Windows Embedded Notes

Revision 1.0, 1st Sept 2011
www.touch-base.com\documentation\general

Overview | OS implementation notes | Windows CE | XP embedded | WES2009 | WES7 | Contact

Overview

Windows is available in embedded form whereby components are used to build a Windows image which is then deployed on target hardware. The main offering for Windows embedded has been Windows CE and Windows XP / 2009 embedded and more recently these are referred to as Windows Embedded Compact (Win CE) and Windows Embedded Standard (Desktop).

Although Compact and Standard form the backbone of the Windows Embedded solution there are other variations of these systems tailored to specific solutions such as POSReady (optimized for Point of Service solutions), Thin Client (tailored for Thin Client deployment) and others. We believe the appropriate drivers for Compact and Standard will work in these environments and will advise further if proven not to be the case.

When using 3rd party software in these environments a number of factors will determine the most appropriate deployment strategy. In cases where the image has been finalised and embedded on a target system then you have no other option other than to perform a post image installation or update. In cases where you are creating the image you need to determine the best way to deploy the 3rd party software so that it becomes part for the embedded image.

This document describes our recommendation for deploying the UPDD driver on the various embedded systems. The table below shows the current deployment methods available followed by detailed instructions for each OS:

Method / Operating system	Component form	First Boot Agent / OOBE stage	Lock down image update	Post image deployment
CE 5, 6 and 7 (Compact)	Yes	N/A	No	Yes – Manual process
XP embedded (Standard)	Yes	Untested / unknown	No	Yes – Setup.exe
WES 2009 (Standard)	No	Yes – Setup.exe	No	Yes – Setup.exe
WES7 (Standard)	No	Yes – Setup.exe	TBA	Yes – Setup.exe

With CE and XPe we offer individual components in a form suitable for embedding in the bootable image. However, in investigating what was required for Windows Embedded Standard 2009 and 7 we discovered a much more suitable and straight-forward method of deploying our software on embedded desktop systems during the embedding process that utilizes the standard UPDD install procedure. This has many advantageous over the individual component implementation in that it is easier to embed and uses a tried and tested installation procedure.

Unless there is a valid request to offer components for WES2009 or WES7 we do not intend to make these available for these embedded OS.

Component form

This is where software is supplied in component form suitable for embedding in the bootable image as a selectable component.

First Boot Agent / OOBE stage

Having created the initial, bootable, image from components this stage allows for the deployment of other required software prior to image lock down suitable for target system deployment.

As described in a Microsoft article: *"First Boot Agent is a set of processes that run on the Microsoft® Windows® XP/2009 Embedded runtime during its first boot. First Boot Agent (also know as FBA) reads and executes instructions placed in a special place in the registry. These instructions are populated by individual components, and can specify a number of different actions, including DLL registration, special data handing, service data installation, and other things. When FBA has finished processing everything, it cleans itself up and reboots the machine. Subsequent boots of the Windows XP/2009 Embedded runtime will not execute the FBA instructions again. FBA is a one-time operation, and usually happens on a golden master device before replication."*

Locked down image update

Under WES7 it is possible to apply updates to the locked down image that is ready to deploy on target systems without having to recreate the image with updated components. We will investigate this further if requested.

Post image deployment

Image has been locked down and deployed on target hardware. UPDD can be updated / installed with normal installation procedures given suitable disk write / registry update permissions. We are not familiar with the requirements to allow updates to a finalized image by there are articles on the web that discuss this, such as this MSDN article.

OS specific implementation notes

The following section describes the implementation of the UPDD software in various embedded environments.

CE

For CE 5, 6 and 7 we offer components and also document a post install procedure, both of which are described in a separate UPDD CE Integrators Guide.

XP Embedded

Prior to fully understanding the significance of the First Boot Agent option under this OS we created components to be built into the image using the Microsoft component designer which is described in a separate UPDD XPe Integration Guide. We have not investigated the First Boot Agent option in this older, now outdated, embedded system but similar instructions to that documented for WES2009 may work.

WES 2009

These instructions describe how to add UPDD to an existing WES2009 definition using the standard UPDD install file (setup.exe) file. Given that we are recommending using the standard installer to install the components on the image then, as a WES2009 integrator, you may be aware of the most appropriate point at which to run the setup program to deploy the UPDD software prior to image lock down that suits your image creation.

Manually running the setup.exe as normal on a newly created WES2009 system during FBA and before image lockdown is likely to work as long as the image has the necessary components to run the setup, such as the explorer shell. In our experiments with this approach we found that the setup program installed as expected but the FBA process doubled up the UPDD processes in the Task Manager which did not affect the running of the software but just looked strange. We believe this would not be an issue on the locked down image and was just a side effect of the FBA mode. The image would also need to contain the UPDD WES2009 dependencies.

Given that WES2009 can create extremely cut down images and may not contain the necessary components needed to manually run the setup.exe program we opted to define the setup.exe program as a "Run Once" function to cater for all scenarios, as documented below.

