

# **Configuring an XP Embedded Thin Client**

**By Sean D. Liming & John R. Malin**  
SJJ Embedded Micro Solutions

Copyright © 2005 SJJ Embedded Micro Solutions, LLC., All Rights Reserved

No part of this guide may be copied, duplicated, reprinted, and stored in a retrieval system by any means, mechanical or electronic, without the written permission of the copyright owner.

First Printing: April 2005

V1.1 Update: October 2005

Published in the United States by

**SJJ Embedded Micro Solutions, LLC.**

11921 Tivoli Park Row #5  
San Diego, CA 92128 USA

[www.sjjmicro.com](http://www.sjjmicro.com)

Attempts have been made to properly reference all copyrighted, registered, and trademarked material. All copyrighted, registered, and trademarked material remains the property of the respective owners.

The publisher, author, and reviewers make no warranty for the correctness or for the use of this information, and assume no liability for direct or indirect damages of any kind arising from the information contained herewith, technical interpretation or technical explanations, for typographical or printing errors, or for any subsequent changes in this article.

The publisher and author reserve the right to make changes in this publication without notice and without incurring any liability.

Windows, .Net Embedded, and Visual Studio are registered trade mark of Microsoft Corporation.

## Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
1.1	THE HARDWARE INSIDE.....	4
1.2	RUNNING TARGET ANALYZER (TAP).....	5
1.3	PLATFORM MACRO COMPONENT AND MISSING DRIVERS .....	6
1.4	CONFIGURING THE SOFTWARE .....	7
1.5	READING FURTHER .....	8
<b>2</b>	<b>XP EMBEDDED THIN CLIENT SOLUTIONS.....</b>	<b>9</b>
<b>3</b>	<b>BIBLIOGRAPHY.....</b>	<b>10</b>

# 1 Introduction

Thin Clients have become very popular computing platforms for IT departments. The ability to run an operating system with all the major applications, such as, word processing, spread sheets, and other company software, on remote servers saves companies millions in computing service fees. Thin Client architecture features very little local storage, and acts like a window to the backend servers that host the major applications. IT departments like these devices because they save hundreds of dollars on the cost of a fully configured PC, and they reduce the application software maintenance costs of these machines for deployment and upgrading. Of course no two IT departments are alike. Each has its own approach to security or custom implementation to access the network, therefore, easy configurability of Thin Clients is important.

There have been several requests on the Windows XP Embedded (XPe) newsgroups asking to modify either Thin Client hardware or make some change to the software. The degree of the changes is quite diverse, but many of these customization requests involve modifying the whole underline operating system image. Minor changes for TCP/IP address and Domain names can be handled within the existing image, but changing the whole image takes some reverse engineering. This article will provide the first steps to customizing XPe Thin Client platforms.

## 1.1 The Hardware Inside

An XPe image consists of a hardware layer (HAL, device drives) and a software layer (TCP/IP, notepad, Explorer). The XPe database groups the OS components into these two categories. Every image includes device drivers that interface to the hardware. A little understanding of the hardware is very helpful when make changes to the XPe OS image.

First, the processor technology is an important aspect to the performance and the physical size of a Thin Client. Many of the Thin Client terminals run either a VIA or Transmeta processors. These processors are ideal for Thin Client hardware platforms due to their low power, good performance, and the fact that they do not require a cooling fan.

The boot media is the next, if not the most important, piece of hardware that you need to understand. Many Thin Clients use solid state technology such as CompactFlash and mini-IDE flash drives as the boot and storage medium. The benefit of non-rotating media makes Thin Client systems a little more robust then their bigger PC counterparts. The flash life is a bit of an issue, though.

Windows XP Pro and XP Embedded are operating systems that are constantly reading from and writing to a disk. This translates in to many flash erase cycles, of which these CompactFlash and mini-IDE flash types have a limited number. Without sophisticate wear leveling technology, CompactFlash and mini-IDE flash cards will wear out much faster than their normal commercial use. To protect the life of the flash, Thin Client OEMs implement XP Embedded's Enhanced Write Filter (EWF) to prevent writes (erase cycles) to the flash drive. All writes are re-directed to the RAM disk (a.k.a. overlay). If you need to make any changes to the OS such as domain connectivity or change TCP/IP static address, the EWF can be disabled to allow updating of this data, and then re-enabled. Setting up EWF is a bit tricky, especially when it comes to CompactFlash cards. When you are re-creating an XPe image for your Thin Client you will have to get familiar with EWF setup.

