Coming Together: Best Practices for XPe in the Corporate Network

By Sean D. Liming with John R. Malin

First Printing: March 2006

Published in the United States by

SJJ Embedded Micro Solutions, LLC.
11921 Tivoli Park Row #5
San Diego, CA 92128 USA

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

Windows is a registered trademark of Microsoft Corporation

1 Coming Together: Best Practices for XPe in the Corporate Network

For several years, I have been working with customers directly and indirectly to build embedded systems with XP Embedded. Devices have included voting machines, slot machines, set-top boxes, medical test equipment, network printers, point-of-sale registers, kiosks, atmospheric test equipment, thin clients, and game machines. As you can see XPe reaches a wide range of applications. Many of these devices need to connect to a corporate network, and that makes developing these systems a little more challenging. I tell OEMs to look at architecture design thinking about how the system is to be used in the field – how the user accesses the system, how it is administered, how it is upgraded – all of these considerations dictate hardware and component/feature choices. In the end there is a careful balance between what the OEM wants and what the IT department will be able to handle.

1.1 The Two Points of View

From the OEM's point of view, locking down the systems so it looks like an appliance is a popular request. Flexibility is a powerful asset in XP Embedded. This flexibility gives OEMs the means to try to hide the fact that a PC, BIOS, or OS is running underneath. They also want to protect the OS from corruption and employee XPe features like the Enhanced Write Filter to protect the OS boot partition. OEMs really don't want customers installing or playing with other PC software. It would not do any good to have video games such as Halo® or Freelancer™ running on a network printer.

The power of flexibility to lock down a system means the OEM has to spend more time architecting the image to address all aspects of the application's life cycle. There are various reasons for locking down the system, but the most important is support. With multiple systems in the field, support can become an expensive task. Creating support updates for a system assumes a certain state of the system at the time of update. It becomes an extremely difficult task to automate the update process for systems that have been modified in unknown ways or had additional applications installed on the system them by the end users.

The challenge comes when the devices need to connect to network. IT managers don't want devices connected to the corporate network that could lead to potential network problems or security issues.

IT managers are very concerned about support too. They typically purchase a device that meets a specific need and function. The device gets installed, and hopefully nothing is required for daily maintenance. For many, security is the main issue. The IT departments don't want devices on the network that can be subject to viruses, worms, and other potential attacks.

The problem is that security is subjective. OEMs will put the basic security features in place, but the IT manager may want something more refined.

What the OEM wants to build and what the IT manager will accept is sometimes in conflict. Bringing these two groups together involves some trade-offs in design and a little education in XPe. There are some best practices that the OEM can employ and the IT manager needs to be familiar with. Here we are going to take a look at these best practices that both worlds can speak too.

1.2 Network Support

The first item to discuss is networking support. If the system doesn't have wired or wireless Ethernet, then there are no issues for the IT manager to be concerned about network wise. If upgrades to the system are important but networking is not implement, the OEM must have a service or other method of upgrade delivery. These methods could include, switching out boot CD-ROMS, flash drives, or sending a technician with a flash key.

Of course networking is a key part of many XPe systems, and the OEM must include the component and features necessary for device connectivity. Fundamentally, a device can connect to a workgroup or a domain. As an OEM, if you don't know what the customer is going to require, the best solution is to provide services for both, but setup the system for workgroup by default.

The following are the basic networking components to have in the XPe configuration:

- File Sharing - Security Update KB896422
- Client for Microsoft Networks
- Core Networking
- MSFS
- Windows Logon (Standard) - Some standalone systems use the Minlogon component, but this component removes the logon features needed for domain participation.
- Computer Name User Interface
- Netlogon/NetJoin
- Netshell – provides the network control panel interface
- System Control Panel – Allows for computer name changes and domain connectivity

As an OEM, you need to provide a mechanism for the IT manager to access the network settings. As an IT manager, you need to make sure that the equipment you buy has the necessary administration tools to do the following:

- Change computer name
- Set a dynamic or static TCP/IP address settings
- Join a Domain
- Set SSID, WEP, or WPA information for 802.11 wireless networks.

If there is a custom shell, some OEMs opt to have a second administration account with a different shell (typically Explorer) that the IT administrator can logon and access control panel and other utilities. Others create special interfaces that provide the necessary capability or simply launch the control panel applet.