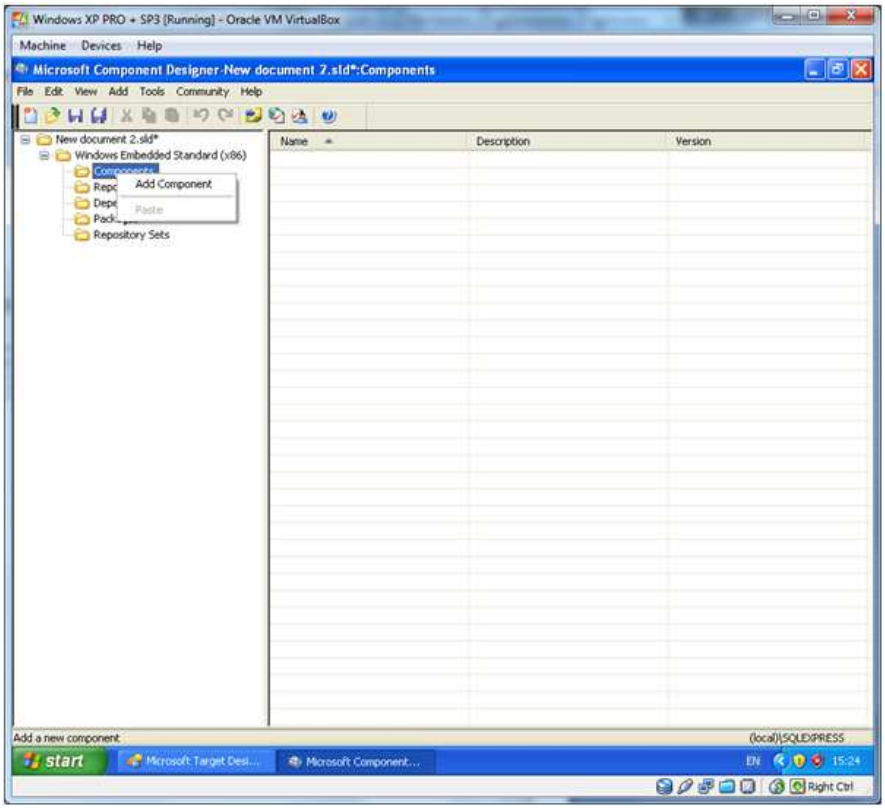
Irrespective of how the driver is deployed in the WES2009 image it is important to finalise driver settings prior to locking down the image and replication to the target hardware. This and other UPDD related issues are discussed here.

Invoking UPDD install as a Run Once function

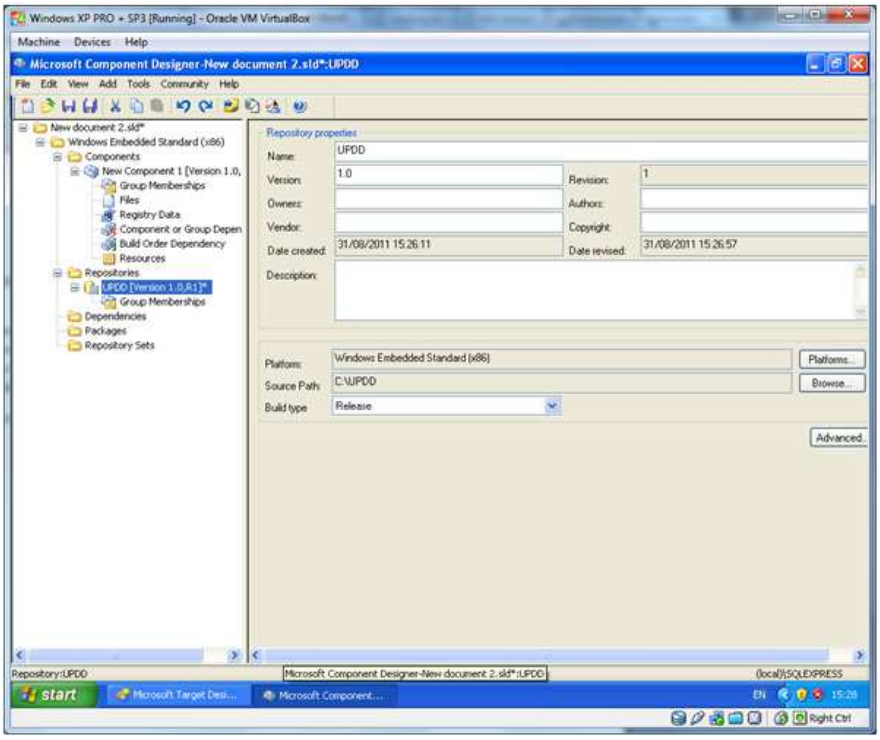
These instructions do not show how to create complete definitions or build and deploy runtime images and it assumes that you have or are creating a working definition already and simply wish to add UPDD to it. A guide for creating and deploying WES2009 images is here.

