

ThinClient Development Kit for Windows XPe

Windows XP Embedded Development Guide for the eBOX-IV 48XX Series (TcDK)

By: Sean D. Liming and John R. Malin

[SJJ](#) Embedded Micro Solutions

Copyright © 2007 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: November 2007

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

1	INTRODUCTION	4
1.1	KIT CONTENTS / OTHER REQUIREMENTS	4
1.2	FURTHER READING, INFORMATION AND TRAINING	5
1.3	WHAT'S NEXT?	5
2	DEVELOPMENT SETUP	6
2.1	DEVELOPMENT SYSTEM REQUIREMENTS	6
2.2	INSTALLING THE TOOLS	6
2.2.1	Part 1: XP Embedded SP1 Setup	6
2.2.2	Part 2: XP Embedded SP2 Setup	7
2.2.3	Part 3: Install Feature Pack 2007	8
2.2.4	Part 4: Install Update Rollup 1.0	9
2.3	IMPORTING COMPONENTS	9
2.3.1	Part 1: Extract the Kit Files	9
2.3.2	Part 2: Import the SLD files	10
3	BACKGROUND ON THE COMPONENTS	12
3.1	WHERE TO BEGIN: TARGET ANALYZER	12
3.2	MISSING DRIVER COMPONENTS AND SUPPORT APPLICATIONS	13
3.3	SPLASH SCREEN – DEVICE BRANDING	13
3.4	CUSTOM COMPONENTS: SJJ602_EXAMPLES	13
4	IMAGE DEPLOYMENT SETUP FOR USB BOOT	14
5	EXERCISE 1: CREATING A BASIC IMAGE	17
5.1	BUILD THE CONFIGURATION YOURSELF	17
5.2	DEPLOYING THE OPERATING SYSTEM	20
6	EXERCISE 2: THIN CLIENT IMAGE WITH EWF	21
6.1	FBWF MANAGER CONSOLE APPLICATION	21
6.2	BUILD THE CONFIGURATION YOURSELF	22
6.3	DEPLOYING THE OPERATING SYSTEM	26
6.3.1	Part 1: Deploying the OS	26
6.3.2	Part 2: Check FBWF	26
6.3.3	Part 3: Run Remote Desktop Connection	27
7	EXERCISE 3: MULTIMEDIA STATION WITH SPLASH SCREEN	29
7.1	BUILD THE CONFIGURATION YOURSELF	29
7.2	DEPLOYING THE OPERATING SYSTEM	30
7.2.1	Part 1: Deploying the OS	31
7.2.2	Part 2: Running Media Player	31
8	EXERCISE 4: PC AS A WIRELESS ACCESS POINT	31
8.1	SETUP XP EMBEDDED IMAGE	32
8.2	CONFIGURE AD HOC NETWORK	32
9	SUMMARY	36
A	BIBLIOGRAPHY	37
A.1	BOOKS	37
B	SJJ EMBEDDED MICRO SOLUTIONS	38
C	SUPPORT	39

1 Introduction

Welcome to the Windows XP Embedded Development Guide for the eBox-IV. Windows XP Embedded is used in a variety of applications from point-of-service, kiosk, ATM machines, gaming machines, industrial controls, robotics, medical test equipment, office automation, set-top boxes, multimedia devices, etc. OEMs take advantage of the XPe's componentization and the Embedded Enabling Features (EEF) to build these embedded devices.

The eBox-IV is an industrial grade PC that provides the core PC technologies that support applications like thin-clients, point-of-sale solutions, multi-media terminals, mini-web-servers, print servers, etc. Best of all, the eBox-IV is enclosed in an industrial grade case that is ready for your application.

This guide will discuss how to create, build, and deploy XP Embedded images for the eBox-IV. Platform and device driver components are included in the kit so you will be able to get to work creating custom images right away. The different exercises build images from the basics to the more complex. Along the way, different techniques and features for custom shells, branding, FBWF, and other features used in XPe will be introduced. Techniques for deploying the image will also be discussed.

1.1 Kit Contents / Other Requirements

Kit Contents

- CD containing:
 - XP Embedded components for the platform and device drivers, and sample configurations for the different demos.
 - Development Guide

Other requirements

- ICOP eBox-IV, containing a fan-less VIA C7 processor, with 512MB system memory and 1GB internal flash drive. If you don't have an internal flash drive you can boot the images from CompactFlash or USB flash disk. USB flash disk will require an additional component called **USB Boot 2.0** in each image.
- 1GB USB Flash disk used for OS transfers.
- Optional: 802.11 Wireless
- PS/2 Keyboard
- PS/2 Mouse
- VGA Monitor
- Full Release of the XP Embedded tools or the 120 day fully functional evaluation version of Windows XP Embedded:

You will need to download Windows XPe Service Pack 2 Feature Pack 2007 and Update Rollup 1.0 from MSDN: <http://msdn.microsoft.com/embedded/windowsxpembedded/default.aspx>

Check the download site:

<http://msdn.microsoft.com/embedded/downloads/xp/default.aspx>

<http://www.microsoft.com/downloads/details.aspx?FamilyID=9bdf1dea-a37e-4d25-83df-aabbaa78914f&displaylang=en>

<http://www.microsoft.com/downloads/details.aspx?familyid=70A79E6F-CE42-45C7-ADD2-52F57959ECF2&displaylang=en>

1.2 Further Reading, Information and Training

Windows XP Pro and Windows XP Embedded are very broad subjects to write about, and the details are outside the scope of a simple getting started guide. This guide is intended as getting started guide for implementing XPe on the eBox-IV. Here are some suggested references for more information on Windows XP Embedded:

- **Books:**

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, 2004, ISBN: 1-932373-96-9. Available at www.sjjmicro.com

- **Community Sites:**

XPe Center @ seanliming.com – www.seanliming.com
XPe Files.com – www.xpefiles.com

- **Training:**

Windows XP Embedded Training Curriculum - Standard 3 Day course covers Windows XPe Service Pack 2 Feature Pack 2007. A development and target PC is required for each student. The course has been designed to support several optional deployment labs. Please see www.sjjmicro.com for more information.

1.3 What's Next?

This guide was designed to be read in sequential order. Each section builds on the previous sections; although the later exercises are independent of each other. If you are more experienced with XPe, you can skip to the sections that are of more importance.

Section 2 covers the development system setup. It will cover installing the XPe software and importing the XPe component for the eBox-IV.

Section 3 will talk about the eBox-IV platform components, so you know what they are and how the XPe components were developed.

Section 4 discusses the OS deployment method provided with this kit. Taking advantage of the new XP Embedded USB Boot 2.0 solution, a USB flash disk with an XPe image will be used to deploy images to the internal flash drive.

Sections 5-8 are the main Exercise sections that will cover building different custom images. Each exercise creates a different image to demonstrate the different projects that one can create. All of the exercises use the hardware platform component, which is the cornerstone for development.

2 Development Setup

Please follow the instructions for setting up the eBox-IV hardware before you perform any of the software setup or exercises.

2.1 Development System Requirements

The development machine requires the following

- Pentium III 500 MHz or equivalent is the minimum, but a Pentium 4 2 GHz or higher is highly recommended.
- 1GB of RAM minimum - 2GB RAM recommended.
- 1GB HDD space available.
- 10/100Base-T Ethernet card (Optional) – for downloading images over an Ethernet connection.
- Windows XP Professional SP1 minimum.
- Windows XPe SP2 with Feature Pack 2007.

Note: You can install the XPe Database on a separate SQL Server for team development. This guide assumes that development tools and database are installed on the same system (Standalone). Please review the on-line help for more information about SQL installation and usage.

2.2 Installing the Tools

The XPe SP2 demo kit and OEM kit come with several CDs. XPe has a sequential installation processes. The installation is organized so you install XPe SP1 first, then install XPe SP2 on top of the XPe SP1, next install XPe Feature Pack 2007 on top of XPe SP2, and finally install Update Rollup 1.0. You will need to download Feature Pack 2007 and Update Rollup 1.0 from the MSDN website. A database is at the center of the development process. You must decide if the database will reside on your local system or on an SQL server. This guide assumes that you are setting up the database on your development system in a standalone configuration.

2.2.1 Part 1: XP Embedded SP1 Setup

Insert CD1 into your development system. The opening installation screen provides the different installation steps.



Step1: Windows Installer for Windows 2000 (only). If your host is a Windows 2000 Professional system, you may need to update Windows Installer to v2.0. During the following steps, you may need to re-boot your Windows 2000 system to complete the installation process.

Step2: Tools setup on the host systems.

Step3: Database Engine Setup: Installs the Microsoft Data Engine on your development system. You will need to reboot the system in order to install the XPe SP1 database in Step 4.

Step4: Database setup: Installs the XPe SP1 database

Step5 (Optional): Remote Boot Setup: If your network environment supports remote boot or PXE server, you can download XPe images to a target with that has a PXE client built into the network card. TFTP and Remote Boot manager services will be installed. These services are intended to be installed on Windows Server 2000 with DHCP server configured and enabled.

2.2.2 Part 2: XP Embedded SP2 Setup

Now that XPe SP1 is installed, you will need to run the setup CD containing the updates for SP2.



Step1: Database Engine Update – This update addressed the vulnerabilities of MSDE.

Step2: Tools Update – Updates several of the development tools. You should reboot the system once you have completed Steps 1 and 2

Step3: Database Update – Updates the database and repositories to include the binaries for XP SP2.

Step4: (Optional): Remote Boot Setup: This version supports Windows Server 2003. If your network environment supports remote boot or PXE server, you can download XPe images to a target which has a PXE client built into the network card. TFTP and Remote Boot manager services will be installed. These services are intended to be installed on Windows Server 2003 with DHCP server configured and enabled.

