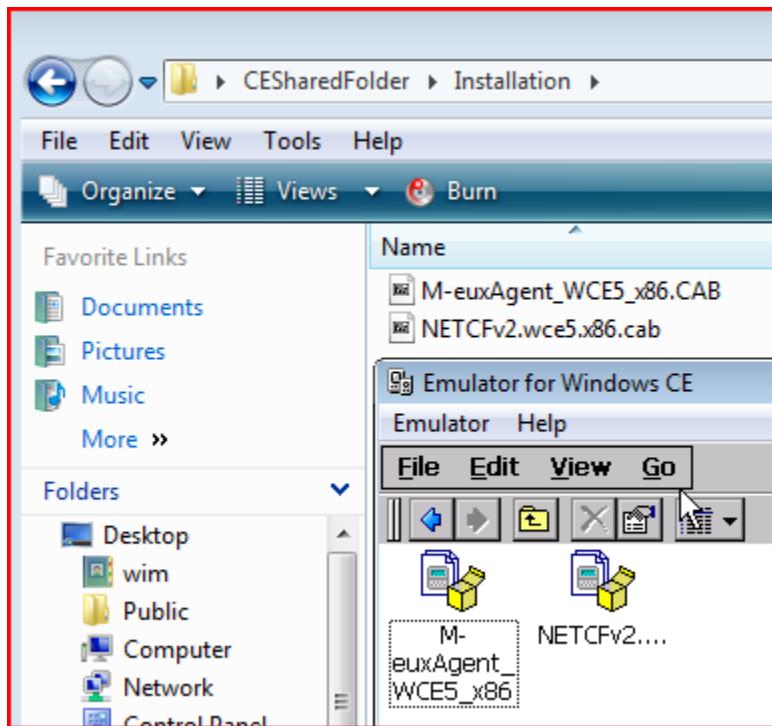


Figure 10 Allow the communication of Device Manager on firewall

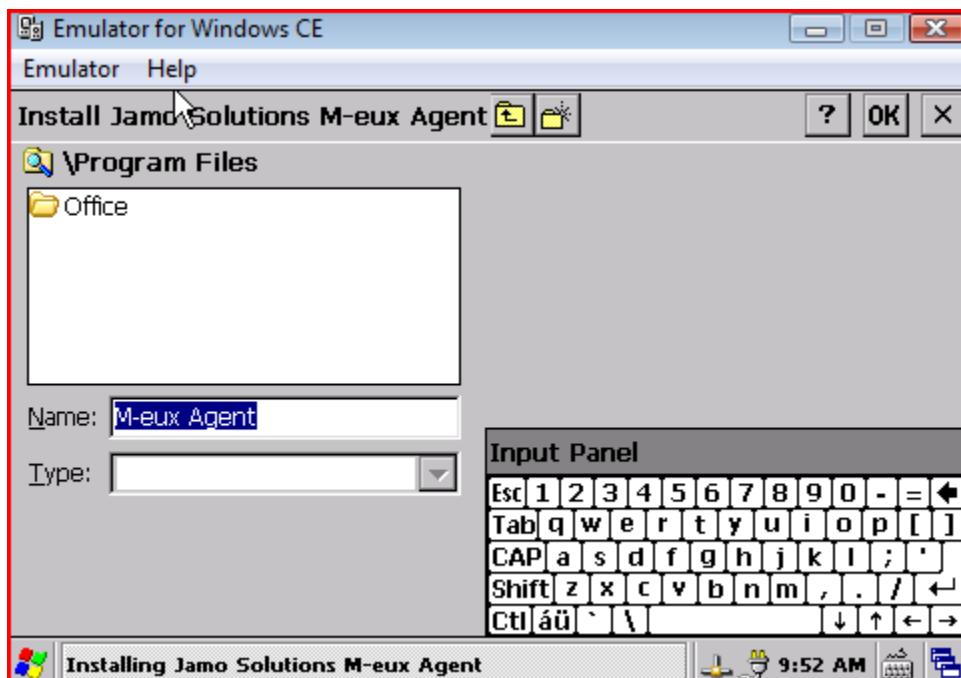
If the Device Manager program does not appear in the above window, you can click the button **Allow another program** in the above picture and point to **MeuxDeviceManager.exe** in the bin directory of the M-eux Test.

Chapter 3: Getting your device ready

1. Place the appropriate *.cab file into the Shared Folder.

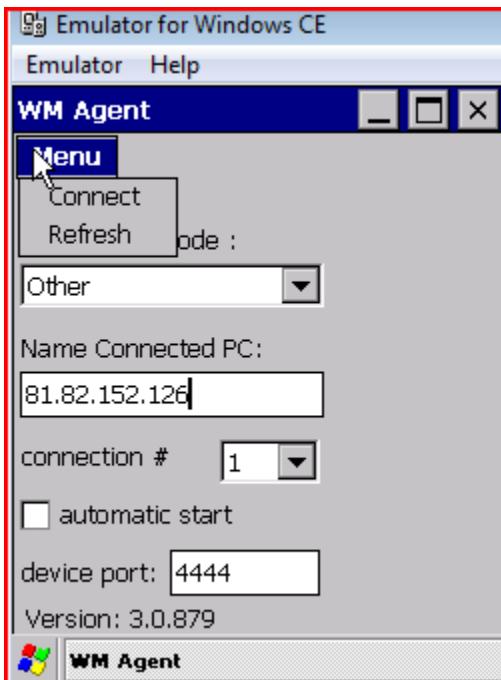


2. Execute the file.

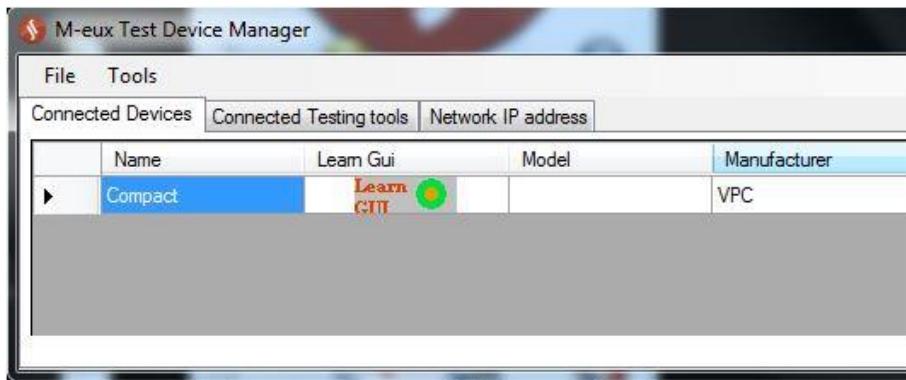


M-eux Test: Windows CE 5 Getting Started Guide

3. Restart the emulator using a soft reset.
4. Start the agent by selecting the agent entry in the start menu, sub menu program files.
5. Enter the IP-address of the PC you want to connect to in the “Name Connected PC” field.
6. Select “Connect” from the menu.