Another detractor to these flash types is boot time. CompactFlash and mini-IDE flash devices are booting with Port I/O mode at a rate of 3 to 5MB/s. A system could take as long as 90 seconds to boot. In contrast, an SATA hard drive uses DMA to speed up access to 155 MB/s.

If your target has an internal header for USB 2.0 and has BIOS support to boot USB 2.0 devices, you could upgrade to M-System's uDiskOnChip (uDOC). The uDOC offers security, reliability, performance, and cost effective solutions for embedded systems. You don't have to worry about flash life since the uDOC features M-System's TrueFFS® wear leveling technology. The boot times are just as fast as a hard drive, since the uDOC is operating over USB 2.0. The overall system performance will improve with the uDOC in your system.

Finally, you will need to be familiar with other devices on the platform such as the Ethernet, audio, and video controllers. You may have to create components for these devices if the drivers are not in the database.

## 1.2 Running Target Analyzer (TAP)

Creating a custom XP Embedded image implies that we have to create a custom configuration in Target Designer. To get the baseline components for the target hardware configuration, we need to run Target Analyzer (TAP.EXE) on the Thin Client system to get a PMQ file that contains a list of the devices in the system.

It should not be too difficult to run TAP since XP Embedded is already running on the Thin Client. This is where understanding the hardware becomes important. In order to run TAP, you need to get TAP on the Thin Client. Here are some options:

1. Attached a USB floppy drive or USB flash key to the Thin Client and run TAP.EXE from the floppy or the flash.
2. Copy TAP directly to the internal storage media. EWF can get in the way, so one way is to remove the CF or mini-IDE from the Thin Client, connect the media to your development PC using an adapter, copy TAP to the media, then place the media back into the Thin Client, boot the Thin Client, and finally run TAP. You will, also, need a method copy the PMQ file that TAP generates to the development machine or save the PMQ file to the media, keeping in mind the EWF issues.
3. Alternatively, if you can get to a network share, you could download TAP to the Thin Client and then run TAP so the output is place back onto the network share.

Of course if you are trying to upgrade the Thin Client from another OS, then you will need to perform some addition steps. The MS-DOS version of Target Analyzer (TA.EXE) could be used, but it is doesn't produce all the components needed to develop a bootable image. You may need to add a few extra components to get the image to boot. Here is a list of some of the components to add:

- PCI standard host CPU bridge
- PCI standard ISA bridge
- Plug and Play Software Device Enumerator
- Primary IDE Channel
- Secondary IDE Channel
- Disk Drive
- Standard Dual Channel PCI IDE Controller
- Standard IDE/ESDI Hard Disk Controller

There maybe more components to add. Ideally, you want to install XP Pro on the target system and then run TAP.EXE under XP Pro. The resulting PMQ file will have the complete list of system components. You may have limited local storage space so installing XP Pro might not be possible. One solution I have used is to build an image based on the MinLogon Sample Macro component to host the TAP session:

1. **Open** Target Designer
2. Create a new configuration called **TAPrun**.
3. Add the **MinLogon Sample Macro** component to the configuration.

4. Run a **Dependency Check** with Auto-resolve enabled.
5. **Build** the image
6. Once the build has completed, copy TAP.EXE to the root of the image

The final XPe image uses the Standard PC HAL so the image should run on all PC platforms that can run XP. The image size is about 25MB. Download the image, with TAP.EXE in the root of the image, to the target device and let FBA run. Once FBA completes, you can run TAP.EXE to get the PMQ file.

The resulting PMQ file will not have a complete list of components, but it will have found the critical components to boot the OS. Follow the steps outline in section 1.3 to import and create a Platform Macro Component. The only problem with this technique is that the Standard PC component will be in the PMQ list instead of another possible Computer component like ACPI. Many systems today support the ACPI specification for power management and other BIOS features. If you want to get the exact Computer components, run TA.EXE from DOS on the target, import the resulting PMQ file, and compare the Computer component results of the TA.EXE analysis under DOS to the TAP.EXE analysis under MinLogon. TA.EXE under DOS will reveal the Computer component that is best suited for the system. You may have to replace the Standard PC component with “Advanced Computer and Power Interface” or “ACPI Uniprocessor PC” components if your hardware has ACPI support, and you will want to add any other missed components that TA found.