2.2.3 Part 3: Install Feature Pack 2007

To install Feature Pack 2007 you will need to download the trial version from the MSDN website:
<http://msdn.microsoft.com/embedded/windowsxpembedded/default.aspx>

Check the download site:

<http://msdn.microsoft.com/embedded/downloads/xp/default.aspx>

The download contains an update to Windows XPe Service Pack 2. You will just need to run the installer to install on top of the installation that you have completed so far.



Step1: Tools Update – Updates several of the development tools.

Step2: Database Update – Updates the database and repositories to include the binaries for Feature Pack 2007.

Option: You can also install or extract the API sets fro EWF and FBWF if you are going to write applications that control EWF FBWF.

2.2.4 Part 4: Install Update Rollup 1.0

To install Update Rollup 1.0 you will need to download the trial version from the MSDN website:
<http://msdn.microsoft.com/embedded/windowsxpembedded/default.aspx>

Check the download site:

<http://msdn.microsoft.com/embedded/downloads/xp/default.aspx> - Look for Feature Pack 2007, click on it, and at the bottom of the page is a link to Update Rollup 1.0

You will just need to run the installer to install on top of the installation that you have completed so far.

2.3 Importing Components

2.3.1 Part 1: Extract the Kit Files

The XPe Development CD contains a self extracting zip file with the materials needed to build the configurations.

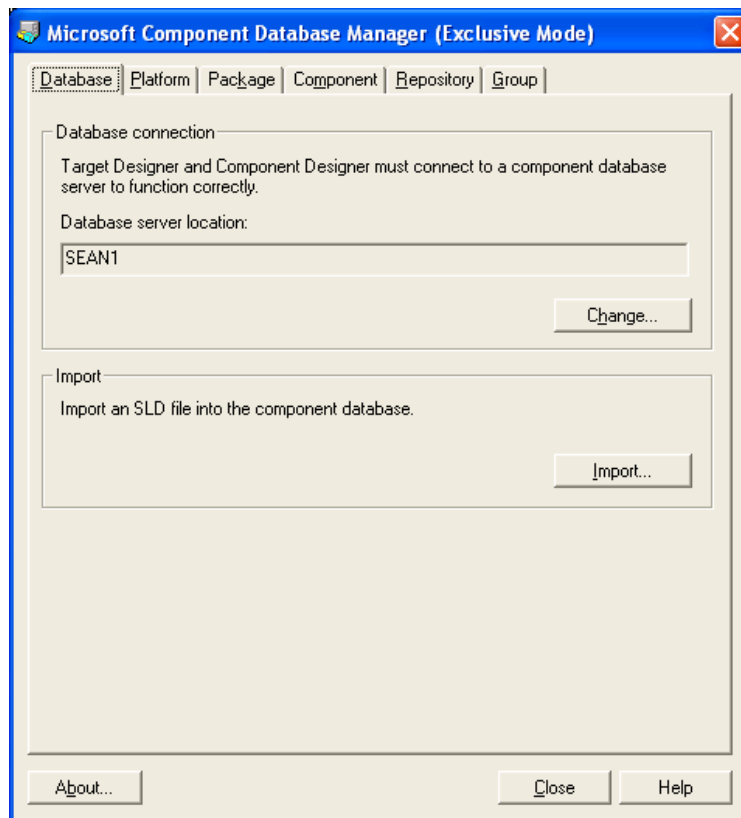
1. **Insert** the CD into the development machine.
2. **Open** File Explorer
3. Right click on the **SJJ602.zip** and select "Extra All" from the context menu to extract the contents to the development machine's hard drive. Take note where you install the files since we will be using them throughout the text.

2.3.2 Part 2: Import the SLD files

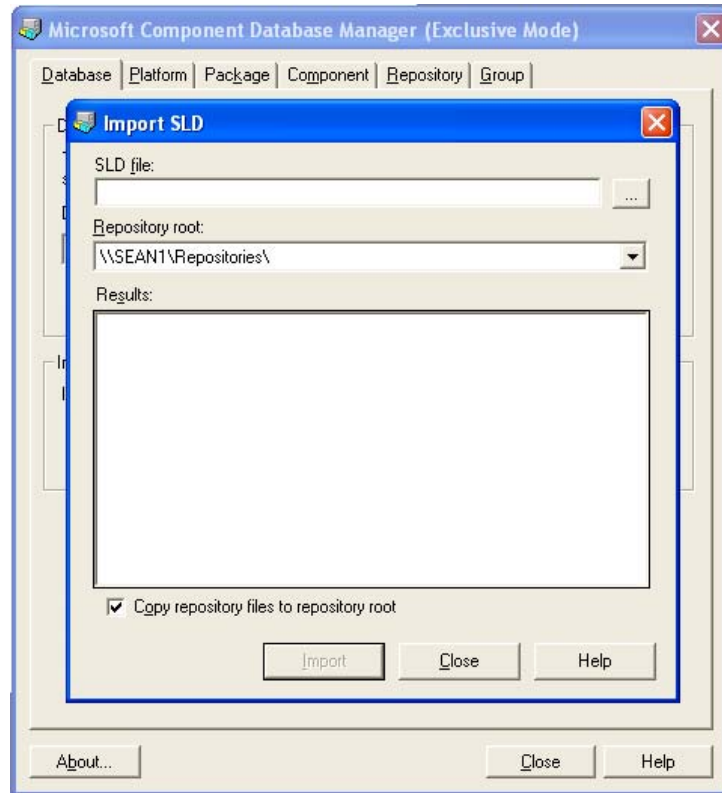
The next step is to import the eBox-IV XPe components into the database. To make this step simple, a single SLD file (SJJ602_TcDK.SLD) has been provided that contains all the components needed for the hardware and exercises.

Note: Make sure that you import the components directly from their folder locations. The various subfolders are linked to the SLD file that is to be imported. These subfolders will be copied into the XPe repository, so Target Designer knows where to get the files when building an image.

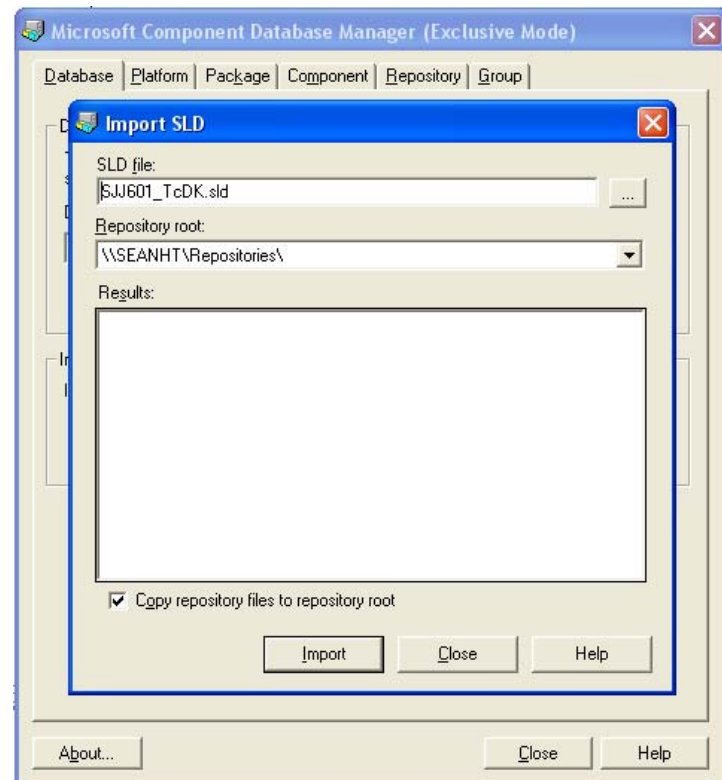
1. Open **Component Database Manager**. CDM opens to the Database tab.



2. Click on the **Import** button. This will open the Import SLD dialog.



3. Click on the (...) button, and locate the SJJ602_TcDK.sld file on the CD.
4. Click Import.



3 Background on the Components

In the last section, you imported the SJJ602_TcDK SLD into the database. The pre-built components were created to speed up the development process. This section discusses some details on the components and how they were created. You can skip this section and move to section 4, if you want to get going and build an image.

The SJJ602_TcDK.SLD contains the following components:

Component	Description	Notes
VIA High Definition Audio - VIA VT3310(UAA) SJJ602 eBoxIV	Audio driver components	
Realtek RTL8139/810x Family Fast Ethernet NIC SJJ602 eBoxIV	Ethernet adapter components	
VIA/S3G UniChrome Pro II IGP SJJ602 eBoxIV (SJJ EMS)	Video driver component	
ICOP_eBox IV 48XX Series	Platform component for the eBox IV 48XX platforms	Platform component can be re-used for different projects.
SJJ602_Transfer	USB boot image	To be used to help deploy the images for the different exercises
SJJ602_Example1, 2, 3, & 4	Three exercise components	
MediaPlayer Shell	Shell for Exercise 4	
RDP ShortCuts	Shortcuts for Exercise 3	
VIA Networking Technologies PCI-Cardbus Wireless LAN Adapter	Wireless network driver component	
Splash Screen SJJ602	Custom Splash screen	

3.1 Where to begin: Target Analyzer

There are over 10,000 components that make up the XPe database, and a majority of those components are related to hardware. The starting point for any new XPe project is to run Target Analyzer on the hardware to capture information about the target hardware configuration and pull in the core components that are required to boot the OS on the hardware. There are two different versions of Target Analyzer: TA.EXE a 16bit MS-DOS application and TAP.EXE a Win32 application. TA.EXE gathers a very small set of information based on sniffing the PCI bus. In contrast, TAP.EXE running in Windows XP Pro, looks inside the registry and finds all the enumerated drivers that are in the system, thus provided the most detailed and complete information on the platform. Both utilities generate a PMQ file that contains the information that the import processor will use to compare against the database.