Firstly you must create a new component and add it to the definition in Target Designer:

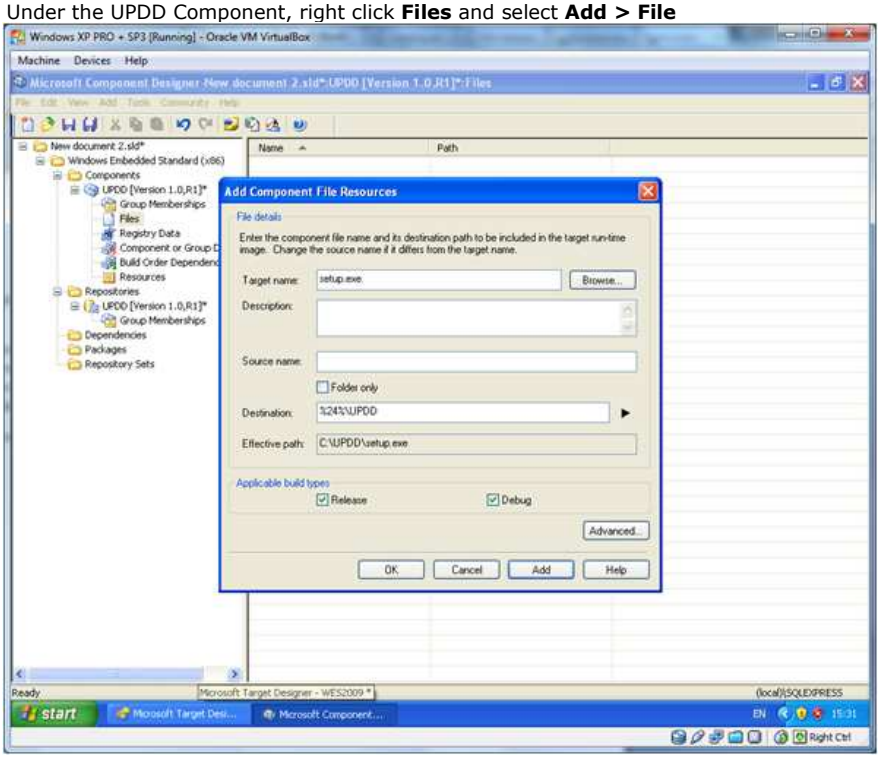
1. Create a directory on the development machine to hold the setup file (**C:\UPDD**)
2. Copy Setup.exe to that directory
3. Run Microsoft Component Designer
4. Create a new component class with **File > New**.
5. Expand 'Windows Embedded Standard'.
6. Right click on **'Components'** and select **'Add Component'**



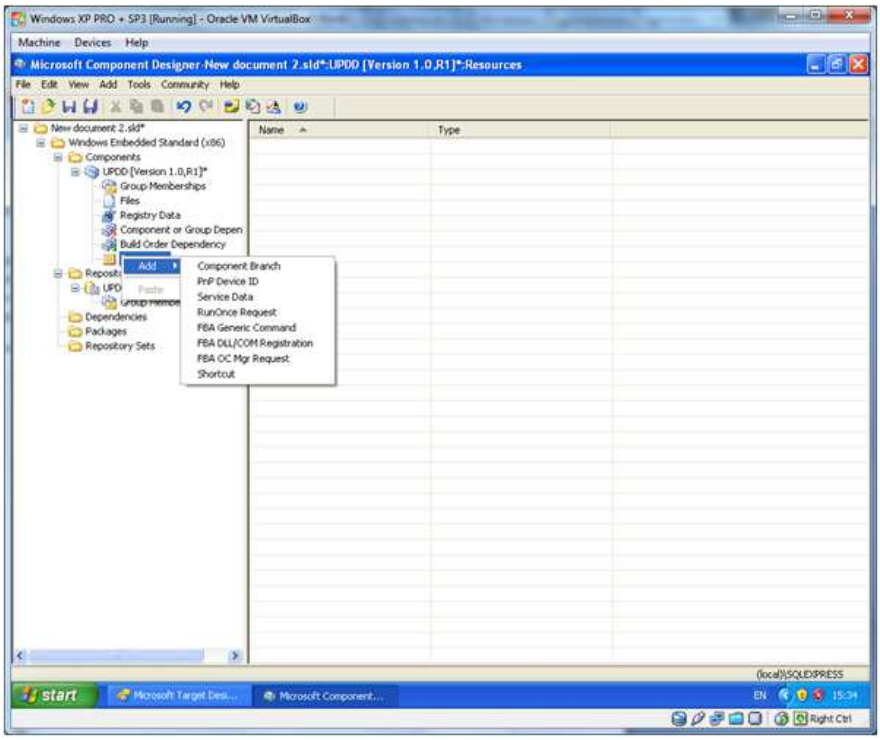
7. Right click on **'Repositories'** and select **'Add Repository'**