7. Make sure the device shows up in the device-manager on the PC



Chapter 4: Launching your application

Tip: This procedure is only required for applications using the Microsoft .NET Compact Framework (CF).

If the application you want to test is a .NET Compact Framework (CF) application, you need to enable our support for .NET CF applications.

The .NET CF support can only be enabled if the application under test is launched by M-eux Test. There are two ways to launch the application under test:

1. You can launch the app under test from the script with the command NetCFApplaunch or one of its variants.
2. You can redirect the executable and launch the app in the normal way (by the icon or the AppLaunch method from the script).

Note that if you choose for option 1, the application cannot be started prior to the execution of NetCFApplaunch in the script. If your application starts up automatically while booting the device, option 2 must be used.

4.1. NetCFApplaunch Method

The NetCFApplaunch has many variants. You need to know whether:

- Your app runs on .NetCF version 2 or version 3.5, and
- Whether your app is using Resco Components and if so, the version of Resco.

The variants are:

METHOD	USAGE	M-EUX CALLING EXECUTABLE
NETCFAPPLAUNCH	Use this command for .NetCF version 2.0 applications.	ManagedWrapper.exe
NETCFAPPLAUNCH35	Use this command for .NetCF version 3.5 applications.	ManagedWrapper35.exe
NETCFRESCOAPPLAUNCH	Use this command for .NetCF version 2.0 applications using Resco 6.11	RescoManagedWrapper.exe
NETCFRESCOAPPLAUNCH35	Use this command for .NetCF version 3.5 applications using Resco 6.11	RescoManagedWrapper35.exe
NETCFRESCO612APPLAUNCH	Use this command for .NetCF version 2.0	Resco612ManagedWrapper.exe

	applications using Resco 6.12
NETCFRESCO612APPLAUNCH35	Use this command for Resco612ManagedWrapper35.exe .NetCF version 3.5 applications using Resco 6.12
NETCFRESCO2005APPLAUNCH	Use this command for Resco2005ManagedWrapper.exe .NetCF version 2.0 applications using Resco 2005

The commands have 3 parameters:

- Parameter 1: the name of your executable.
- Parameter 2: the full path to the directory where your executable is located.
- Parameter 3: command line arguments of the application. In most cases, no arguments are passed and you specify the empty string "".

If you execute the command, following actions take place:

- The tool copies the corresponding managedwrapper.exe from the C:\Windows directory to the directory specified as the second parameter 2.
- The managedwrapper.exe process will start up the specified application (parameter 1) inside its process

Doing so enabled our support.

This option cannot be applied in following cases:

- The application is started up at boot time of the system. Either you stop/kill the application manually before executing the NetCFApplaunch command or you apply option 2.
- The application is consulting the name of its process to access for example files. The name of the process is normally the name of the executable. By using the NetCFApplaunch command, the process name is ManagedWrapper.exe. You have to use option 2 in this case.

4.2. Redirect Method

On windows mobile/CE an application is started up by its executable. An icon or automatic startup during boot refers to this executable.

In this solution, we make that an executable of M-eux test replaces the executable of the application under test. The procedure is as follows:

1. Rename the executable of the application under test: example: your executable name is myapp.exe. Rename it to a new name, for example: myappOrig.exe
2. Copy the corresponding managedwrapper.exe to the directory of your executable (see table in option 1).
3. Rename our executable to your name. In the example rename it to myapp.exe

4. Put a file in the directory with name: myapp.exe.redirect. This file is a XML file with following content:

```
<managedWrapper>
    <program>myappOrig.exe</program>
</managedWrapper>
```

What happens when applying this redirect setup? Here is the explanation using the example data:

When clicking on the icon or when rebooting, the system will call myapp.exe. This is the M-eux application previously named managedwrapper.exe. This application will look for the myapp.exe.redirect file. Open it, read out the program name and startup myappOrig.exe

With other words, the application under test is called by the M-eux test application managedwrapper.exe

Chapter 5: Your first script

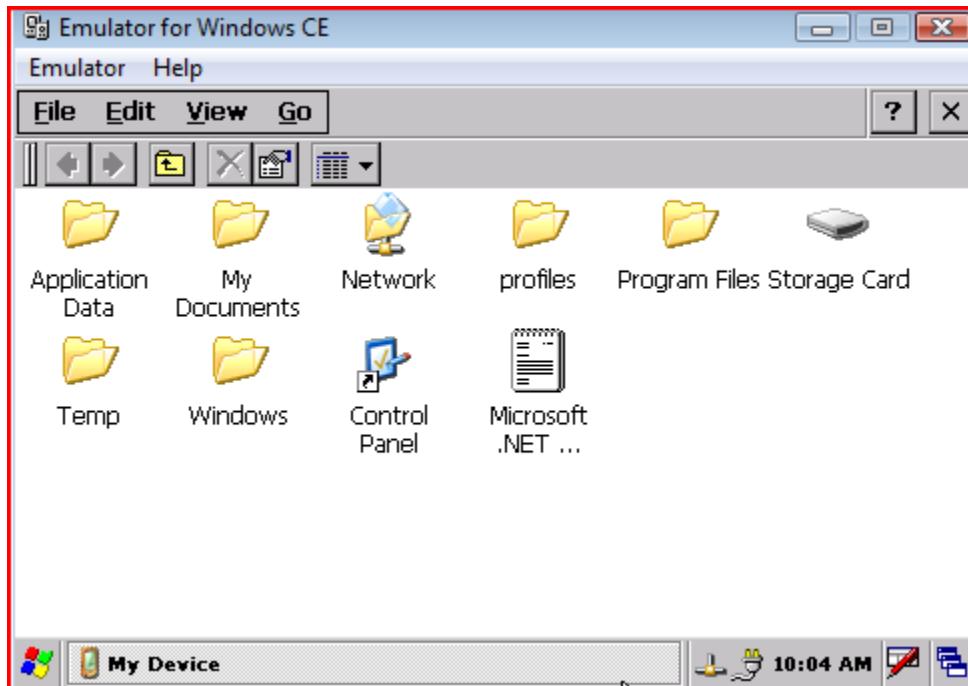
M-eux Test extends three well known environments to create your scripts: Quick Test Professional from Hewlett Packard and Quick Test Professional from Microsoft and Eclipse. Detailed information on both environments can be found in the guide: ‘User’s Guide – QTP’ or ‘User’s Guide Quick Test Professional’ or ‘User’s Guide - Eclipse’.