Windows XP Pro with SP2 and the eBox-IV device drivers were installed on the eBox-IV. TAP.EXE was run on the system to capture all the enumerated hardware details including multimedia streaming support. The resulting component is the ICOP_eBox-IV 48XX Series.

Note: If you don't have the audio driver installed, TAP.EXE will find the audio codec device but not the Microsoft streaming support required for audio. As a result you may have an audio driver that loads correctly in the XPe image but has no audio available because of the missing streaming components.

The platform macro component has been created to save you time, so you do not have to install XP Pro and run TAP.EXE yourself. Any devices that you want to attached to the USB, Parallel, and serial ports will require a component, if there isn't one in the database.

3.2 Missing Driver Components and Support Applications

When a PMQ file is imported into Component Designer, it is a good idea to create a log file. The log file will provide information on what was found in the database, and most importantly, what wasn't found. In the case of the eBox-IV, there were several drivers that were not in the database: audio, video, and Ethernet.

Every XP driver must (or should) have an INF file that is used for setup. Component Designer can import an INF file to generate a driver component. The INF files for each of the missing device drivers were imported to create the various SLD files. In some cases the INF file was part of a setup utility that must be installed in order to obtain the INF file. Component Helper from the *Windows XP Embedded Supplemental Toolkit* was used to capture the INF file, driver files, and any application files during installation. For example, the Via Audio components were generated using Component Helper. The driver component was separated from the extra audio applications that come with the system. The video adapter and Ethernet components were simply generated with the INF files.

3.3 Splash Screen – Device Branding

One of the most commonly requested features is to remove all Windows splash screens during booting, and in some cases, OEMs want their own splash screen to appear for branding purposes. This can be accomplished with the /bootlogo switch in the boot.ini file and a custom boot.bmp file in the \Windows directory.

There is no option to set this boot logo switch from within target designer. The Splash Screen SJJ ICOP component provides an example on how to add your own splash screen without a lot of POST-build / POST-FBA work. The solution adds a new boot_new.ini file to the root of the image. During FBA, the PostFBA.EXE exchanges the boot.ini file created with the Target Designer with the new boot_new.ini file. The boot.bmp file in the \Windows directory will be displayed on the screen during a normal XPe boot.

Note: If your system requires further boot switch settings, add them to the boot_new.ini and re-import the SLD file.

If you want to add your own splash screen, just edit the boot.bmp file with a graphics editor.

3.4 Custom Components: SJJ602_Examples

The exercise macro components are used with each of the exercises. These components contain dependencies on the software layer components

There are also two components that are used in some of the exercises: RDP ShortCuts and MediaPlayer Shell. The RDP ShortCuts component contains two shortcut resources that place shortcut links on the desktop and start bar. The RDP ShortCuts component will be use in Exercise 3. The MediaPlayer shell is an example of a custom shell component that will be used in Exercise 4. Any application can be used as a shell in XPe, including Media Player, so you can use this component as a basis for making your own application as a shell.

4 Image Deployment Setup for USB Boot

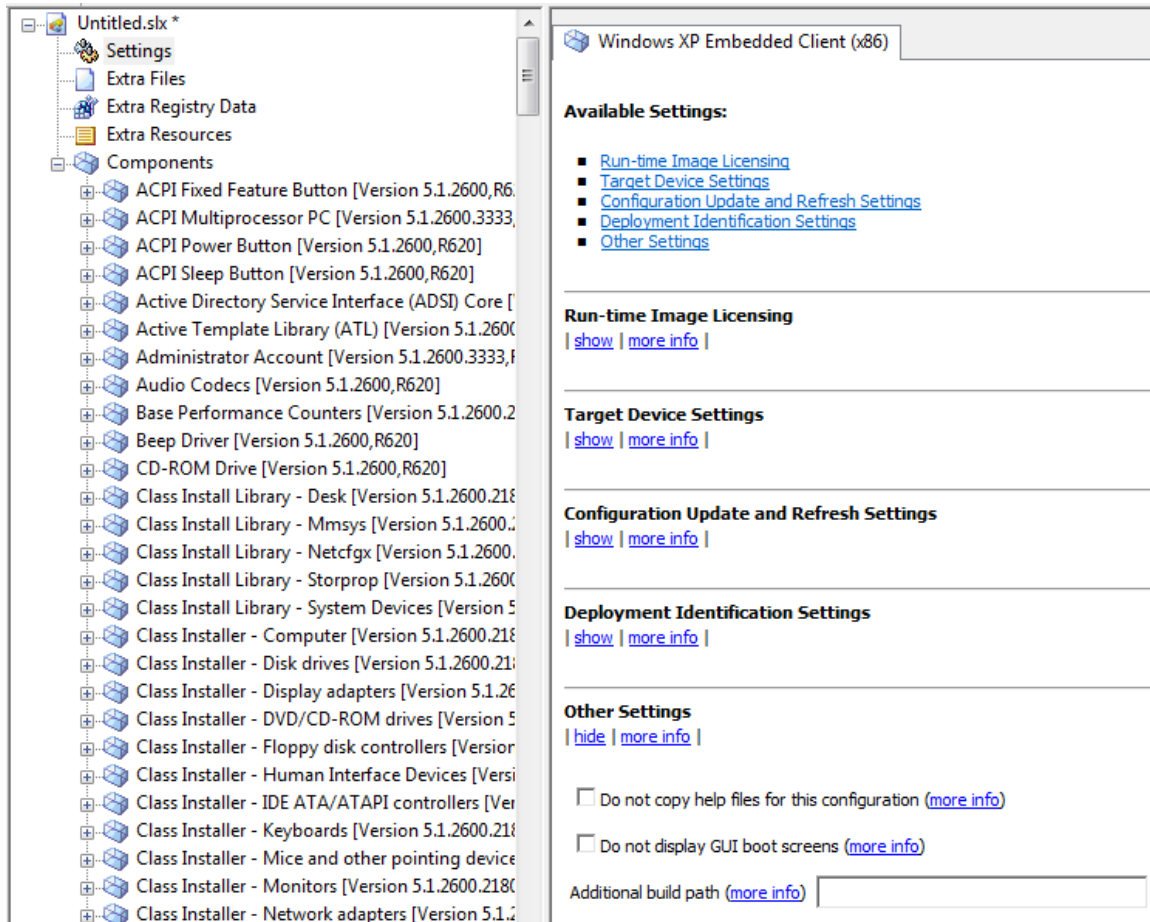
Before we get into the exercises and build images, we need to discuss how to get an XPe image loaded onto the eBox-IV's internal flash drive. Transferring the image from the development system to the target system is truly left up to the OEM. Every target system is different. There are a couple of issues that must be addressed when discussing deployment:

1. XPe images are big, > 35 MB, so a practical deployment method must be used to transfer the OS to the target that can transfer a large image quickly.
2. Long file and folder names **MUST** be preserved. Using XCOPY under DOS will not work. A 32-bit environment or alternative solution must be used to transfer the OS, so that all long filenames and long folder names will be preserved in the copied image.

The eBox-IV doesn't have built in CD ROM or floppy drives, so that rules out certain transfer methods. With Feature Pack 2007, we can take advantage of the XPe USB Boot 2.0 solution and the eBox-IV 48XX's USB flash boot BIOS support. The kit comes with the SJJ602_Transfer macro component that builds the custom XPe image. The resulting XPe image allows deploying an XPe image from a directory on the USB disk or network share.

Here are the steps to create the bootable USB flash disk that will be use to deploy the exercise images in the next section:

1. Make sure that you have imported the SJJ602_TcDK into the XPe database.
2. **Open** Target Designer.
3. Create a new configuration called **XPe Transfer**.
4. Add the **SJJ602_Transfer** component to the configuration. The component can be found in the "SJJ Embedded Micro Solutions TcDK SJJ602" category.
5. Now we need to edit some of the settings for the components we just added. At the very top of the configuration, under XPe Transfer, click on **Settings**.
6. The details pane on the far right should change. Click on the "**Show**" under Other Settings.
7. Uncheck **Do not copy help files for this configuration**. The S3 video driver that comes with the kit requires the help system to be in place.



8. The next step is to run a **dependency check**. You can run Check Dependencies several ways: from the menu bar: Configuration->Check Dependencies, click on the Check Dependencies on the tool bar, or hitting F5. Check Dependencies looks at the configuration to make sure that all the components in the configuration have their dependent component or component group relationships satisfied. The Check Dependencies should finish with no errors or warnings. Make sure *Auto-resolve dependencies* is selected in Target Designer options.
9. The final step is to **build the image**. Like Check Dependencies, building the image can be initiated several different ways: from the menu: Configuration->Build Target Image..., click on the hammer in the tool bar, or hit F7.
10. A build dialog will appear. The destination should be set to the "Windows Embedded Images" folder. Keep all the defaults and click on the **Build** button.
11. If another dialog appears asking to delete the contents in the "Windows Embedded Images" folder, click the Yes button. The image should build with 1 warning. The warning is to indicate that the image is a test image and is only good for approximately 120 days.
12. **Save** the configuration.

The next step is to prepare the flash disk using the new UFDprep.exe utility.

13. Insert the flash disk in to the development PC.
14. Open a command window
15. Run the **UFDprep.exe** utility, which can be found under "\Program Files\Windows Embedded\Utilities":

c:\>ufdprep <drive>, where <drive> is the drive letter for the USB flash disk you want to format

Note: *If you are developing in Windows Vista, then you don't need to run UFDprep.exe. You only have to format the USB flash disk under Windows Vista. Windows Vista lays down the correct boot sector, but you must rename the XPe image's NTLDR file to BOOTMGR or the image will fail to boot.*

16. Once the format has completed, **copy** the XPe image found in the **Windows Embedded Images** folder to the USB flash disk.
17. Undock the USB flash disk.
18. Insert the USB flash disk into the target.
19. Boot the target and hit the DEL key to enter the BIOS setup.
20. In the BIOS Settings, set the boot order to boot from the USB flash disk before the hard drive.
21. Save the settings, and exit the BIOS setup.
22. Let the system boot from the USB flash disk. The system will run through FBA and the XPe build will boot.
23. Create a subdirectory on the flash disk called **XPe Image**. This subdirectory will be used to transfer the local image. The USB flash disk is now ready for use.
24. Power down the target system and remove the USB flash disk.