### 1.3 Platform Macro Component and Missing Drivers

Capturing the PMQ file with TAP on the target system is the most challenging aspect of this process. The next step is to create a Platform Macro Component that contains the hardware components and basing line for the configuration.

1. Open **Component Designer**
2. From the menu, select **File->Import**. This will bring-up the open import file dialog.
3. Locate and **OPEN** the devices.PMQ file.
4. The Import File dialog appears. Enter a log file name: \deviceMacro.log and click the **Start** button. When completed, the import dialog list the number of components found out of the number of devices in the PMQ list. A new SLD file is created with a component titled devicesTAP.
5. In the details plane of the devicesTAP component, change the name of the component to XPES\_target.
6. Next to Prototype in the details pane, click **Browse**.
7. The Select Prototype Component dialog appears. Transverse the component tree **Software->Test & Development->Selector Prototype Component**. Highlight the component and click **OK**.
8. Select Group Membership, and from either the menu or right click the mouse select **Add Group Membership**.
9. The Add Component Group Membership dialog appears. The logical place to put the platform is under platforms. Expand the tree until you locate platforms: **Categories->Hardware->Platforms**, and then click **OK**.
10. **Save** the SLD file to the hard drive.
11. Open **Component Database Manager** and import the newly created SLD file into the database.

If there were any missing components, such as network, audio, or video devices, you will have to create custom components for them. The deviceMacro.log file will list the names of the missing device components and their PCI IDs. Sometimes the hardware vendor will supply the drivers on CD, but most of the time you will have to search and download these drivers from the chip manufacturer. Since the VIA and Transmeta processors are typically a two chip solution, finding the drivers should be straight forward.

Most of the time, the driver's INF file can be imported into Component Designer to create the SLD and component. Other components sometimes require a little more work with the help of Component Helper, DependencyWalker, and FilMon/RegMon tools.

Once you have created the components and imported them into the database, you should edit the Platform Macro Component and add the newly created hardware components as dependencies.

#### 1.4 Configuring the Software

Now that the hardware layer is ready, the next part is the software customization of the image from within Platform Builder. You can mold the image to your specification since the configuration is being created from scratch. Some of the possible changes are:

- Custom Shell - Instead of a stock shell such as Explorer, you can create a custom shell with its own look and feel, or create multiple shells for specific user types such as administrator and user.
- Security Settings - Custom security settings can be added to lock users out of specific applications, directories, or drives.
- Third Party Software - There are many companies that develop software for Thin Client solutions and most of these companies have XPe components that can be imported into the XPe database.
- Administration Support – The Thin Client should have some basic administration support components for networking, display, etc..

Component	Thin Client Usage
Domain Participation	The Domain Participation component is a macro component that bundles components which make it possible for an embedded device to participate in Windows domain security.
Netshell	Command line utility to set the TCP/IP settings for network adapters. Required to access the network settings in the network control panel.
TCP/IP Utilities	Contains a number of TCP/IP utilities including ipconfig.exe
System Control Panel	Provides the interface to change the Domain and computer name
Display Control Panel	Change video resolution
Date/Time Control Panel	Set the local time zone, date, and time
Audio Control Panel	Support systems with sound and audio

The most important feature to enable and the most challenging is EWF. Unless you are using the uDOC, you will need to implement EWF to protect the flash drive. Mini-IDE flash drives shouldn't be a problem since they are IDE flash disk that are fixed devices. Many manufacturers setup EWF with a second partition as the EWF volume. You have your choice of EWF implementation and overlay type, since you're recreating the image from scratch. Since this is flash media, RAM or RAM-REG overlays are the two best choices.

CF cards are a different story. Most off-the-shelf CF cards are ID'd as removable, which creates most of the headaches to setup EWF on CF. XP/XPe will only view one partition on a removable media, thus RAM-REG is the only overlay type choice that can be made. CF cards that are ID'd as fixed devices can be ordered special from the manufacturer. The choice of using the RAM overlay with the second partition is possible. You may find that the Thin Client manufacturer has already configured the flash drive as fixed so all you need to do is replace the image on the drive.

