Gizmondo Installation

* Strictly Confidential *

Gizmondo Installation

Installation

1. Plug it all in

Insert the battery, connect the power adaptor, connect the USB cable from the Gizmondo to your PC. Once the Gizmondo has booted up, on your PC it should be asking for a USB driver. Point it to "CDROM\Setup\1. USB Driver\wceusbsh.sys". If it doesn't ask for a driver you can go to "CDROM\Setup\1. USB Driver\wceusbsh.inf" and right click on the file to install it manually.

2. Installing

1. The Compiler
Next we need to install the compiler, run "CDROM\Setup\2. Embedded C++\setup.exe". This will install all the components needed for Embedded Visual C++. There is a serial.txt file with the serial number needed when installing the product.

Once it has finished installing, we need to patch it up to the lastest version. Run "CDROM\Setup\3. Embedded C++ SP4\setup.exe".

Now we need to add the Gizmondo Terminal SDK so that Visual Studio knows what it's compiling for. Run "CDROM\Setup\4. Gterm\GTERM_SDK.msi".

We'll cover setting up Visual Studio to use the GTERM platform later in the document.

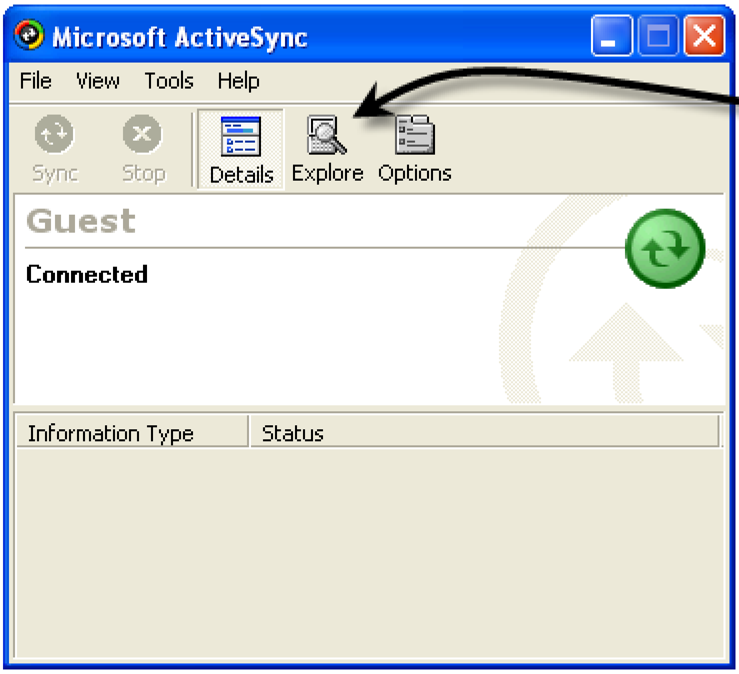
2. Active Sync
Now we need to install the "ActiveSync" client that allows us to read and write files on the Gizmondo flash disk and SD cards.

Run "CDROM\Setup\5. ActiveSync\MSASYNC.EXE".

Unplug your Gizmondo mini USB and plug it back in. Active Sync should now fire up and ask you to create a partnership.

Go ahead and create a new partnership – give your device a name, and set up what apps you wish to synchronise (none of them!).

You should now be greeted by the active sync dialog box. Click on the "Explore" icon and you're in!



Take a look on the Flash Disk!

You only need to setup the partnership once, next time you boot up the unit decline the option to create a new partnership (unless you like typing and clicking check boxes for no reason!)

3. Power Toys
Next, install the useful windows mobile power toys by right clicking on "CDROM\Setup\6. Mobile Power Toys\ WindowsMobilePowerToys.msi" and selecting install.

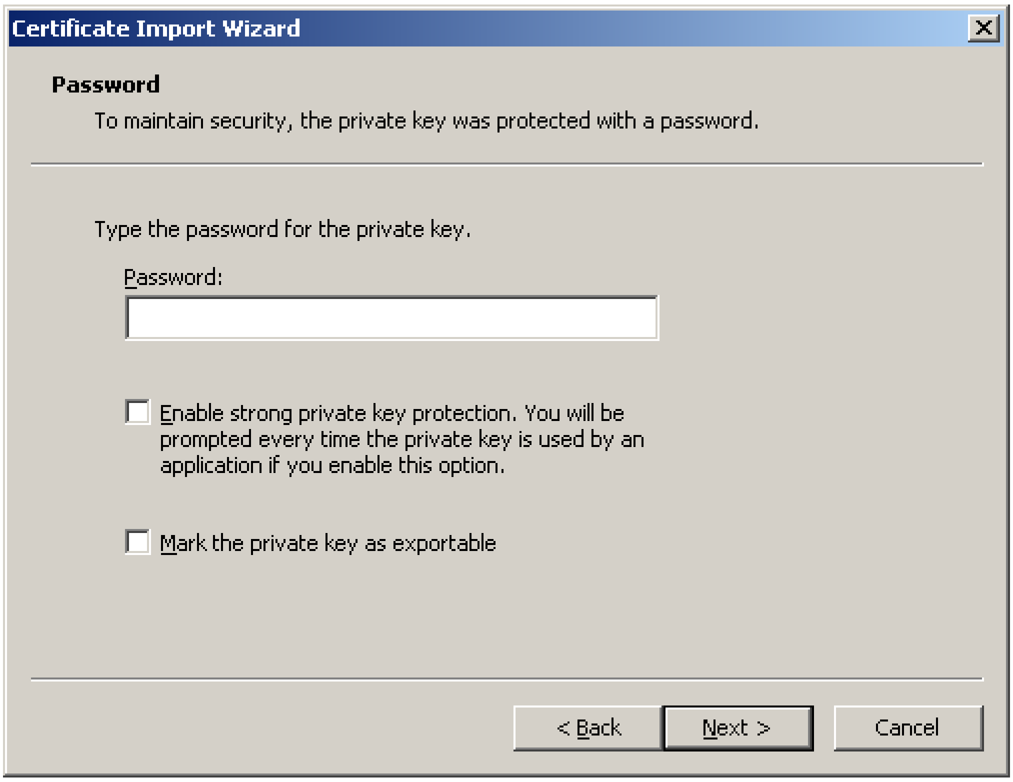
This adds some useful windows and command line tools including "rapistart" which is a command line function to remotely start up an application on the Gizmondo. Try typing "rapistart about.exe" to get the system information dialog box up on the Gizmondo screen.