If a system is headless, where access to the GUI controls is not possible, command line utilities provide access to network settings via a remote Telnet session. Alternatively if the image supports Window Management Instrumentation (WMI) support is enable, one could access the device remotely using VBscript. Currently there are no command line utilities as of this writing that support 802.11 setup.

1.3 Security

Security is a bit of a subjective topic. The more the system is locked down the more in-flexible the system can be. A different shell for different users is one method of locking down the system from general user access. In this scenario there are two accounts on the system; one is a power user account that has the main shell that interacts with the user. The custom shell provides little access to system settings or administration tools. The power user account is important because it acts as the backup, since access to administration controls is limited. The second account is an administration account that has a full shell such as Explorer. The administrator can use a hot key locally in the user account to log-off or logon via remote desktop connection and log onto the administration account to perform any administrative tasks.

Different shells for different users are a simple security layer. A deeper layer would be to create custom local security policies. Local security policies can restrict users' or groups of users' access to drives, specific files, enforce network access restrictions, and many other available features. By default access to these policies are disabled in XPe. The following components need to be added to a configuration to enable the generic security policies:

- Security Shell Extension
- Windows Security Configuration Editor Client Engine
- Security Settings Editor – secpol.msc allows a mechanism for editing the local security policies on the device.
- Group Policy Client Core
- Group Policy Core Administration MMC Snap-In – Group policy editor

Once the local security policies have been enabled, one can create a custom security policy for the device. One nice feature from the OEM point of view is restricting access to specific drives; namely preventing general user access to the OS partition. There are many more policies that can be set. Unless the OEM and customer agree on which policies need to be enforced, most OEMs would have to at least provide a device with the security polices enabled; and like Ethernet settings, provide the administrator access to the local or group policy editor. IT managers need to be proactive with OEMs and request or ask about setting security policies editors when purchasing a product that is connected to the network.

Restricting user access is one aspect of security; the other is keeping out outside attackers. Most of these are the typical items such as virus software, firewall, anti-spyware, etc. For the OEM, the most obvious choice is to add the Windows Firewall to the image. Any special application that needs access to the network can be addressed in the Firewall component's settings with Target Designer. If the IT managers need to configure application access, the Firewall control panel needs to be available.

As XPe SP2 was being launched, several virus software makers were adding support for XPe. Virus software does add a key level of security if a device is on an open network. The challenge comes when virus definitions need to be updated. Some embedded devices are booting off of flash with limited space so constant update could fill the disk over time.

Last layer of protection that the OEM can easily implement is Data Execution Prevention (DEP). DEP monitors memory to see if programs are using system memory safely. DEP operates either standalone or in cooperation with DEP enabled microprocessors. You will have to refer to your microprocessor's documentation for more information on DEP compatibility. DEP marks some memory locations as "non-executable". If a program tries to run code (malicious or not) from a protected location, DEP closes the program and notifies the user of the issue. Think of DEP as the inner most ring to the security system. The feature is enabled by the OEM in the HAL component for the target system.

1.4 Boot Devices, EWF, and Device Upgrades

We mentioned flash media in the last section. XPe can boot from different types of media: Component Flash, hard drive, USB flash disk, IDE flash disk, CD-ROM, DVD-ROM, or network boot. The choice of boot media plays an important role in how a system is maintained.

Many embedded systems use solid state media (flash or CD-ROM) as the primary boot media. Nearly all thin client manufactures use flash media to boot the OS. Two key design issues, Enhance Write Filter (EWF) management and upgrades, come into play that the OEM needs to address and the IT manager needs to be aware of.

EWF protects a volume partition from write access by re-directing any write to a separate media called an overlay. A EWF overlay can be either RAM or a separate partition on the drive. The protected drive will still look like a read write disk. Flash devices have a short life span unless wear-leveling technology is built in or the OEM uses the EWF to protect the OS partition. Anytime changes or updates need to be made, EWF needs to be turned off. Most OEMs create an application that allows the IT administrator to control the state of the EWF. Sometimes this EWF manager is a separate application or system tray icon. The IT manager needs to be aware of the state of EWF when making changes like network settings.

Upgrades are also an important part of the life cycle of a device. There are two types of upgrades, incremental and full OS. Incremental updates are updates of a single file, security patch, or service pack. Full OS is replacing the whole OS in the system. For the OEM, planning is the key here. How the system gets upgraded, what gets upgraded, who performs the upgrade, and if EWF is in the system, how is EWF controlled are issues that need to be addressed early in the design. With many viruses, worms, and patches coming out each day, IT departments want to ensure that their network is safe with the latest updates.