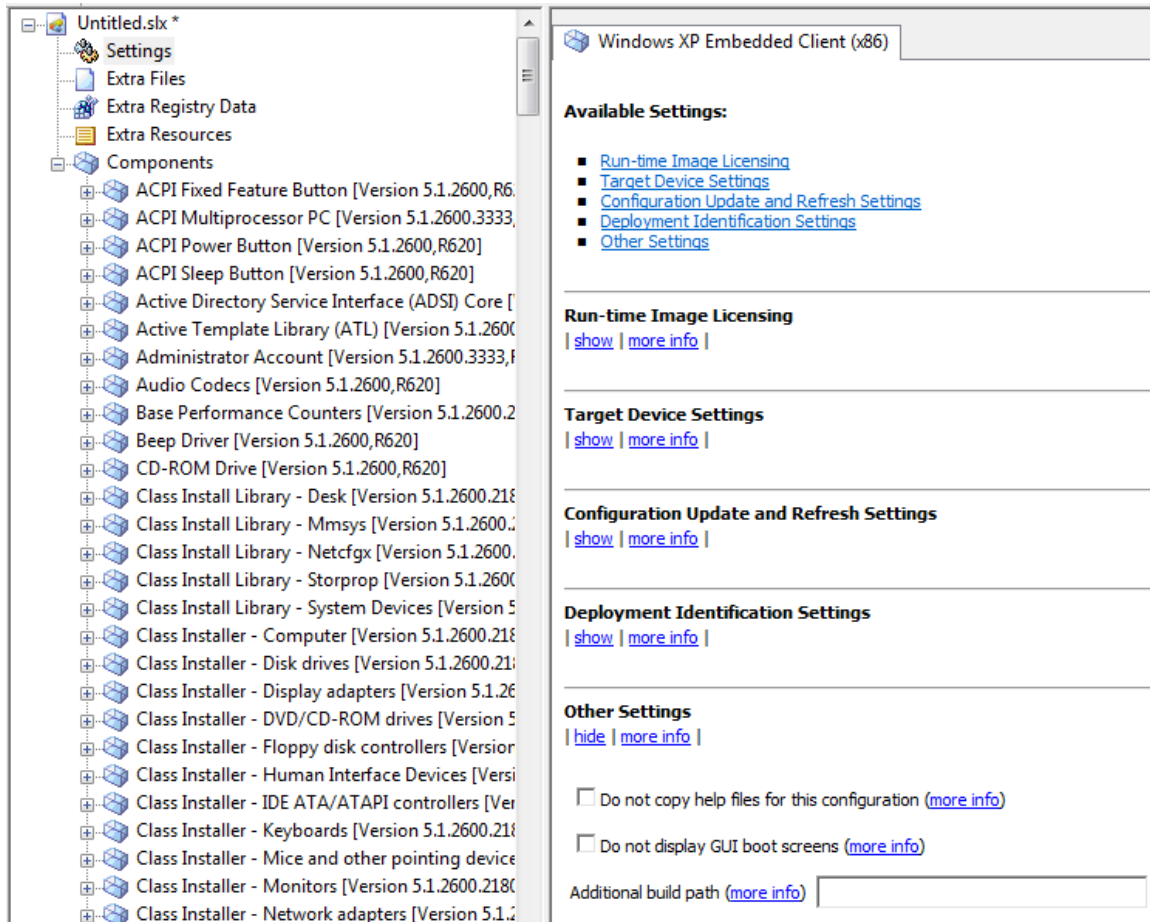
5 Exercise 1: Creating a Basic Image

The first exercise is to build a very simple and very basic XPe image, so you can get familiar with building and deploying the OS image. The shell will be the Command Shell, and we will use the ICOP_eBox-IV 48XX Series as the platform component. You should have completed the importing of the SJJ602_TcDK.SLD file in section 2 and created the transfer image for the USB drive in section 4 before starting this exercise.

5.1 Build the Configuration Yourself

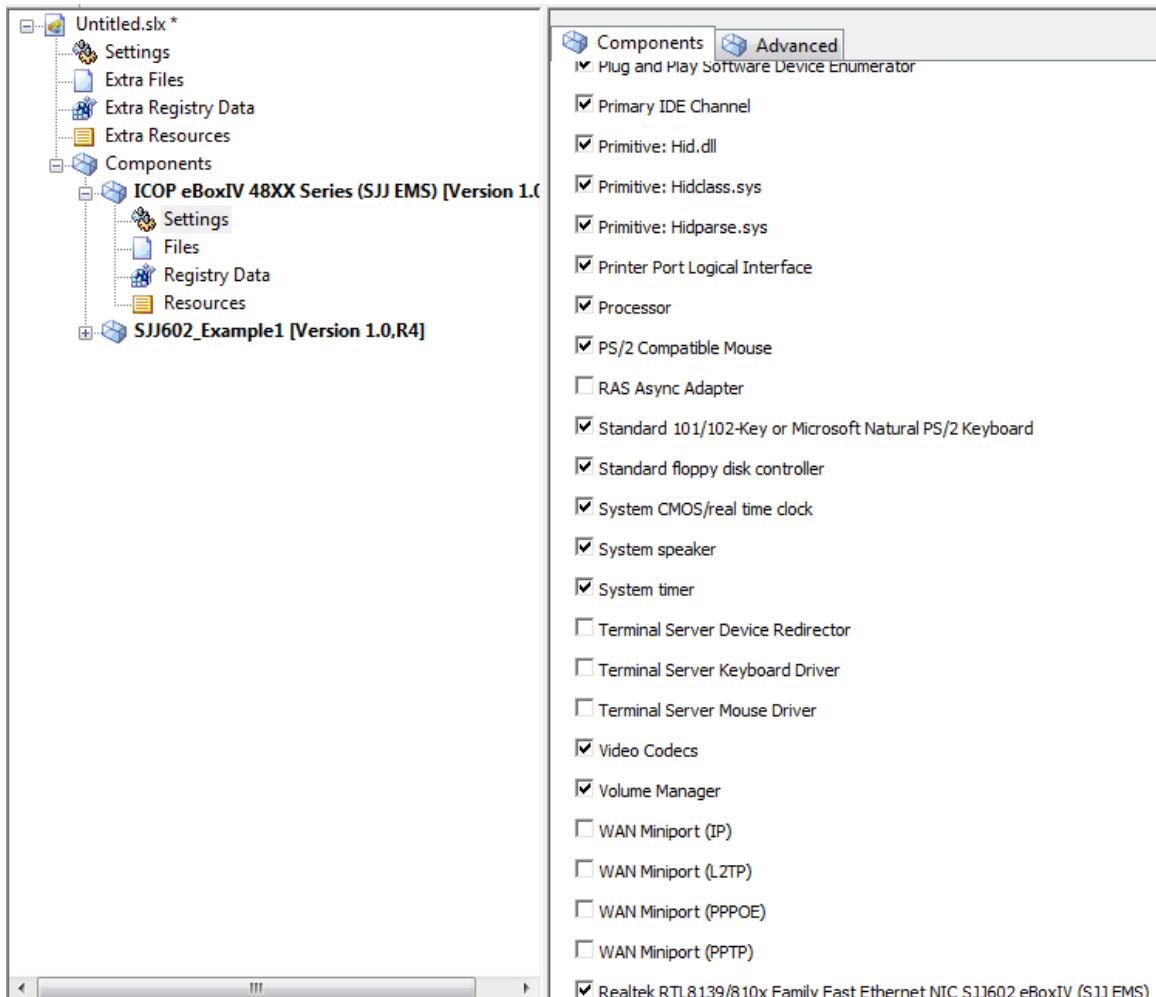
Target Designer is used to create and build the custom XPe images. In this section we first create a new configuration, add components to the configuration, make some setting changes, resolve all component dependencies, and then build the image.

1. Open Target Designer.
2. From the menu, select **File->New**.
3. The New configuration dialog appears. Type in the new configuration name: **602_Exercise1**.
4. Using the Component Browser on the left-hand side, add the following components:
 - **ICOP_eBox-IV 48XX Series** (Located under: Software->Test & Development->Hardware->Platforms).
 - **SJJ602_Example1** (Located under SJJ Embedded Micro Solutions TcDK SJJ602).
5. Now we need to edit some of the settings for the components we just added. At the very top of the configuration, under 602_Example1, click on **Settings**.
6. The details pane on the far right should change. Click on the **"Show"** under Other Settings.
7. Uncheck **Do not copy help files for this configuration**. The S3 video driver that comes with the kit requires the help system to be in place.