The following section explains the creation of the first script in QuickTest Professional. The section after that explains creation of the script in Visual Studio, and Eclipse is explained last.

5.1. Using QuickTest Professional

This section explains how to create your first script in QuickTest Professional. The test case will navigate through the file system of the device using device version of the File Explorer program. The recorded commands, as shown in QTP in the expert view, are mentioned in each step. Any command in the expert view of the script must be specified on one line! In this document, a command might be printed over several lines if the length of the command exceeds the width of the page.

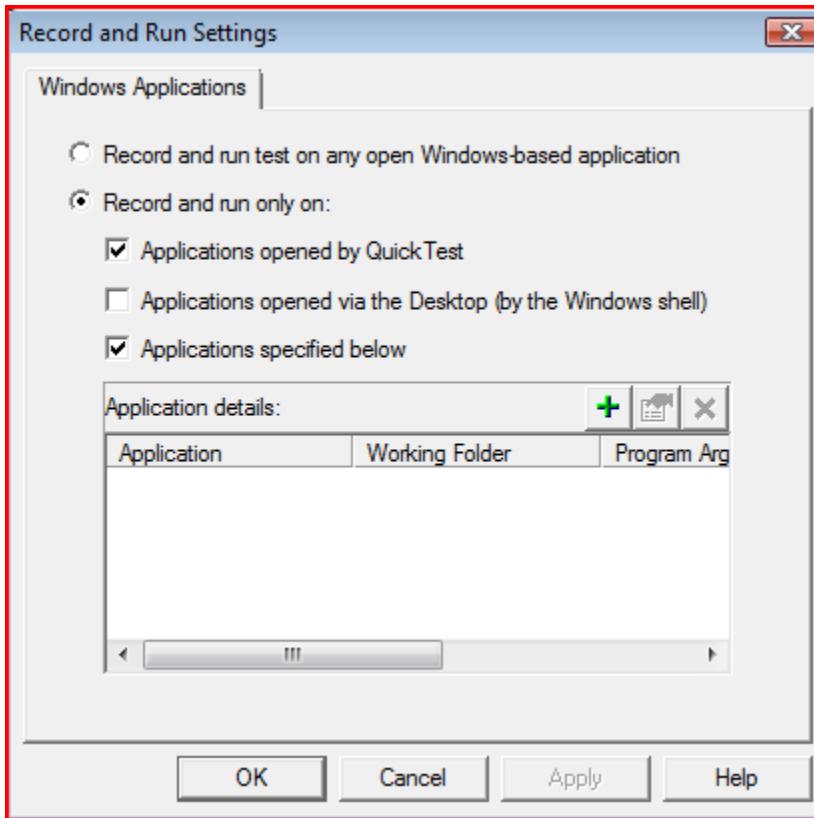
- 1) Open File Explorer and navigate to My Device.



- 2) In QTP click “Record”



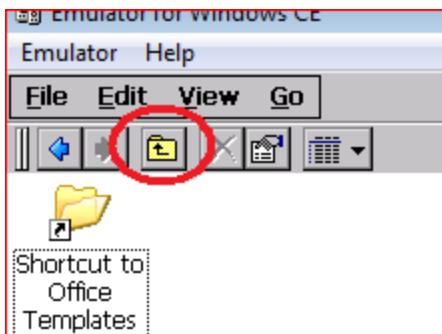
- 3) The Record and Run settings option window will appear. Select the 2nd option “Record and run only on”.



- 4) On the Emulator click on ‘My Documents’ folder. In QTP expert view this will generate following statement :

```
MobileDevice("WindowsCE").MoWindow("My Device").MoObject("Shell Embedding").MoObject("DefShellView").MoListCtrl("SysListView32").Select "My Documents"
```

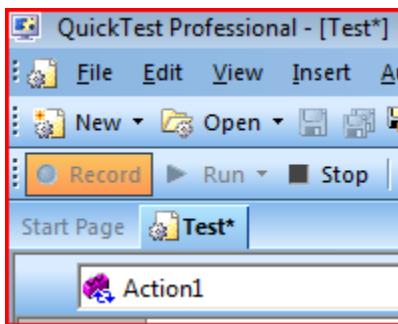
- 5) Next go back up by clicking ‘up’ button on the Toolbar.



In QTP expert view this will generate following statement:

```
MobileDevice("WindowsCE").MoWindow("My Documents").MoObject("ReBarWindow32").MoToolBar("ToolbarWindow32").Select "#3"
```

- 6) Press 'Stop' in QTP to stop recording.



- 7) In order to complete the test case, file explorer program needs to be started automatically from the script. It is a pre-condition that the file explorer is running and showing the root directory. Insert this line in the beginning of the script, above the existing lines:

```
MobileDevice("WindowsCE").AppLaunch "\windows\explorer.exe","\"",""
```

This command will make the device open File Explorer in the root directory.

In order for the script to function properly, correct name of the device should be used instead of "WindowsCE", hence you cannot copy-paste any command from this document. It is necessary to type this and all the following commands manually. QTP will automatically insert the correct name when you type MobileDevice(

- 8) Insert the next line under the one we added in the previous step:

```
itemCount = MobileDevice("WindowsCE").MoWindow("My Device").MoObject("Shell Embedding").MoObject("DefShellView").MoListCtrl("SysListView32").GetItemCount
```

The above statement should be entered on one line. In this document it is split in several lines due to formatting reasons.

This command will count the number of items in the root directory of File Explorer and store the value in the itemCount variable.