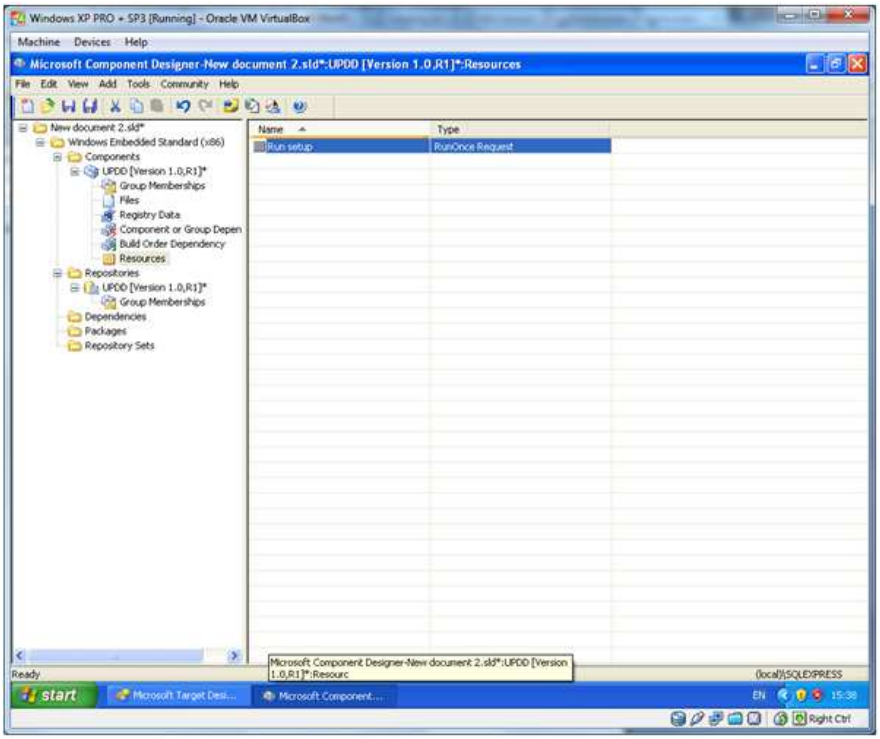
8. Select the New Repository and change the **Name** field to 'UPDD' and the Source Path to **'C:\UPDD'**
9. Select the New Component again and change the **Name** field to 'UPDD'
10. Use the '**Repositories'** button in the UPDD component properties to associate this component with the newly created UPDD repository.
11. Under the UPDD Component, right click **Files** and select **Add > File**



12. In the new dialog, browse the **Target Name** and select 'C:\UPDD\Setup.exe'
13. By default the **Destination** will be set to **'%24%\UPDD'**, which is fine (%24% is the boot drive root)
14. Click **'Add'** to create the entry, then close the 'Add Component File Resources' dialog
15. Now that the file is set to copy to the new OS, we need to run it once to install the program: Right click **'Resources'** and select **'Add > RunOnce Request'**.



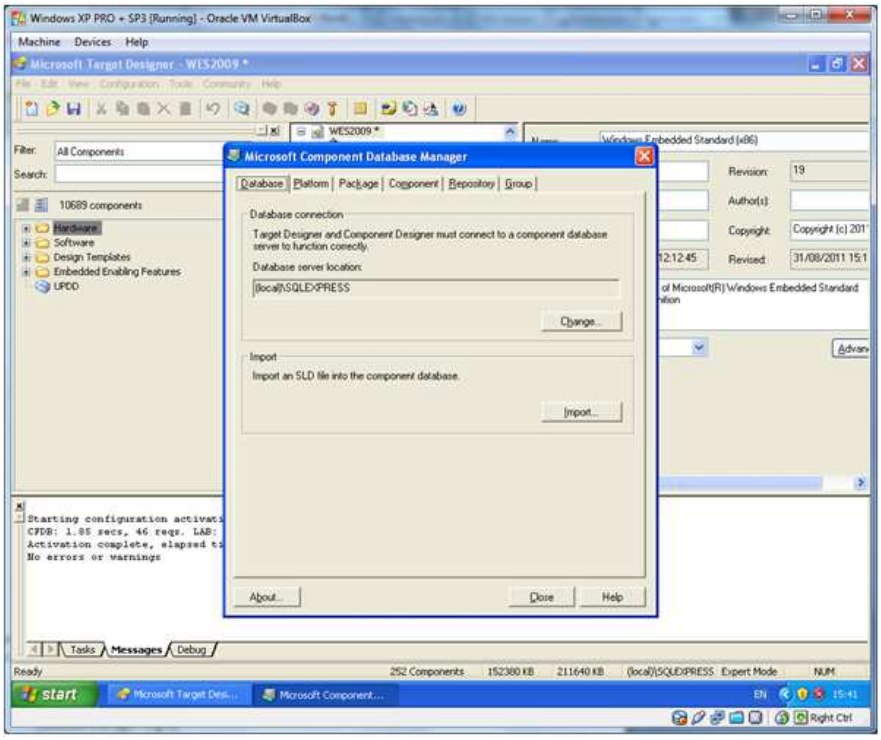
16. In the new dialog change the **Name** to 'Run setup' and adjust the Extended Properties thus:
    a. Arguments -s
    b. FilePath %24%\UPDD\setup.exe
17. Leave the other properties alone and click **'Add'**, then close the **'Add Component Resources'** dialog.