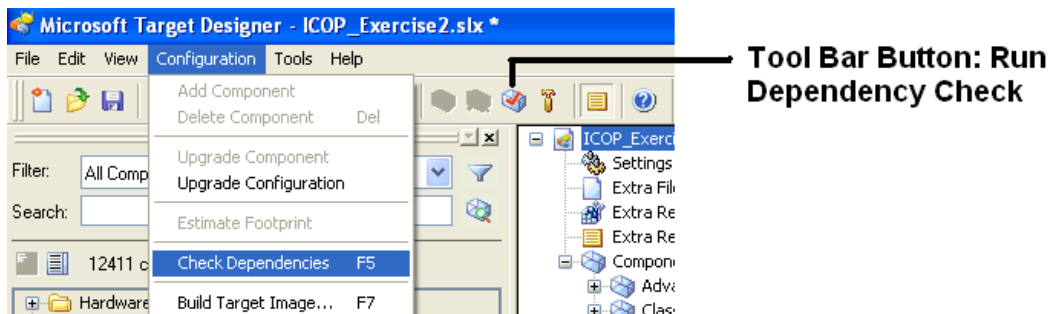


8. Select **Settings** under the ICOP_eBox-IV 48XX Series component
9. To create smaller images you will want to remove items that are not needed in an image. In this case, we will remove some hardware components before we perform a Check Dependencies. The details pane on the right will change to the settings for the ICOP_eBox-IV 48XX Series component. Uncheck the following items:

- Communications Port
- Printer Port
- Realtek RTL8139 Ethernet Adapter
- Via HD Audio Controller
- Via High Definition Audio – both components
- RAS Async Adapter
- Terminal Server Device Redirector
- Terminal Server Keyboard Driver
- Terminal Server Mouse Driver
- WAN Miniport (IP)
- WAN Miniport (L2TP)
- WAN Miniport (PPPOE)
- WAN Miniport (PPTP)



10. Now that the changes have been made, the next step is to run a dependency check. You can run Check Dependencies several ways: from the menu bar: **Configuration->Check Dependencies**, click on the Check Dependencies on the tool bar, or hit **F5**. Check Dependencies looks at the configuration to make sure that all the components in the configuration have their dependent component or component group relationships satisfied. The Check Dependencies should finish with no errors or warnings. Make sure *Auto-resolve dependencies* is selected in Target Designer options before running Check Dependencies.



11. The final step is to build the image. Like Check Dependencies, building the image can be initiated several different ways: from the menu: **Configuration->Build Target Image...**, click on the hammer in the tool bar, or hit **F7**.

12. A build dialog will appear. The destination should be set to “Windows Embedded Images” folder. Keep all the defaults and click on the **Build** button.
13. If another dialog appears asking to delete the contents in the “Windows Embedded Images” folder, click the **Yes** button. The image should build with 1 warning. The warning is to indicate that the image is a test image and is only good for approximately 120 days. The image should be around 118MB in size.

5.2 Deploying the Operating System

The USB flash disk with XPe image created in section 4 will be used to deploy the OS image to the target system's internal flash disk.

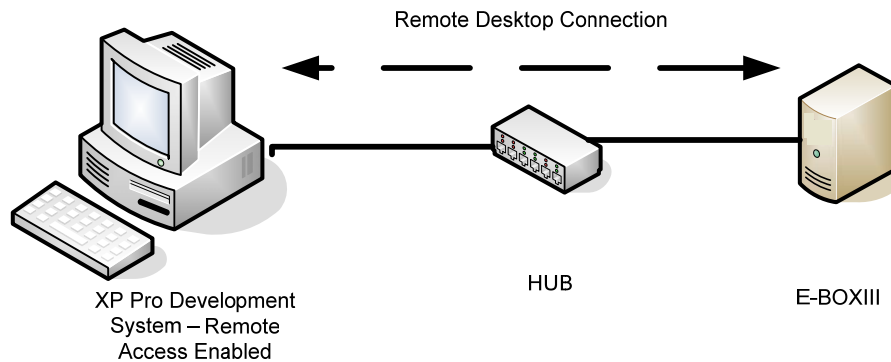
1. Insert the USB flash disk that has the XPe transfer image into the development system.
2. **Copy** the 602_Exercise1 XPe image found in \Windows Embedded Images folder to the **XPe Image** folder on the USB flash disk. Copy the full OS image directory tree.
3. **Unmount** the USB flash disk.
4. Insert the USB flash disk into the eBox-IV 48XX.
5. **Turn on** the power to the eBox-IV 48XX.
6. Once the system boots to the XPe image, go to the Control Panel->Administrative Tools-Computer Management->**Disk Management**.
7. **Format** the D drive with FAT or NTFS.
8. **Copy** the image in the XPe image folder to the D drive. Copy the full OS image directory tree.
9. When the copy is completed, **power down** the system.
10. **Remove** the USB flash disk.
11. **Turn on** the power to the eBox-IV 48XX and let the system boot the internal flash disk. The system will boot the XPe image and start FBA.

The system boots the Command Shell. There is not much that the image can do, but we successfully demonstrated how to build an image and download the image to the target system.

6 Exercise 2: Thin Client Image with FBWF

The eBox-IV is an ideal platform for industrial thin client use. Now that we are familiar with building and deploying images to the eBox-IV, let's build a basic thin client image that contains the Remote Desktop Connection application, which is one of the most popular thin client applications on the market. The RDC will connect to your XP Pro development workstation that has Remote Desktop access enabled. Explorer is going to be the shell. A special component called the RDP ShortCuts provides the shortcuts that will appear on the desktop and Start bar. The two shortcut resources are good examples of how to create shortcuts in Component Designer.

To complete the exercise, you will need to have an Administrator or Power user account with a password on the development station. You will also need to know the TCP/IP address of the XP Pro system. This lab assumes that the network supports DHCP.



The image will also include the File Base Write Filter (FBWF) and the Registry Filter. FBWF protects a partition from write access, but maintains the writes in an overlay so the partition still looks read/writable. You can also create write through sections so data can be written to the disk, but still have the OS protected. FBWF helps to protect the OS from corruption. A command line utility, FBWFMGR.EXE, can be used to control the state of FBWF: enable, disable, and commit changes. The Registry Filter helps to preserve the TSCAL and domain secret key information when FBWF is running. Please see the help files and various white papers about FBWF and the Registry Filter.

6.1 FBWF Manager Console Application

Here are the different options for using FBWFMGR.EXE

Syntax:

```
fbwfmgr [/? | /help /[switch] | /displayconfig | /overlaydetail | /enable | /disable | /addvolume
[volumename] | /removevolume [volumename] [1|0] |
/addexclusion [path] | /removeexclusion [path] |
/setthreshold [threshold] | /setcompression [1|0] | /setpreallocation [1|0] ]
```

Table 3 lists the option commands:

Switch	Description
displayconfig	Displays all configuration information for the write filter including protected volumes list, overlay configuration and write-through paths. The command returns:

Switch	Description
	<p>State—Indicating current filter state (enable or disable) and state for next boot.</p> <p>Protected Volumes—List of protected volumes including the current and next boot state.</p> <p>Compression—Current and next boot state for cache compression.</p> <p>Threshold—Current and next boot values for the overlay cache threshold.</p> <p>Write-Through Paths—Displays a complete list of active and next boot Write-Through paths.</p> <p>Pre-allocation Status—Displays current and next boot status for cache pre-allocation.</p>
overlaydetail	<p>Displays detail on the current overlay contents for all protected volumes. The command returns:</p> <p>Contents—Files and folders currently in the overlay for all protected volumes including sizes (size of data in overlay) and open file handles.</p> <p>Memory Usage—Total amount of memory being consumed by the overlay.</p>
enable	Enables the write filter on the next restart.
disable	Disables the write filter on the next restart.
addvolume	Adds a volume to the protected volume list for next boot.
removevolume	Removes a volume from the protected volume list for next boot.
addexclusion	Adds a Write-Through path to the exclusion list for next boot.
removeexclusion	Removes a write-through path from the exclusion list for next boot.
setthreshold	Sets the overlay threshold value for next boot.
setcompression	Sets overlay compression as enabled (1) or disabled (0) for next boot.
setpreallocation	Sets cache pre-allocation as enabled (1) or disabled (0) for next boot.
commitfile	Commits a specified file.
restorefile	Restores a specified file.
?	Displays usage and help.
help / [switch]	Displays help information for a specific FBWF Manager switch.

Table 2 – FBWFMGR Command Options

The following are the Field descriptions:

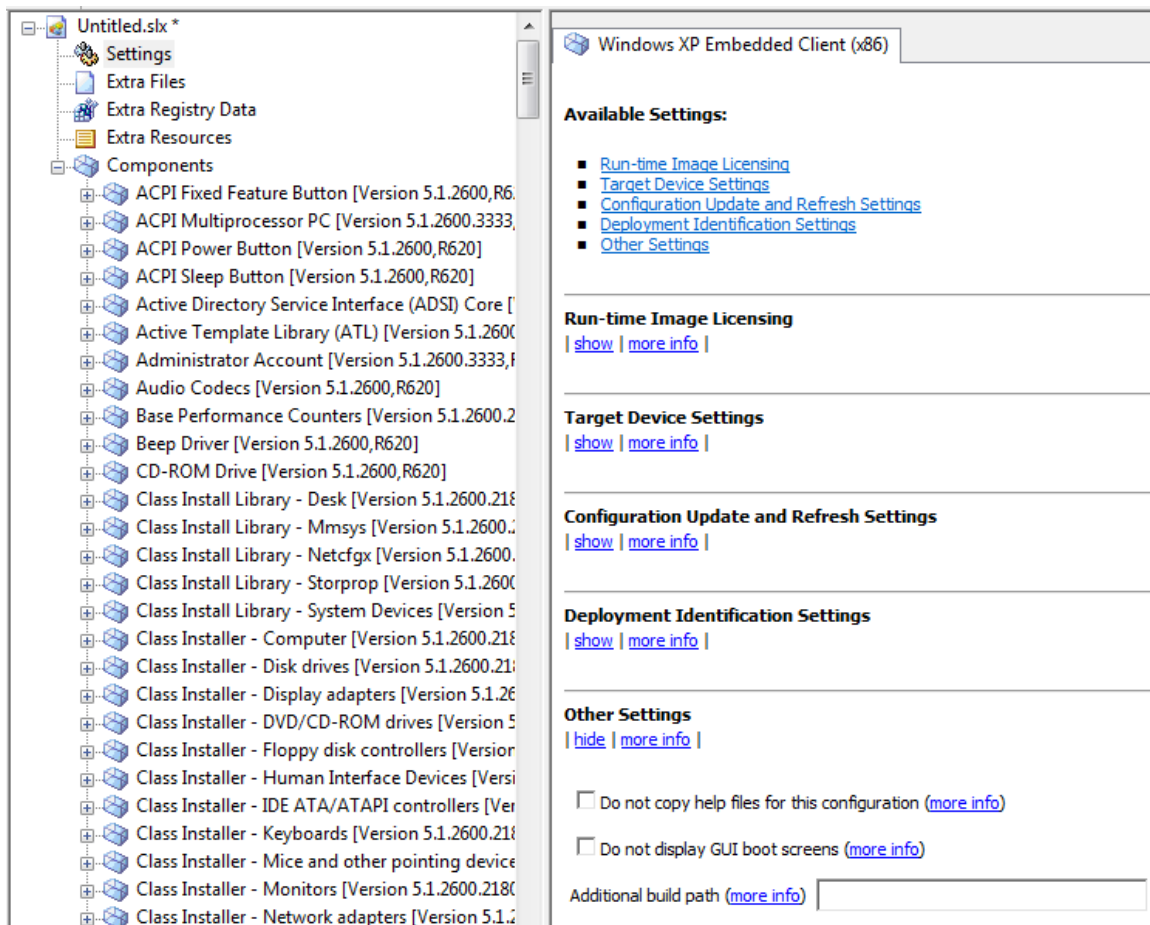
Field	Meaning
volumename	Full path to a volume
1	Remove exclusion list
0	Preserve exclusion list
path	Full file or directory path
threshold	Overlay threshold in MB