Not only will you have to enable EWF, but you will want to control EWF; so updates and modifications to the image can be made within a working image. The EWF API can be used to add custom controls to shells or EWF administrator applications. EWFMGR.EXE and the EWF

API can only be accessed via Administrator accounts. Power Users, Users, Guest accounts can not change the state of EWF.

If you need to duplicate the image, you will have add the System Cloning Tool component to the configuration and add a special reseal library to address EWF.

Once the configuration is completed and the image is built, the image will need to run FBA on the target system. The image will have to be copied down to the target media, either through physically removing the device from the target system or using a software solution like Remote Recover.

### **1.5 Reading Further**

The overview presented in this article gives you the basic tips to get started on configuring a Thin Client with a custom image. The real work is in the details. There are many solutions for working with Windows XP Embedded that can be found in a number of books, articles, and websites:

- [msdn.microsoft.com/embedded](http://msdn.microsoft.com/embedded)
- **Windows XP Embedded Advanced**, Sean D. Liming, RTC Books, 2003, ISBN:0-929392-77-9
- **Windows XP Embedded Supplemental Toolkit**, Sean D. Liming, Cedar Hill Publishing, February 2005, ISBN: 1-932373-96-9.
- [www.seanliming.com](http://www.seanliming.com) – XPe Center @ seanliming.com – contains various articles and links to different resources on Windows XP Embedded.



## 2 XP Embedded Thin Client Solutions

Thin Clients come in many form factors: terminal, notebook, and tablet. There are many companies that are building hardware and software solutions for XPe Thin Clients. The table below lists a few companies and websites that have a solution for XP Embedded Thin Clients.

We like to keep the table up to date. Please let us know if you find a company that makes an XPe Thin Client solution that is not in the table.

<b>Company</b>	<b>Product Type</b>	<b>Website</b>
Acute Network Technologies	Terminals	<a href="http://www.acutetech.com">www.acutetech.com</a>
Amir Technolgy Lab, Inc (ATLabs)	Terminals	<a href="http://www.atlabs.com">www.atlabs.com</a>
BOSaNOVA, Inc.	Wireless Tablets, Terminals	<a href="http://www.bosweb.com">www.bosweb.com</a>
Citrix	Software Add-on	<a href="http://www.citrix.com">www.citrix.com</a>
GraphOn	Software Add-on	<a href="http://www.graphon.com">www.graphon.com</a>
Hewlett Packard	Terminals	<a href="http://h18004.www1.hp.com/products/thinclients/index_t5000.html">http://h18004.www1.hp.com/products/thinclients/index_t5000.html</a>
ICOP	Embedded Terminals	<a href="http://www.icop.com.tw">www.icop.com.tw</a>
Jetro Platforms, Ltd.	Software Add-on	<a href="http://www.jp-inc.com">www.jp-inc.com</a>
Linux Technology Ltd.	Terminals	<a href="http://www.lt21.com.tw">www.lt21.com.tw</a>
Maxspeed	Notebook, Terminals	<a href="http://www.maxspeed.com">www.maxspeed.com</a>
Neoware Systems	Terminals	<a href="http://www.neoware.com">www.neoware.com</a>
NLynx Technolgies	Terminals	<a href="http://www.nlynx.com">www.nlynx.com</a>
Devon IT, inc.	Terminals	<a href="http://www.ntavo.com/ntatterminal.php">www.ntavo.com/ntatterminal.php</a>
Praim	Wireless Tablets, Terminals	<a href="http://www.praim.com/default.htm">www.praim.com/default.htm</a>
RemoteScan	Software Add-on	<a href="http://remote-scan.com">http://remote-scan.com</a>
Tarantella, Inc.	Software Add-on	<a href="http://www.tarantella.com">www.tarantella.com</a>
ThinPATH Systems	Software Add-on	<a href="http://www.thinpathsystems.com">www.thinpathsystems.com</a>
VXL	Terminals	<a href="http://www.vxl.net">www.vxl.net</a>
Wyse	Terminals	<a href="http://www.wyse.com">www.wyse.com</a>

### **3 Bibliography**

*Windows XP Embedded Advanced*, Sean D. Liming, RTC Books, 2003, ISBN:0-929392-77-9

*Windows XP Embedded Supplemental Toolkit*, Sean D. Liming, Cedar Hill Publishing, February 2005, ISBN: 1-932373-96-9.