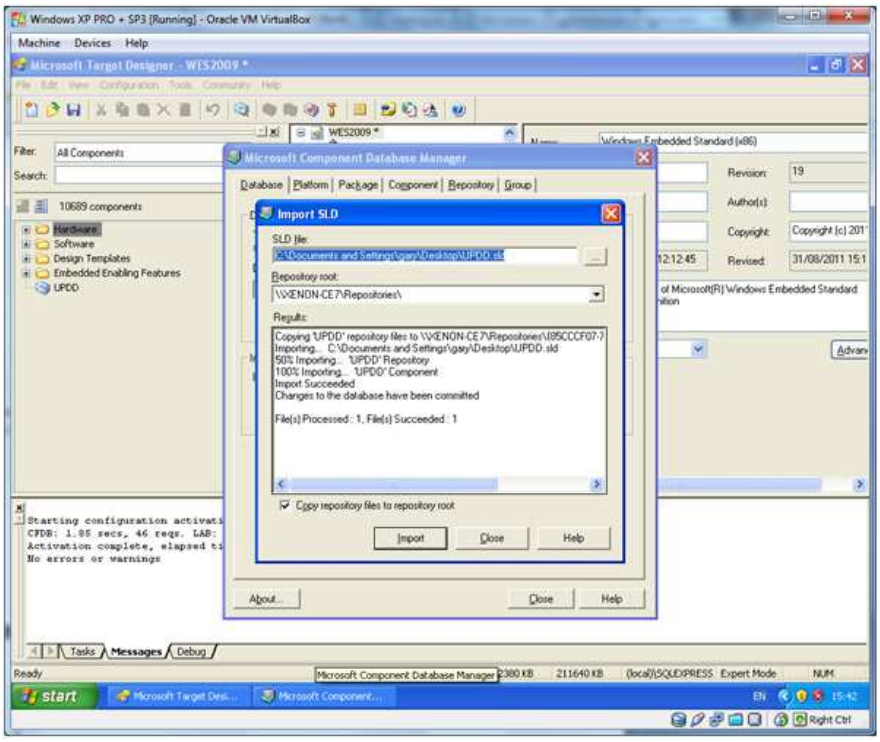
18. Save the component somewhere as **UPDD.sld**

You now have a component with the file needed and a command to run the installation silently (the '–s' argument) once upon First Boot. Now to add this component to the definition in target designer.

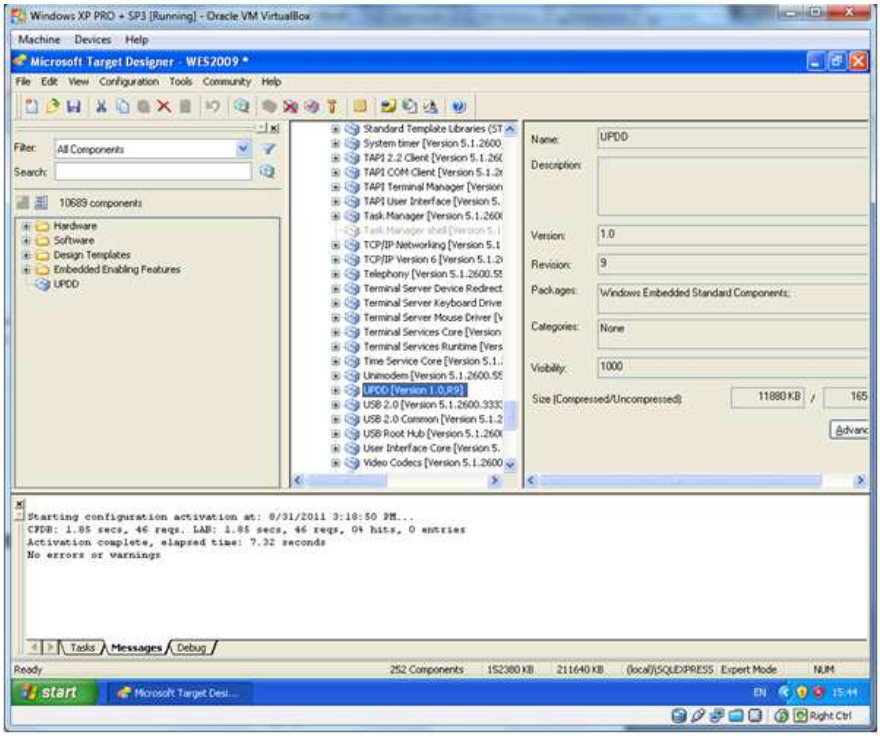
1. Run Target Designer and load or create a new definition to work on.
2. Firstly, the new component must be added to the list on the left. Select **Tools > Component Database Manager.**



3. Select **'Import'**, then set the **'SLD File'** field to the **UPDD.sld** file we just created, click import.



4. Click **'Close'** twice to return to the target designer, then refresh the component list on the left with **F8** or **Tools > Refresh**. The UPDD component should be on the left.
5. Add the component to the definition by dragging it to the central pane.



6. UPDD has several dependencies (components required for UPDD to work) so add the following components if they are not already included:
   a. **Software>System>User Interface>Shells>Windows Shell> Add Hardware Control Panel**
   b. **Software>Security>Infrastructure>Windows Update for Device Driver**

   Note: These dependencies are for a USB device, serial devices may have other component dependencies.