- 9) Finally, add the following line to the end of the script in order to compare the current number of items in the root directory with the value when the file explorer was first opened. This expected value was previously stored in the itemCount variable:

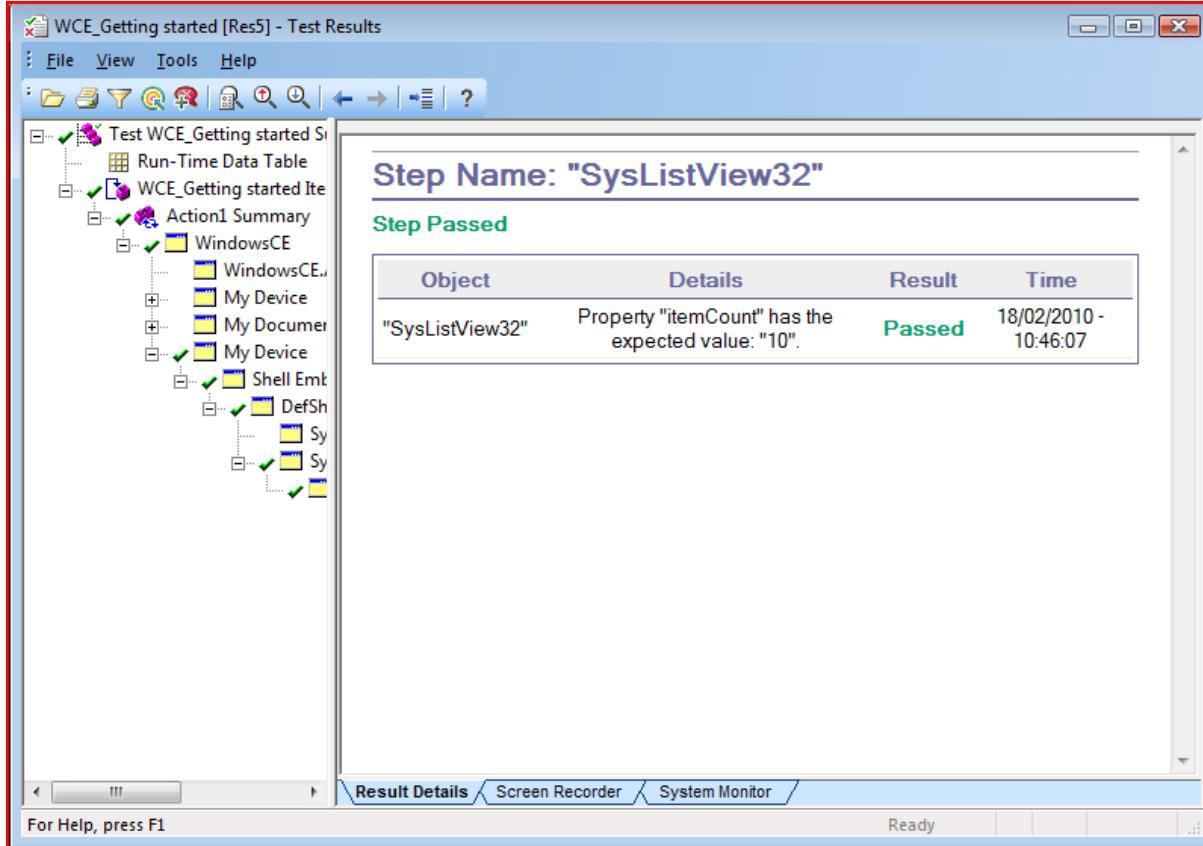
MobileDevice("WindowsCE").MoWindow("My Device").MoObject("Shell Embedding").MoObject("DefShellView").MoListCtrl("SysListView32").CheckProperty "itemCount",itemCount

The above statement should be entered on one line. In this document it is split in several lines due to formatting reasons.

Now the script is finished and should look like this:

```
MobileDevice("WindowsCE").AppLaunch "\windows\explorer.exe", "\", ""  
itemCount = MobileDevice("WindowsCE").MoWindow("My Device").MoObject("Shell Embedding").MoObject("DefShellView").MoListCtrl("SysListView32").GetItemCount  
MobileDevice("WindowsCE").MoWindow("My Device").MoObject("Shell Embedding").MoObject("DefShellView").MoListCtrl("SysListView32").Select "My Documents"  
MobileDevice("WindowsCE").MoWindow("My Documents").MoObject("ReBarWindow32").MoToolBar("ToolbarWindow32").Select "#3"  
itemCount = MobileDevice("WindowsCE").MoWindow("My Device").MoObject("Shell Embedding").MoObject("DefShellView").MoListCtrl("SysListView32").GetItemCount  
MobileDevice("WindowsCE").MoWindow("My Device").MoObject("Shell Embedding").MoObject("DefShellView").MoListCtrl("SysListView32").CheckProperty "itemCount",itemCount
```

- 10) Press “Run” in QTP. Once the script execution is finished, you will receive Test Results, where you can see the outcome of CheckProperty function.