Table 3 – FBWFMGR Field Descriptions

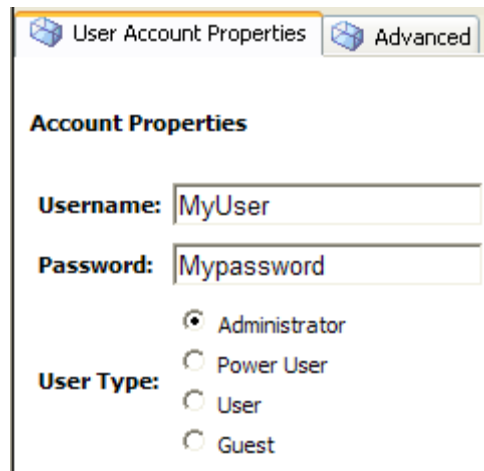
6.2 Build the Configuration Yourself

Example 1 was a simple image, and this exercise will be a little more involved. In this example we will build a Thin Client platform that will connect to the desktop system via RDP. The image will boot to the Explorer shell and shortcuts for Remote Desktop Connection will be available to connect to the remote PC. FBWF will be setup to protect the life of the flash disk and will be controlled using FBWFMGR.EXE.

1. Open **Target Designer**.
2. From the menu, select **File->New**.
3. The New configuration dialog appears. Type in the new configuration name: **602_Exercise2**.
4. Using the Component Browser on the left hand side, add the following components:
 - **SJJ602_Exercise2** (Located under SJJ Embedded Micro Solutions TcDK SJJ602).
5. Now we need to edit some of the settings for the components we just added.
6. Now we need to edit some of the settings for the components we just added. At the very top of the configuration, under 602_Example2, click on **Settings**.
7. The details pane on the far right should change. Click on the “**Show**” under Other Settings.
8. Uncheck **Do not copy help files for this configuration**. The S3 video driver that comes with the kit requires the help system to be in place.



9. In the configuration, locate and expand the **User Account** component and click on **Settings**.
10. Edit the following property settings with account information that matches the user account on the desktop system that the target will connect to. The picture below is just an example. Make sure to use the appropriate Username and Password for your system.



The screenshot shows the 'User Account Properties' dialog box with the 'Advanced' tab selected. The 'Account Properties' section contains the following fields and options:

- Username:** A text box containing 'MyUser'.
- Password:** A text box containing 'Mypassword'.
- User Type:** A group box containing four radio button options:
 - ☒ Administrator
 - ☐ Power User
 - ☐ User
 - ☐ Guest

- UserName: <Enter the account name>
- Password: <Enter the accounts password>
- User Type: Administrator

11. Under the File Based Write Filter component – set the protected volume to C:

FBWF Configuration Settings	
Advanced	
File Based Write Filter Configuration	
Enable FBWF	<input checked="" type="checkbox"/>
Cache Type	Pre-allocated
Maximum Cache Size in MB (If nothing is specified, the value will default to 64MB)	
Virtual size display mode	<input type="checkbox"/>
Disable page-file support	<input checked="" type="checkbox"/>
Disable System Restore	<input checked="" type="checkbox"/>
Disable Background Disk Defragmentation	<input checked="" type="checkbox"/>
Disable Low Disk Warning Notification	<input checked="" type="checkbox"/>
Number of Protected Volumes	1
Protected Volume # 1	
Volume e.g. C:	C:
Write-Through Files and Folders (separated by newline), for example: \Documents and Settings\All Users\Shared Documents\My Music \Windows\temp	

12. Now that the changes have been made, the next step is to run a dependency check. You can run Check Dependencies several ways: from the menu bar: **Configuration->Check Dependencies**, click on the Check Dependencies on the tool bar, or hitting **F5**. Check Dependencies looks at the configuration to make sure that all the components in the configuration have their dependent component or component group relationships satisfied. The Check Dependencies should finish with no errors or warnings. Make sure *Auto-resolve dependencies* is selected in Target Designer options before running Check Dependencies.
13. The final step is to build the image. Like Check Dependencies, building the image can be initiated several different ways: from the menu: **Configuration->Build Target Image...**, click on the hammer in the tool bar, or hit **F7**.
14. A build dialog will appear. The destination should be set to "Windows Embedded Images" folder. Keep all the defaults and click on the **Build** button.
15. If another dialog appears asking to delete the contents in the "Windows Embedded Images" folder, click the **Yes** button. The image should build with 1 warning. The warning is to indicate that the image is a test image and is only good for approximately 120 days. The image should be around 197MB in size.

6.3 Deploying the Operating System

The USB flash disk with XPe image created in section 4 will be used to deploy the OS to the target system's internal flash disk.

6.3.1 Part 1: Deploying the OS

1. Insert the USB flash disk that has the XPe transfer image into the development system.
2. On the USB flash disk, delete any previous image found in the XPe Image folder. Delete the full OS image directory tree.
3. **Copy** the 602_Exercise2 XPe image found in **Windows Embedded Images** folder to the XPe Image folder on the USB flash disk. Copy the full OS image directory tree.
4. **Unmount** the USB flash disk.
5. Insert the USB flash disk into the eBox-IV 48XX.
6. **Turn on** the power to the eBox-IV 48XX.
7. Once the system boots to the XPe image, go to the Control Panel->Administrative Tools-Computer Management->**Disk Management**.
8. **Format** the D drive with FAT or NTFS.
9. **Copy** the image in the **XPe Image** folder to the D drive. Copy the full OS image directory tree.
10. When the copy is completed, **power down** the system.
11. **Remove** the USB flash disk.
12. **Turn on** the power to the eBox-IV 48XX and let the system boot the internal flash disk. The system will boot the XPe image and start FBA.

6.3.2 Part 2: Check FBWF

You should be able to run CMD.EXE from start->run and then run FBWFMGR.EXE to check the status of FBWF. The status should show that FBWF is enabled

```
C:\Windows\System32>fbwfmgr
```

```
File-based write filter configuration for the current session:
```

```
filter state: enabled.  
overlay cache data compression state: disabled.  
overlay cache threshold: 64 MB.  
overlay cache pre-allocation: disabled.  
protected volume list:  
  \Device\HarddiskVolume1  
Write-Through list of each protected volume:  
  \Device\HarddiskVolume1:
```

```
File-based write filter configuration for the next session:
```

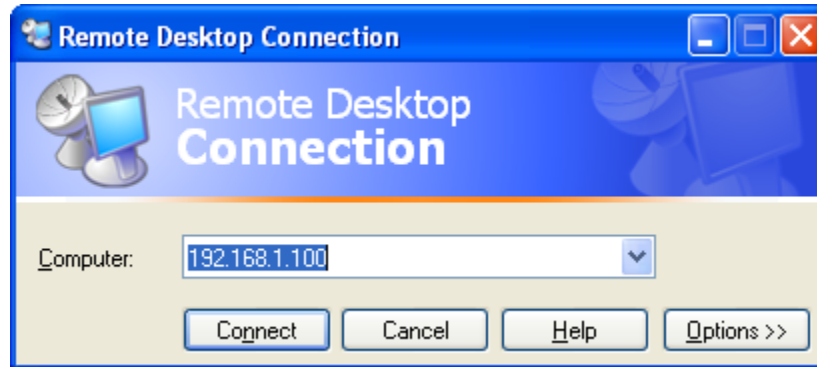
```
filter state: enabled.  
overlay cache data compression state: disabled.  
overlay cache threshold: 64 MB.  
overlay cache pre-allocation: disabled.  
protected volume list:  
  \Device\HarddiskVolume1  
Write-Through list of each protected volume:  
  \Device\HarddiskVolume1:  
    \Data
```

Try creating a file or folder on the system, and then rebooting the target to see if the file or folder remains. The file or folder should be lost since it was stored in the RAM overlay and lost during the reboot. You can temporarily disable FBWF using FBWFMGR.EXE in order to add programs or make changes to the system.