The other useful tool is "Active Sync Remote Display". Launching this app will allow you to visualise whatever is on the Gizmondo screen on your PC desktop. If you happen to do any test apps that require windows button presses you get physically click them on the remote display screen and the message will be sent to the Gizmondo's window pump.

4. Signing
In order for your applications to run on a Gizmondo they have to be signed first. This will install the signing procedure onto your PC.

First copy the directory "CDROM\setup\7. Signing" to a temporary directory on your hard drive and run "install.bat" from there.

This will firstly install the "GizDevRun.pfx" file, and will present you with an installation dialog. Keep clicking the next button with the default settings until it comes up with this dialog asking you for a password.



Certificate Import Wizard

Password

To maintain security, the private key was protected with a password.

Type the password for the private key.

Password:

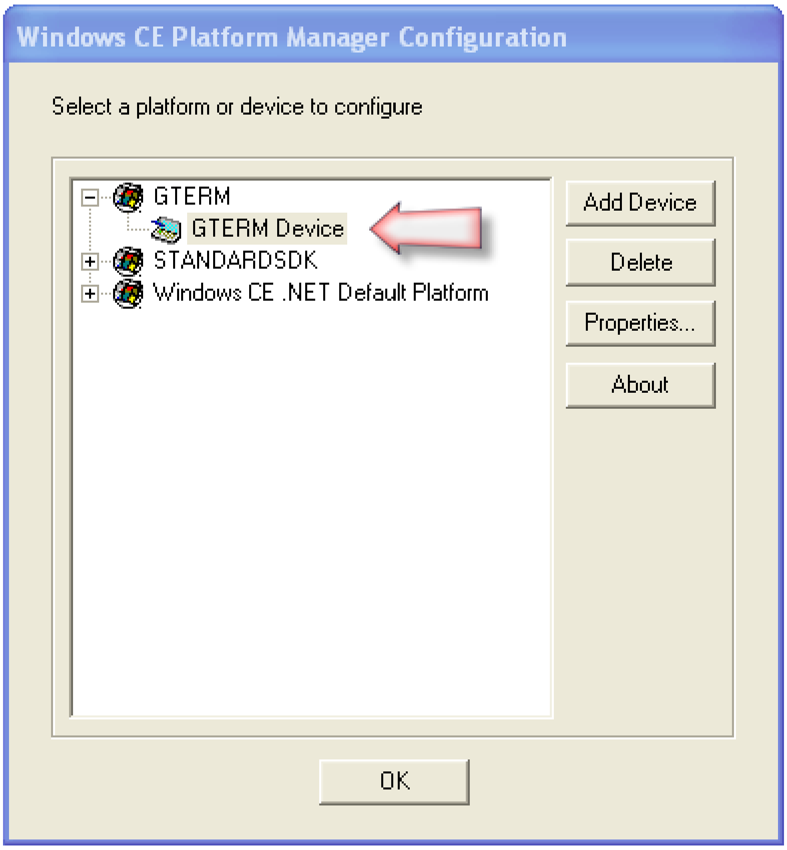
□ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

□ Mark the private key as exportable

Copy in the password from the GizDevPword.txt file, and keep clicking next again with the default settings until it’s finished. It will come up with a message box saying installation successful.
The install routine will now copy over a number of pre-signed files which are needed to allow Microsoft Embedded Studio to debug on the Gizmondo.

Setting up Visual Studio

Launch Visual Studio Embedded and go to Tools->Configure Platform Manager.

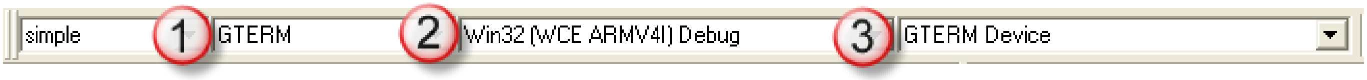

Windows CE Platform Manager Configuration
Select a platform or device to configure
GTERM
GTERM Device
STANDARDSDK
Windows CE .NET Default Platform
Add Device
Delete
Properties...
About
OK

In the list of platforms there should now be a “GTERM” tree, click on the tree and select “GTERM Device” then press OK to finish.

Now, go to the Tools->Select Remote Tools and select “Use WCE500 Tools”.

Setting up a Project

Create or load a WINce project. If the WCE Configuration bar isn’t already on your IDE, right click in some empty toolbar space and selected it. On the second IDE dropdown **(1)** menu select GTERM, on the third **(2)** dropdown select “Win32 (WCE ARMV4I…”, and on the final dropdown **(3)** select “GTERM Device”.



To sign your apps to run on the Gizmondo, you need to add a post-build event to your project settings. In the post-build tab of the project settings add a new line :-

GzSign $(TargetPath)

You can reference the application filename implicitly if required but please note any paths used in the filename much use the backslash (\) not the forward slash (/) as that confuses the signing program. The signing process will create a file called $(TargetPath).unsigned. This is a copy of the unsigned version of the file, and is required when doing a submission.

And there you have it. You will now be able to create and run applications on the Gizmondo device.