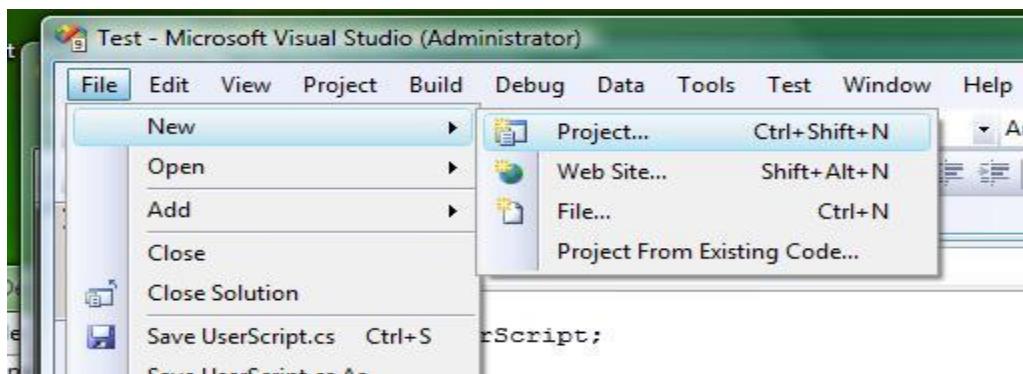


5.2. Using Visual Studio

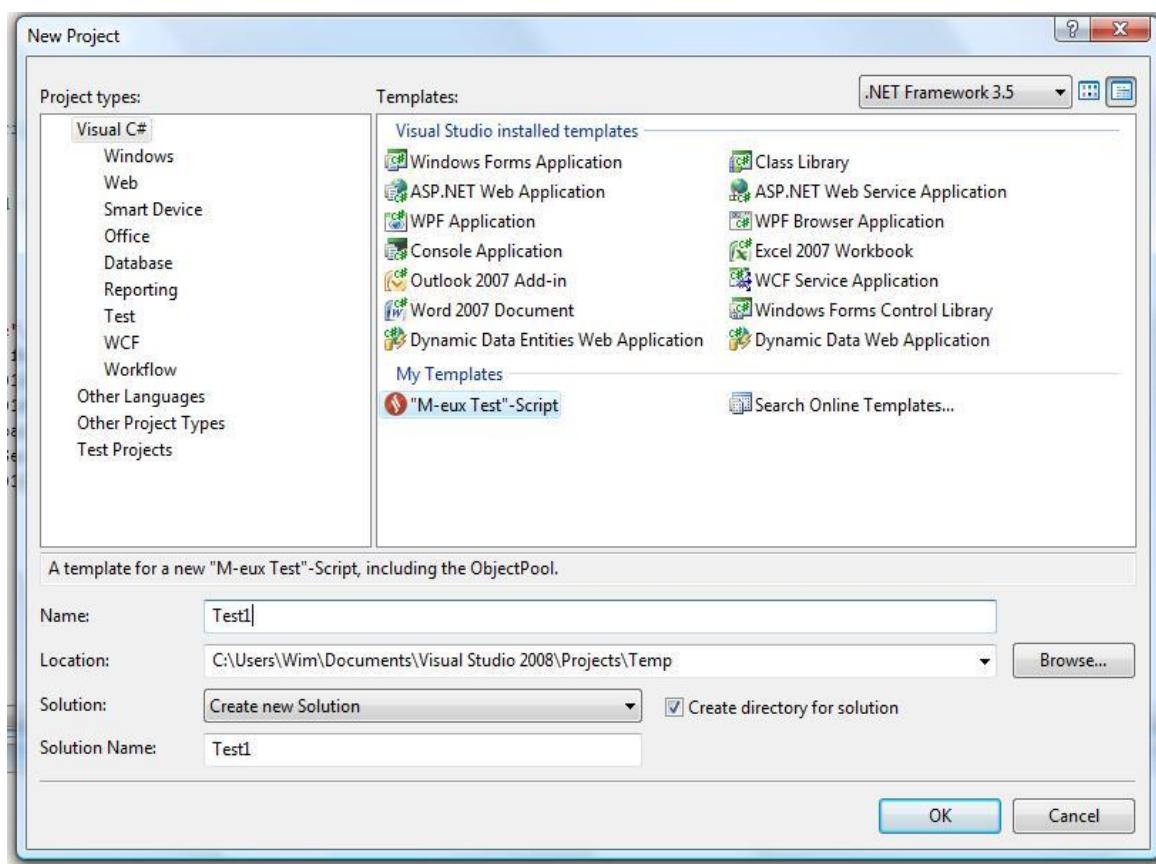
This section explains how to create your first script in Microsoft Visual Studio. The test case will navigate through the file system of the device using device version of the File Explorer program. The recorded commands are mentioned in each step. Any command in the script must be specified on one line! In this document, a command might be printed over several lines if the length of the command exceeds the width of the page.

The following script was made on an emulator. If made on a real device, another name for the device will be recorded in the script.

1. In Visual Studio click "File" -> "New" -> "Project...".

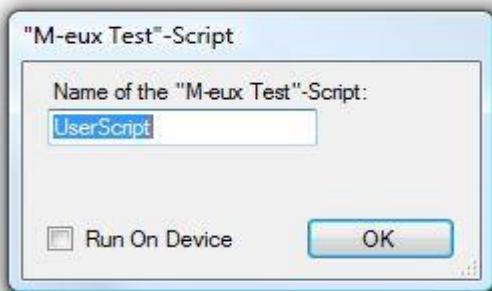


2. In the next window select "Visual C#" in the Project type's column on the left and "M-eux Test - Script" under My Templates on the right. Type in a name for your script and press "OK".

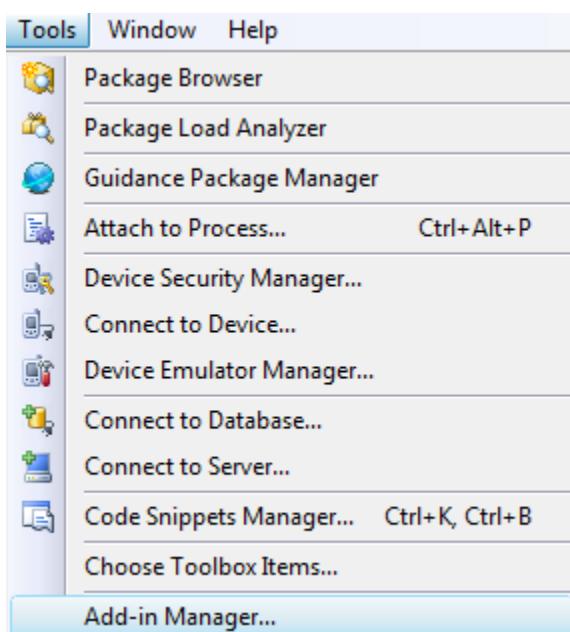


M-eux Test: Windows CE 5 Getting Started Guide

3. In the following window press “OK”.



4. Open the “Add-in manager” dialog box from the “Tools Menu”.

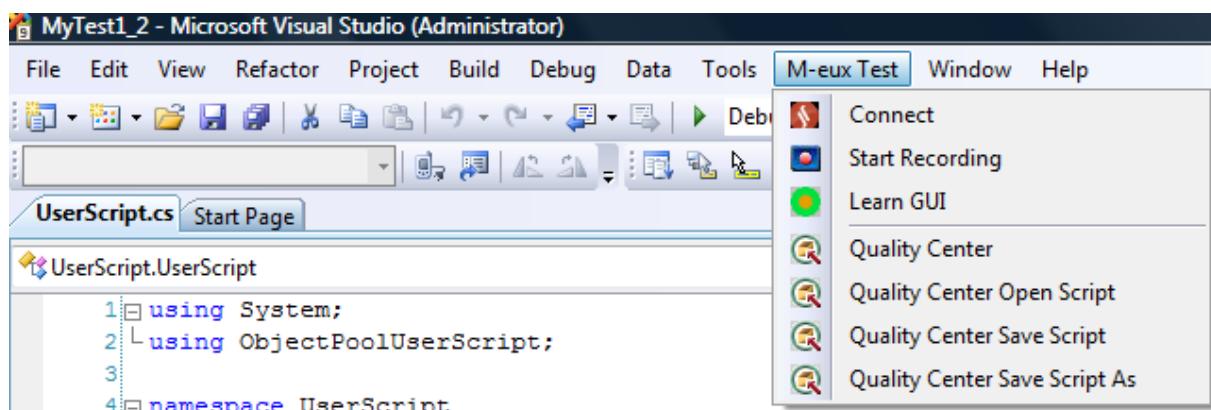


5. Select the “M-eux VS Addin” option

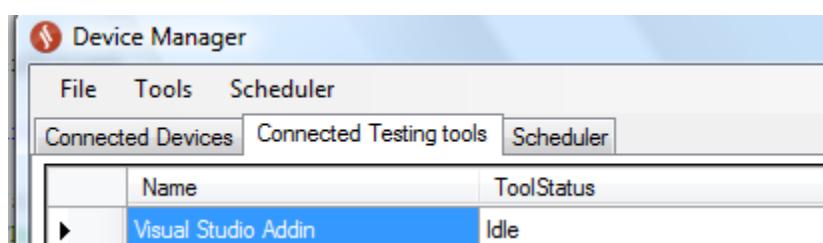


6. The “M-eux Test Menu” will be added in Visual Studio

M-eux Test: Windows CE 5 Getting Started Guide



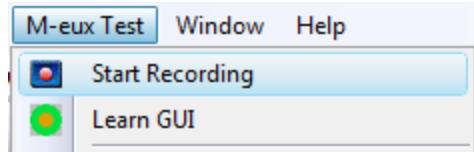
7. Select "Connect" from the "M-eux Test Menu" to connect Visual Studio to the device manager. The "Visual Studio Addin" will be listed in "Connected Testing Tools".