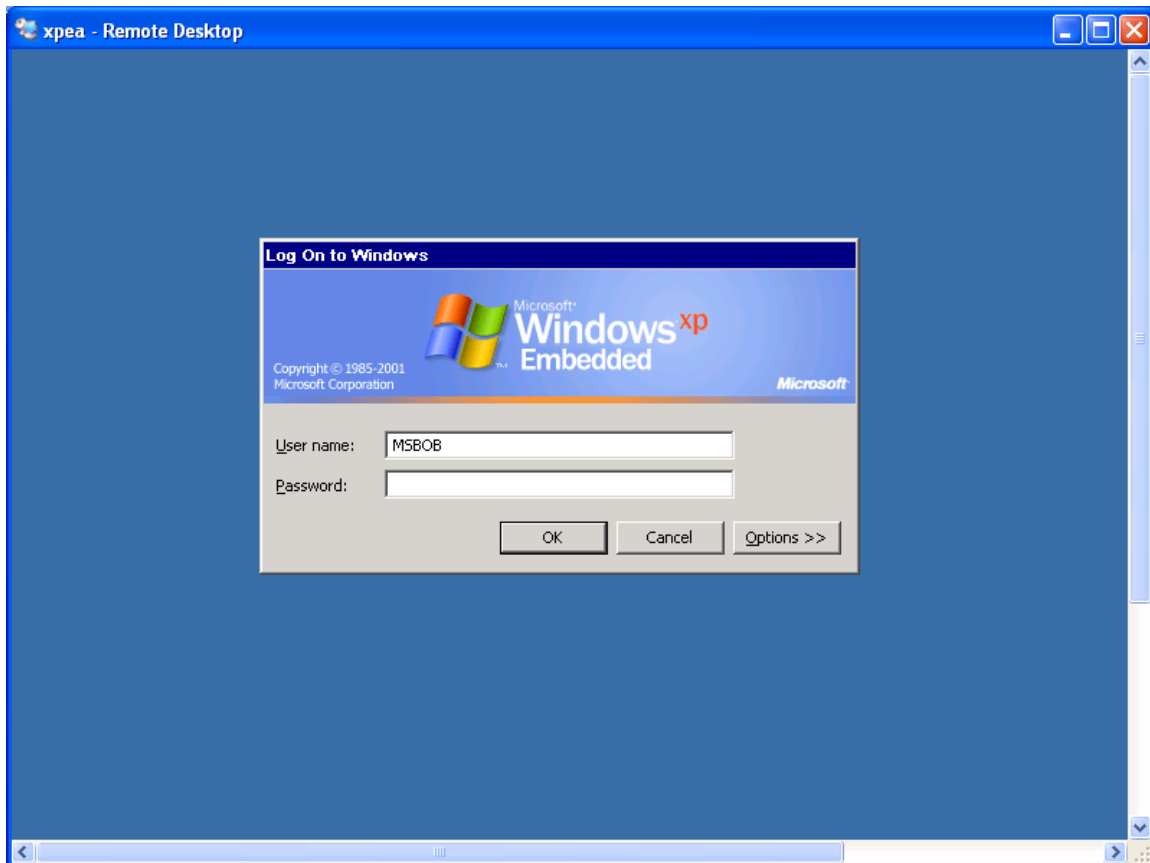
6.3.3 Part 3: Run Remote Desktop Connection

Now let's try the Remote Desktop Connection.

1. On the development system, use network settings or IPCONFIG /all to get the TCP/IP address of the development system.
2. On the eBox-IV, click on the Remote Desktop link found on the desktop.
3. A dialog appears asking for the computer name. Type the TCP/IP address of the development system and click the **Connect** button.



4. The remote connection is established, and the remote window appears. A Windows Logon dialog box appears in the remote window. Type in the user account and password to access the system.



5. The Remote Desktop logs in, and you will notice that the logon screen appears on the target system. XP only supports one connection to a machine at a time. If you logon to the target locally, the remote desktop connection would be terminated.

7 Exercise 3: Multimedia Station with Splash Screen

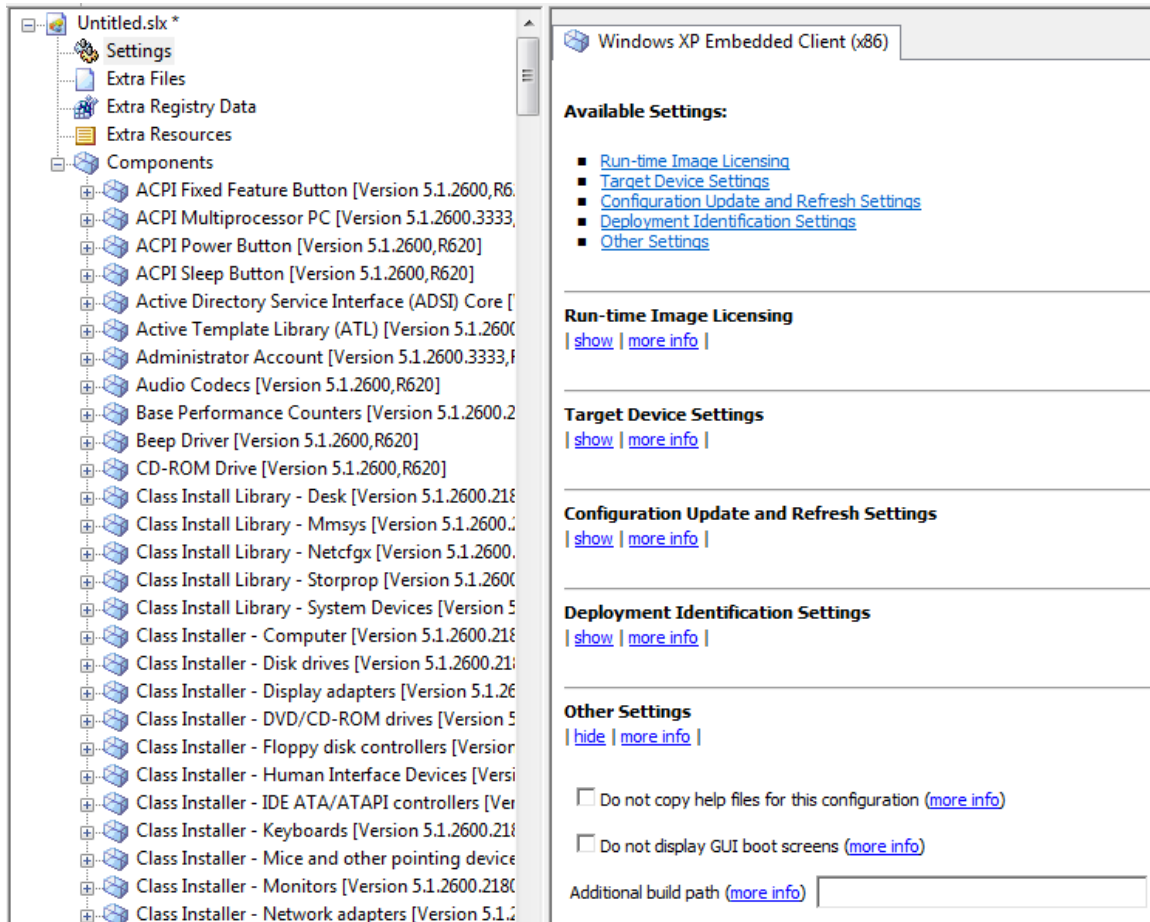
The eBox-IV has an integrated AC97 for sound. The size of the eBox-IV makes the platform ideal for a small multimedia solution. We will now use the full platform macro component in this exercise to playback music from MediaPlayer. To make this image a little different, we will use a custom shell: MediaPlayer Shell. There is also an example of a Splash screen to replace the standard Windows splash screen. All you have to do is add the SJJ_SPLASH screen component; the FBA resource will replace the BOOT.INI with one that supports the splash screen.

You will need to connect speakers or headphones to the audio out port for this exercise.

7.1 Build the Configuration Yourself

The following are the steps to re-create the configuration.

1. Open **Target Designer**.
2. From the menu, select **File->New**.
3. The New configuration dialog appears. Type in the new configuration name: **602_Exercise3**.
4. Using the Component Browser on the left hand side, add the following components:
 - **SJJ602_Exercise3** (Located under SJJ Embedded Micro Solutions TcDK SJJ602).
5. Now we need to edit some of the settings for the components we just added. At the very top of the configuration, under 602_Exercise3, click on **Settings**.
6. The details pane on the far right should change. Click on the **"Show"** under Other Settings.
7. Uncheck **Do not copy help files for this configuration**. The S3 video driver that comes with the kit requires the help system to be in place.



6. Now that the changes have been made, the next step is to run a dependency check. You can run Check Dependencies several ways: from the menu bar: **Configuration->Check Dependencies**, click on the Check Dependencies on the tool bar, or hitting **F5**. Check Dependencies looks at the configuration to make sure that all the components in the configuration have their dependent component or component group relationships satisfied. The Check Dependencies should finish with no errors or warnings. Make sure *Auto-resolve dependencies* is selected in Target Designer options before running Check Dependencies.
7. The final step is to build the image. Like Check Dependencies, building the image can be initiated several different ways: from the menu: **Configuration->Build Target Image...**, click on the hammer in the tool bar, or hit **F7**.
8. A build dialog will appear. The destination should be set to "Windows Embedded Images" folder. Keep all the defaults and click on the **Build** button.
9. If another dialog appears asking to delete the contents in the "Windows Embedded Images" folder, click the **Yes** button. The image should build with 1 warning. The warning is to indicate that the image is a test image and is only good for approximately 120 days. The image should be around 303MB in size.

7.2 Deploying the Operating System

Once the image has been built you might want to include some media files for testing the image playback.