Once again the boot media plays an important role in how a system is maintained in the field. If a system has enough room to handle incremental updates and patches than the OEM can implement some upgrade features. XPe comes with several solutions, and many OEMs decide to implement some of their own.

XPe comes with support for System Update Service (SUS). System Update Service (SUS) was introduced to let the IT Managers control the updates that get distributed company wide. A SUS server downloads the latest updates from Microsoft's Windows Update server and stores the updates locally. The IT manager can approve and control the updates to their network. OEMs can include the Windows Update Agent for SUS 1.0 Servers component in their systems, and the device needs to provide access to the Global Policy Editor to setup the appropriate polices. In addition, SUS servers exist on Domains so the device needs to be attached to a domain, which means IT administrators also need to access network settings. If you haven't already, you can begin to see a theme here.

A more sophisticated management solution is to use Microsoft's System Management Server (SMS). SMS allows you to manage the system across a wide enterprise network. An administrator at one desk can manage all of the systems across a single interface. Where SUS can only support Microsoft update, SMS supports custom updates and more. Once again the OEM needs to build in SMS into the image and provide access to the SMS setup and network utilities to the IT administrator.

Both SUS and SMS are widely used in domains today. Not every system is connected to a domain. This allows OEMs to come up with their own upgrade solution. To provide the most flexibility when not connected to a domain, XPe comes with Device Update Agent (DUA). DUA is considered the lowest common denominator for incremental updates. The DUA service runs in the background and can fire off at a set specific time to grab an update locally or remotely across a network. The update consists of a DUA command file and any file updates that need to be made to the system. The DUA command set is a simple no-frills command language that performs the basic of administrative tasks: move, add, and delete. Conditional branching can be performed by integrating and calling VBScripts. How updates get to the device is up to how the OEM wants to support the device:

- Customer downloads the update from OEM website to a central server that all devices will pull down at the customer set specific time.
- Device dials up OEM's central server to download the updates.
- Customers receive the latest updates on a CD, and run a utility that launches DUA and performs the update.

Now full updates are slightly different issue. The size of an XPe image can ran from 55MB to 500MB and beyond. Downloading a full OS across a dial-up connection might not be the best solution. Boot media plays in to the solution. A full OS solution could be simply replacing the solid state media with a new version of the OS. If network connectivity is available, a commercial solution like Winternal's Remote Recover or other customer solution could be used to download a full OS image.

As the IT manager, you need to understand what the maintenance solution for the device is and how this affects daily operation.

1.5 Summary

When embedded systems get connected to a network, there are two forces at play, OEMs who want their products to be dedicated appliances, not general workstations and IT managers that want a safe, secure network and devices that are hassle free. The bottom line for IT managers is that if a device is connected to the network, they do not want to have to physically go to the device to configure it, check its status, or upgrade it, whether the device is in an enterprise environment or a corporate intranet environment. OEMs need to recognize this need and provide remote services that will aid IT managers in realizing that goal. The other hot button for IT managers is network security. No IT manager wants to add a device to the network that is going to make the network more vulnerable to attack or compromise the network security restrictions that are in place. Thus the major items that OEMs need to include in their systems for their device to play nice on the corporate network are:

- Access to administrative functions such as network settings, local policies, system settings, etc. This can be accomplished by providing the administrator access either through "dual shell / dual account strategy or a hidden hot-key enabled, password protected, application that allows access.
- The ability to control EWF, if EWF is in the system. The IT administrator needs to commit changes or turn on/off EWF.
- Enable basic security policies by adding the Security Shell Extension.
- Enabling Data Execution Protection

From these basic features the rest falls to architecture design. I stress in my books and classes that architecture is a key to developing a solid XPe system. OEMs have to make the decision how the system performs in the field, and the decisions need to be made early in the design. Customers play an important role in the development of embedded systems, but many times the OEM has to guess at what is required. Both sides come together when the basic system administration tasks are made available.

2 References

Windows XP Embedded Supplemental Toolkit Covering XP Embedded Service Pack 2, Sean D. Liming, Cedar Hill Publishing, 2004, ISBN: 1-932373-96-9

Windows is a registered trademark of Microsoft Corporation