8. Open File Explorer and navigate to My Device.



9. Select "Start Recording" from the "M-eux Test Menu".

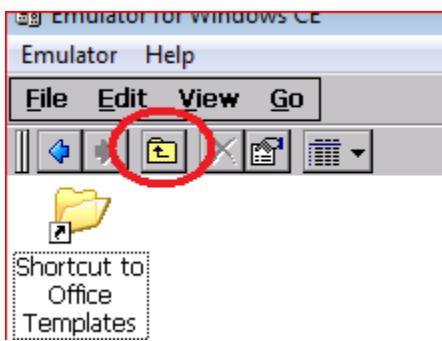


10. Place the cursor in Visual Studio behind
 // Have Fun

11. On the Emulator click on 'My Documents' folder. In Visual Studio this will generate following statement :

```
windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.Select("My  
Documents");
```

12. Next go back up by clicking 'Up' button on the toolbar.



In Visual Studio this will generate following statement:

```
windowsCE.my_Documents.reBarWindow32.toolbarWindow32.Select("#3");
```

13. Now select "Stop Recording" from the "M-eux Test Menu"



14. In order to complete the test case, the file explorer program needs to be started automatically from the script. It is a pre-condition that the file explorer is running and showing the root directory. Insert this line in the beginning of the script, above the existing lines:

```
windowsCE.AppLaunch(@"\windows\explorer.exe", @"\", "");
```

This command will make the device open File Explorer in the root directory.

15. Insert the next line under the one we added in the previous step:

```
String itemCount =  
windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.GetItemCount();
```

The above statement should be entered on one line. In this document it is split in several lines due to formatting reasons.

This command will count the number of items in the root directory of File Explorer and store the value in the itemCount variable.

16. Finally, add the following line to the end of the script in order to compare the current number of items in the root directory with the value when the file explorer was first opened. This expected value was previously stored in the itemCount variable:

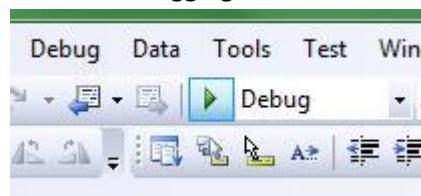
```
windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.CheckProperty(  
"itemCount", itemCount);
```

The above statement should be entered on one line. In this document it is split in several lines due to formatting reasons.

Now the script is finished and should look like this:

```
using System;  
using ObjectPoolUserScript;  
  
namespace UserScript  
{  
    public partial class UserScript : Script  
    {  
        /// <summary>  
        /// The actual script code should be programmed / recorded in this method  
        /// </summary>  
        protected override void RunCore()  
        {  
            // Have Fun  
            windowsCE.AppLaunch("\\windows\\explorer.exe", "\\", "");  
            String itemCount = windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.GetItemCount();  
            windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.Select("My Documents");  
            windowsCE.my_Documents.reBarWindow32_2.toolbarWindow32_3.Select("#3");  
            windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.CheckProperty("itemCount", itemCount);  
        }  
    }  
}
```

17. Press “Start Debugging” to run the script.



Once the script execution is finished, you will receive Test Results, where you can see the result of CheckProperty function.

Script Result Summary

```
18/02/2010 11:22:01 : Done : BEGIN RUN : UserScript, Version=1.0.3701.20459, Culture=neutral, PublicKeyToken=null
18/02/2010 11:22:03 : Done : WindowsCE.AppLaunch("windows\explorer.exe", "\", "")
18/02/2010 11:22:05 : Done : WindowsCE.My_Device.Shell_EMBEDDING.DefShellView.SysListView32.GetItemCount()
18/02/2010 11:22:08 : Done : WindowsCE.My_Device.Shell_EMBEDDING.DefShellView.SysListView32.Select("My Documents")
18/02/2010 11:22:11 : Done : WindowsCE.My_Documents.ReBarWindow32.ToolbarWindow32.Select("#3")
18/02/2010 11:22:13 : Pass : Property 'ItemCount' has the expected value: '10'.
18/02/2010 11:22:13 : Pass : Script finished successfully!
18/02/2010 11:22:13 : Done : END RUN : UserScript, Version=1.0.3701.20459, Culture=neutral, PublicKeyToken=null
```