Following the above instructions the UPDD component and its dependencies have been added to the definition and WES2009 will silently install UPDD as part of the First Boot Agent.

Notes

USB PnP process

Prior to locking down the image it is important to connect any PnP touch device (e.g. USB) to allow for the Windows PnP process to complete. Internally this will add the device and complete the driver load sequence.

Driver settings

Following installation adjust the driver's settings
and calibrate as desired prior to image lockdown and deployment.

Serial Devices

If installing for a serial device and the UPDD package supports more than one device the you can either;

1) Not add the '-s' argument in step 16 to allow for the selection of the serial device during the FBA stage
2) Configure the serial device using the UPDD Console after the software has been installed
3) Use a setup.exe that only supports the desired touch device (request from Touch-Base)
4) Use a setup.exe that lists the desired serial device as the first device (request from Touch-Base)

Enhanced Write Filtering

Following image lockdown, if it is deemed necessary to allow further UPDD setting changes or new calibration to be performed and preserved over a reboot then the UPDD settings file must be writeable and persistent. In this case EWF (Enhanced Write Filtering) will need to be configured to exempt '**C:\Program Files\UPDD**' in order for UPDD to read/write files as required.

Predefined calibration

The setup.exe program can be delivered with predefined calibration if it is not possible / desirable to calibrate as part of the FBA process. See dump4tba calibration option described here.

WES7

These notes describe how to deploy UPDD software using the standard UPDD setup installation method during the OOBE stage and prior to image lock down.

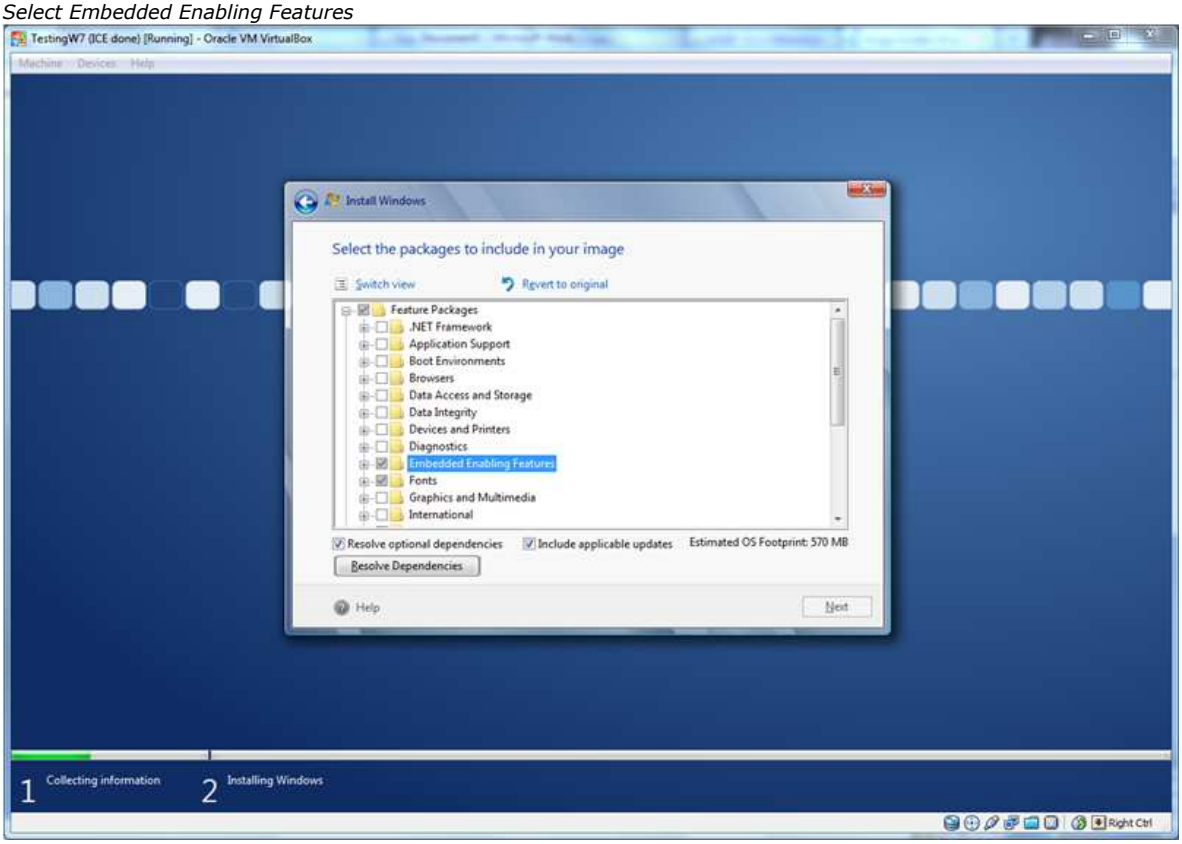
For WES7 the image must contain the 'tablet' component if the UPDD 'extended touch' feature (which utilises the UPDD Virtual HID component to interface with the OS) is to be utilised. If the 'tablet' component is not included disable the 'UPDD Extended Touch' setting. If working with a standard Microsoft build (see http://www.microsoft.com/windowsembedded/en-us/develop/windows-embedded-standard-7-os-components.aspx) note that only WS7P contains the table component (as defined in the Shell and User Interface section). WS7E and WS7C do not appear to contain the tablet component.