Script result	
Status	Times
Done	6
Pass	2
Warning	0
Fail	0
Error	0

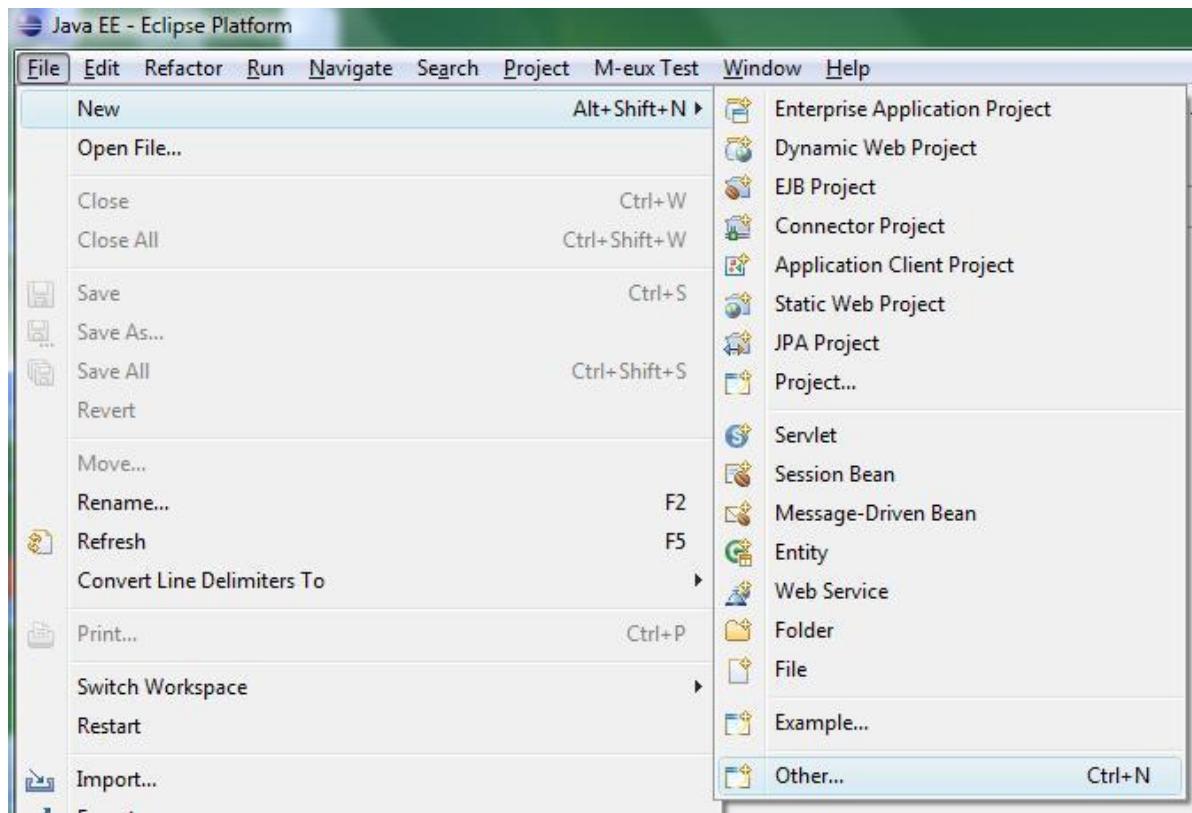
5.3. Using Eclipse

This section explains how to create your first script in Eclipse. The test case will navigate through the file system of the device using mobile version of the File Explorer program.

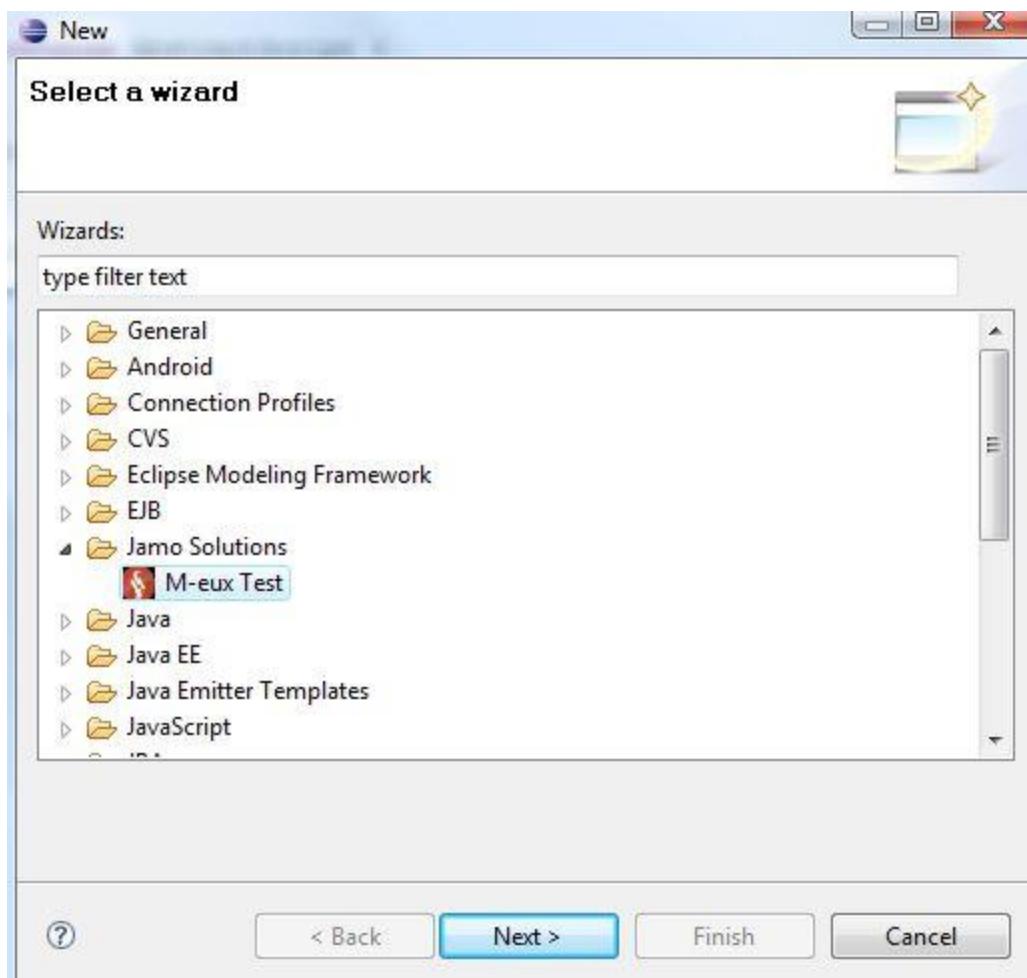
Any command in the script must be specified on one line! In this document, a command might be printed over several lines if the length of the command exceeds the width of the page.

1. In Eclipse click “File” -> “New” -> “Other...”

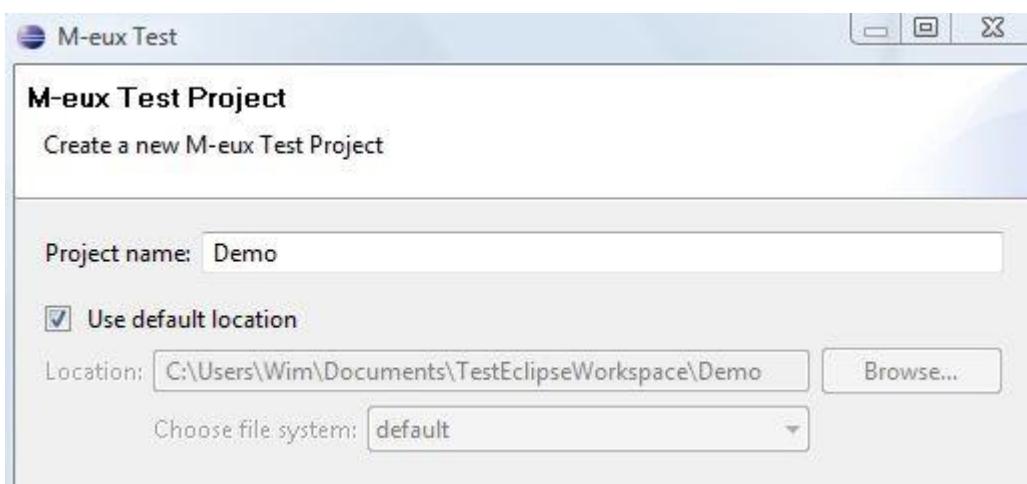
M-eux Test: Windows CE 5 Getting Started Guide