With UPDD 4.1.8, build 2110 and above, the UPDD setting "penservicesavailable" indicates if the penservice component is available. This is based on the presence of HKLM\SYSTEM\\CurrentControlSet\services\TabletInputService branch in the

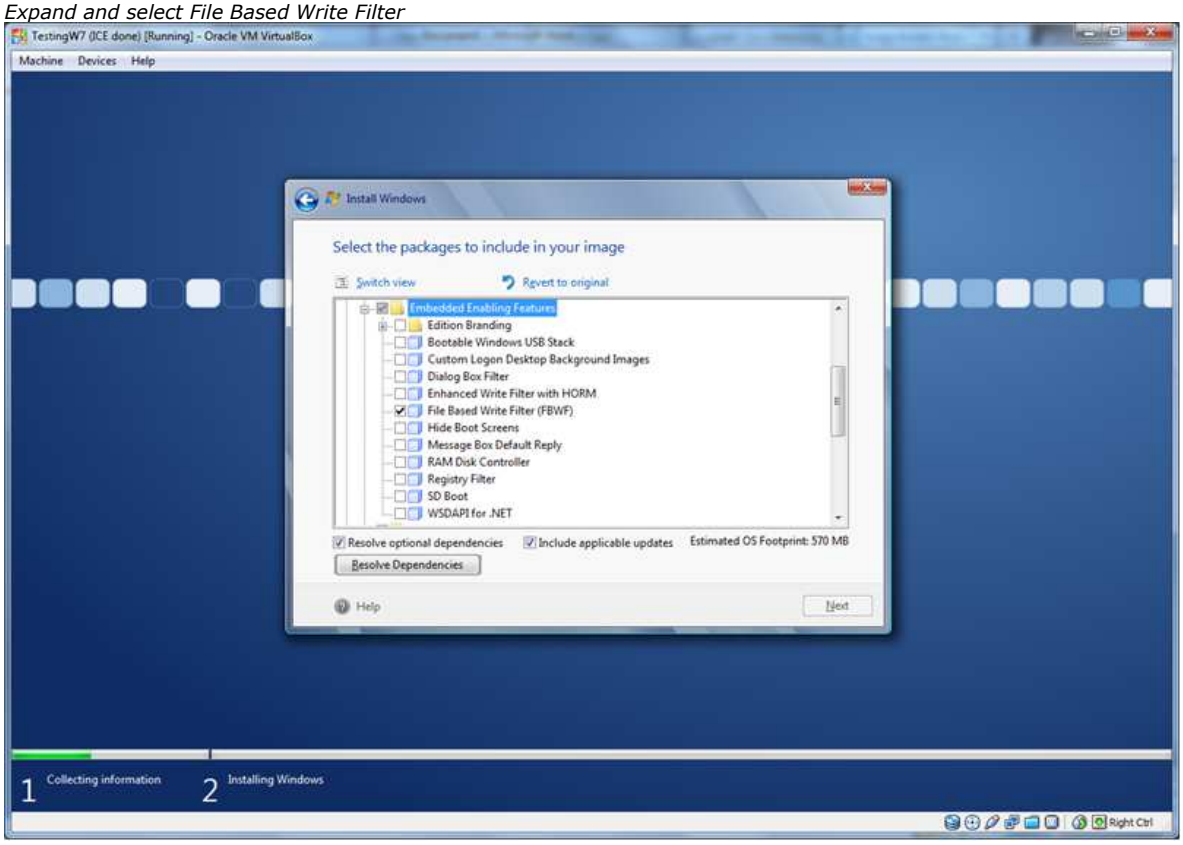
registry.

Install WES7 as normal using the IBW or ICE as required. A useful guide for installing WES7 can be found here. At this point no special behaviour is required for UPDD *unless* write filtering is intended for use; if write filtering is intended it must be File Based Write Filtering (FBWF) which is added to the configuration in the **Featurepack > Embedded Enabling Features > File Base Write Filtering** as below:

Select Embedded Enabling Features

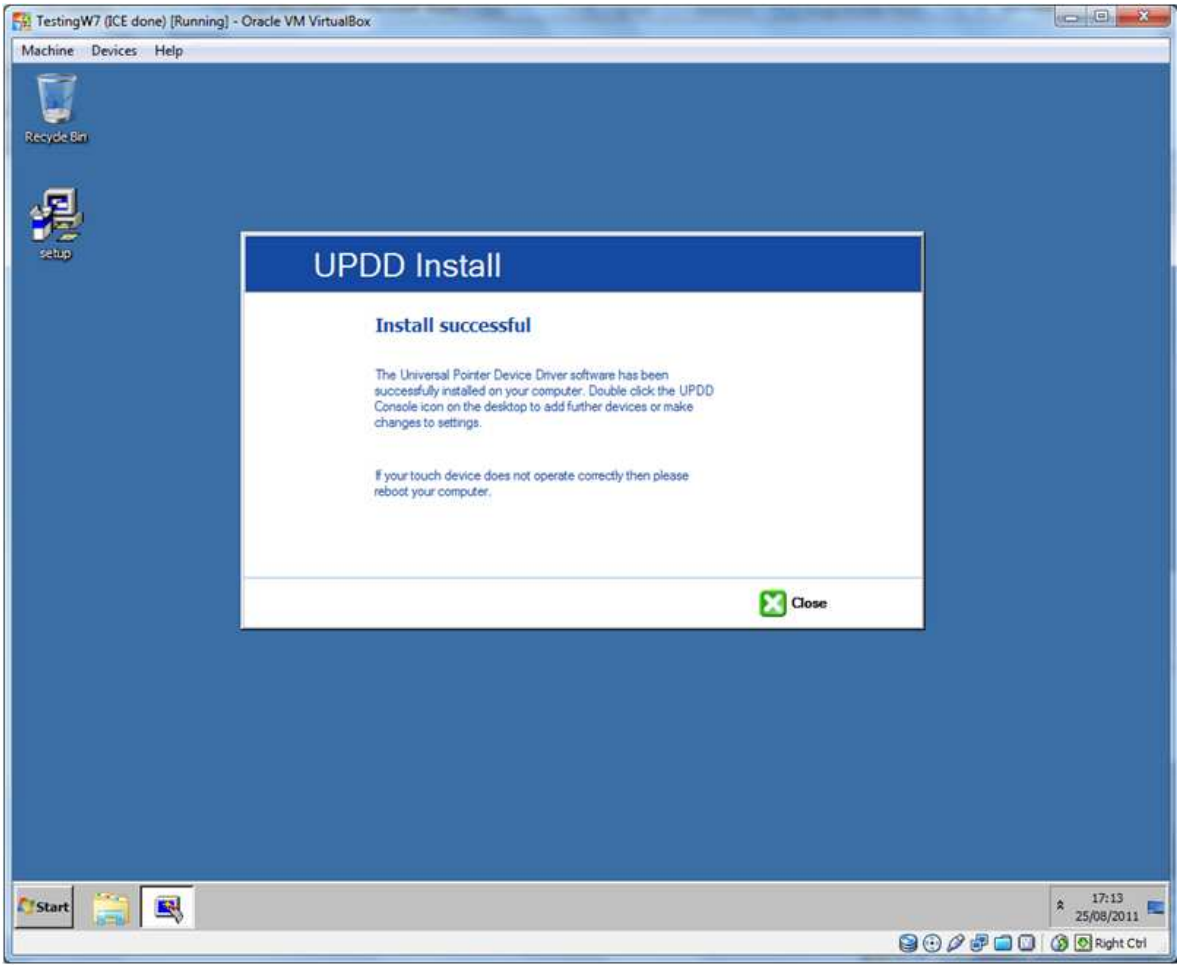


Expand and select File Based Write Filter



After install has completed the system enters an "Out-Of-Box-Experience (OOBE)" mode. At this point, installation of UPDD is simply a matter of running the standard **setup.exe** file and following onscreen instructions.

During installation the user should select the controller being configured and in the case of a PnP device (e.g. USB) the device should be connected to allow for the Windows PnP process to complete. Internally this will add the device and complete the driver load sequence.



UPDD is now installed on the system. Adjust the driver's settings and calibrate as desired. If write filtering is not in use then UPDD installation is completed and ready for image deployment using Deployment Imaging Servicing and Management (DISM).

However if it is deemed necessary to allow further UPDD setting changes or new calibration to be performed and preserved over a reboot then the UPDD settings file must be writeable and persistent. If write filtering is in use then see the FBWF Note below.

Notes

FBWF – File Based Write Filter

If file based write filtering is in used, follow the instructions below to add UPDD to the exclusion list (notes below assume C: as standard, replace with different drive letter if required):

1. Open command prompt and navigate to C:\Windows\System32
2. **FBWFMGR /ENABLE** (*this sets FBWF to begin on the next boot)*
3. **FBWFMGR /ADDVOLUME C:** (*Instructs FBWF to protect that drive from changes)*
4. **FBWFMGR /ADDEXCLUSION C: "\PROGRAM FILES\UPDD\"** (*Excludes folder from protection)*
5. Exit the command prompt

Upon the next boot, FBWF will be active and only files in that directory will persist in any changes through reboots. Note that this step should be taken last, just prior to taking the image as it will be locked afterwards; naturally, any other exclusions should be made at this time in the same manner after step 4. UPDD is now installed and able to write to the tbupdd.ini file; it is ready for DISM.

Predefined calibration
The setup.exe program can be delivered with predefined calibration if it is not possible / desirable to calibrate as part of the FBA process. See dump4tba calibration option described here.

Contact

For further information or technical assistance please email the technical support team at technical@touch-base.com