2. In the following window select “Jamo Solutions” -> “M-eux Test”



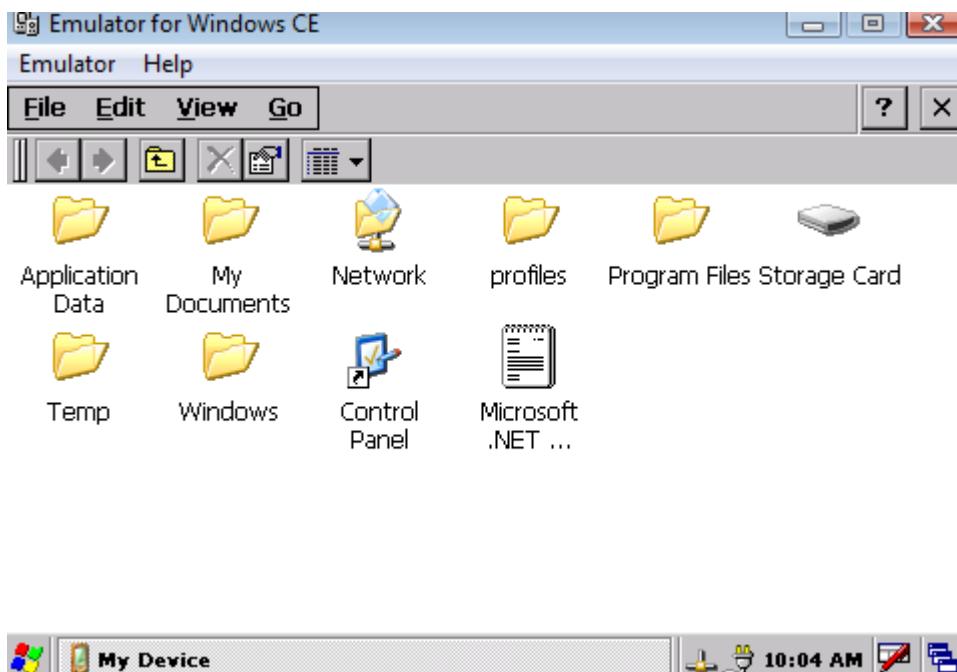
3. Specify project name and desired location



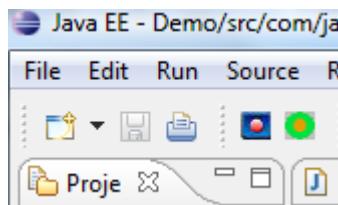
4. Press “Connect” button on the toolbar to connect Eclipse to Device Manager. The “Eclipse Addin” will be listed in “Connected Testing Tools” in Device Manager.



5. Open the File Explorer and navigate to My Device.



6. Press “Start Recording” button on the toolbar in Eclipse.



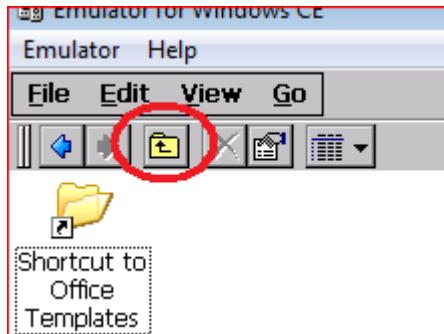
Following steps will be recorded in the body of the runCore function of UserScript.java in Eclipse until the recording is stopped. Statements will always be added at the end of the script.

7. On the Emulator click ‘My Documents’ folder.

Following statement is recorded:

```
windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.select("My Documents");
```

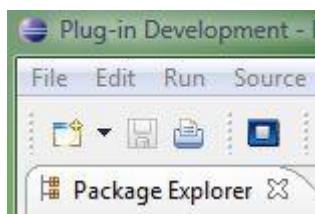
8. Click ‘Up’ button on the toolbar.



Following statement is recorded:

```
windowsCE.my_Documents.reBarWindow32.toolbarWindow32.select("#3");
```

9. Press “Stop Recording” button on the toolbar



10. In order to complete the test case, the file explorer program needs to be started automatically from the script. It is a pre-condition that the file explorer is running and showing the root directory. Insert this line in the beginning of the script, above the previously recorded lines:

```
windowsCE.appLaunch("\windows\explorer.exe", "", "");
```

This command will make the device open File Explorer in the root directory.

11. Insert the next line under the one we added in the previous step:

```
String itemCount  
windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.getItemCount();  
The above statement should be entered on one line. In this document it is split in several lines due to formatting reasons.
```

This command will count the number of items in the root directory of File Explorer and store the value in the itemCount variable.

12. Finally, add the following line to the end of the script in order to compare the current number of items in the root directory with the value when the file explorer was first opened. This expected value was previously stored in the itemCount variable:

```
windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.checkProperty("itemcount", itemCount);
```

The above statement should be entered on one line. In this document it is split in several lines due to formatting reasons.

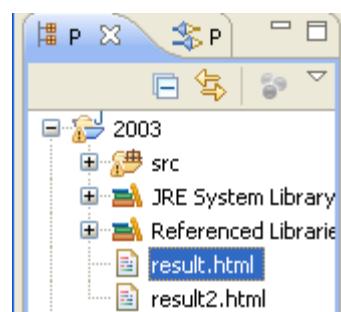
Now the script is finished and should look like this:

```
public void runCore() throws MeuxException {
    windowsCE.appLaunch("\\windows\\explorer.exe","","");
    String itemCount = windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.getItemCount();
    windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.select("My Documents");
    windowsCE.my_Documents.reBarWindow32.toolbarWindow32.select("#3");
    windowsCE.my_Device.shell_EMBEDDING.defShellView.sysListView32.checkProperty("itemcount", itemCount);
```

13. Press “Run” -> “Java Application” to run the script



You can access the result via Project Explorer on the left. You may need to refresh Project Explorer to be able to see the result.



Chapter 6: Summary

Now you should be able to start with the testing on Windows CE 5.

We wish you a good journey with your test adventure using M-eux Test tool. If you are facing any issues with our tool, or having trouble about making your test scripts for your test cases, you can always contact us at support@jamosolutions.com.