7.2.1 Part 1: Deploying the OS

1. Insert the USB flash disk that has the XPe transfer image into the development system.
2. On the USB flash disk, delete any previous image found in the XPe Image folder. Delete the full OS image directory tree.
3. **Copy** the 602_Exercise3 XPe image found in Windows Embedded Images to the XPe Image folder on the USB flash disk.
4. **Unmount** the USB flash disk.
5. Insert the USB flash disk into the eBox-IV 48XX.
6. **Turn on** the power to the eBox-IV 48XX.
7. Once the system boots to the XPe image, go to the Control Panel->Administrative Tools-Computer Management->**Disk Management**.
8. **Format** the D drive with FAT or NTFS.
9. **Copy** the image in the XPe image folder to the D drive. Copy the full OS image directory tree.
10. When the copy is completed, **power down** the system.
11. **Remove** the USB flash disk.
12. **Turn on** the power to the eBox-IV 48XX and let the system boot the internal flash disk. The system will boot the XPe image and start FBA.

7.2.2 Part 2: Running Media Player

1. Once FBA has completed, the image should boot to the MediaPlayer.
2. You may be asked a few setup options for MediaPlayer. Follow the Wizard's instructions and choose the defaults.
3. With the speakers or headphones attached to the eBox-IV, select a song to play.

Note: Closing media player will not force Windows to re-launch media player. You will have to reboot the system to bring back Media Player.

Note: Use the Power switch to turn on and off the system.

8 Exercise 4: PC as a Wireless Access Point

We constantly look for unique techniques and ideas for Windows XP Embedded. One area that we have not explored specifically is wireless capabilities. As more systems are remotely connected via wireless, a flexible access point would be ideal to link the systems together. This requires a system best suited for application, and the eBox III and IV series offer the right size, performance, and price. The wireless add-on module for the eBox series combined with the internal wired Ethernet support opens up wireless bridging opportunities.

New wireless technologies seem to come out every year. We have 802.11 a, b, g, and now n. Every time one of these new technologies reaches the market, new hardware such as access points and wireless adapters have to be purchased to take full advantage of the new technology. Creating a wireless access point using a PC make some sense since a PC offers the most flexible hardware platform. With Windows XP Embedded, you can build a wireless access point (WAP) using off the shelf software and device drivers. Any upgrades in technology can be added to the system since the system is a PC running the Windows operating system.

You could also add support for local data storage where data and multimedia files can be shared as a network storage device. You can also add support to manage the system remotely by adding a few networking features:

- Web Server

- FTP Server
- Remote Desktop Server
- Telnet Server
- SSH



eBox IV with Wireless Networking Module

The eBox-IV-48XX shown above has two wired 10/100Base-T Ethernet ports and wireless 802.11b/g support. You can easily create a wireless router. With the USB ports, you could add an 802.11n module configured for full speed mode to have a wireless access point that support the latest wireless technologies.

8.1 Setup XP Embedded Image

A custom image can be built that supports everything needed to create an ad hoc access point.

1. Open Target Designer
2. Add the ICOP_eBox-IV 48XX Series (SJJ EMS) platform macro component
3. Add the SJJ602_Example4
4. Make sure that you have enabled the help system.
5. Optional components to include would be for remote management services such as:
 - Terminal Services
 - Telnet Server
 - Terminal Server Remote Desktop
6. Optional: Add the USB Boot 2.0 component if you are going to boot from USB flash disk.
7. Run dependency check
8. Build the image
9. Deploy the image to the target and let the image run through FBA.

8.2 Configure Ad Hoc Network

Once the system has completed FBA, the following steps can be used to setup the ad hoc access point. First the Wireless gets configured with an ad hoc network connection and then a wired connection provides access to the Internet for all users to access the system.

1. Open **Control Panel**
2. Open the **Network Connections** control panel.
3. Right click on the wireless adapter and choose **Properties** from the context menu.

4. Click on Wireless Networks Tab and make sure that “Use Windows to configure my wireless network settings” is checked. The default is checked.

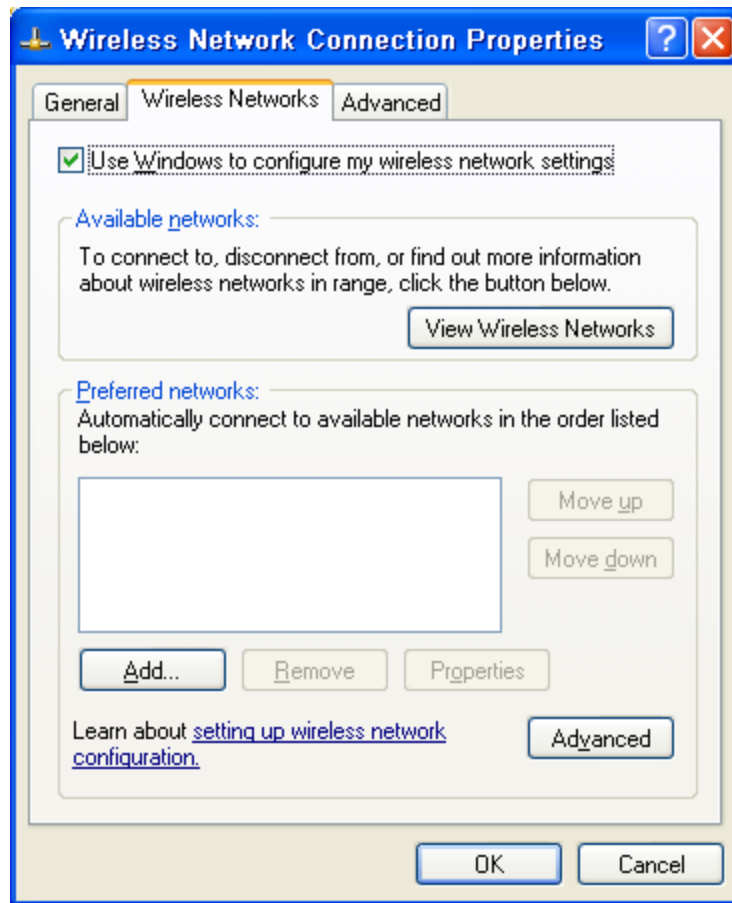


Fig 8.1 Wireless Properties

5. Click on **Advanced** button
6. Select “Computer-to-computer (ad hoc) networks only”

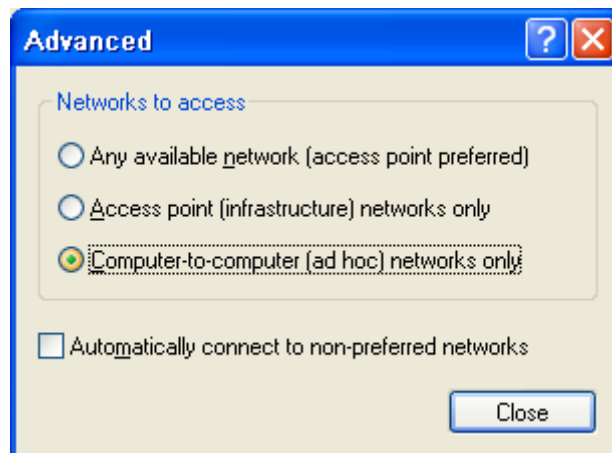


Fig 8.2 Set Ad Hoc Access

7. Under preferred networks click **Add**

8. Add hot spot name or SSID
9. Set Network Authentication to **Shared**.
10. Set Data Encryption: **WEP**

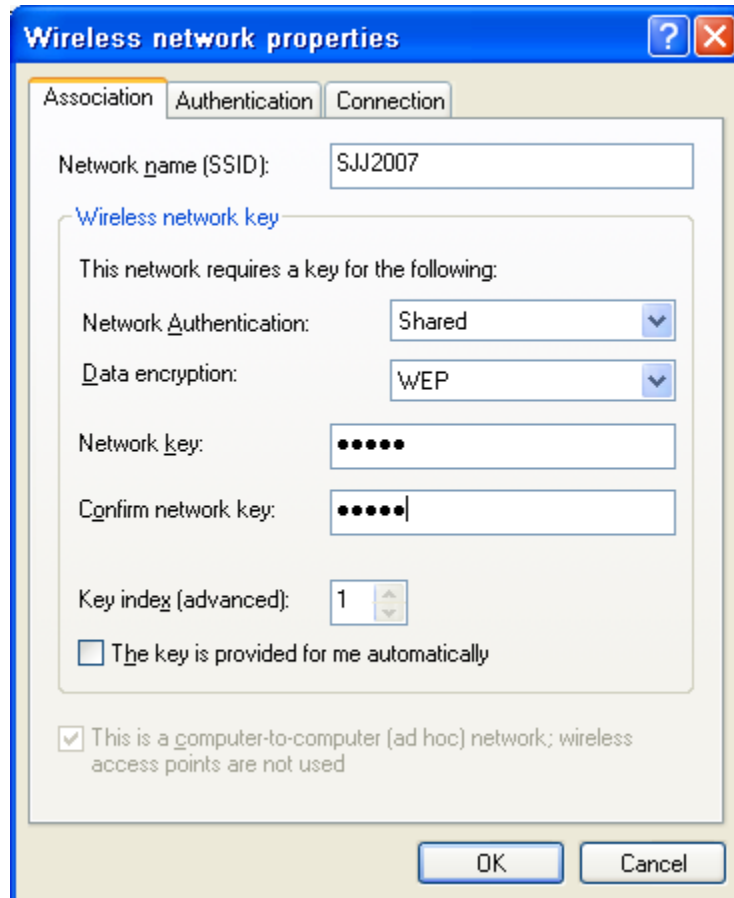


Fig 8.3 Create the Ad Hoc Network Name and Network Key

11. **Uncheck** the option "The key is provided for me automatically"
12. Create a network key and confirm - make sure you follow password rules
13. Click **OK** and you should see hot spot listed.

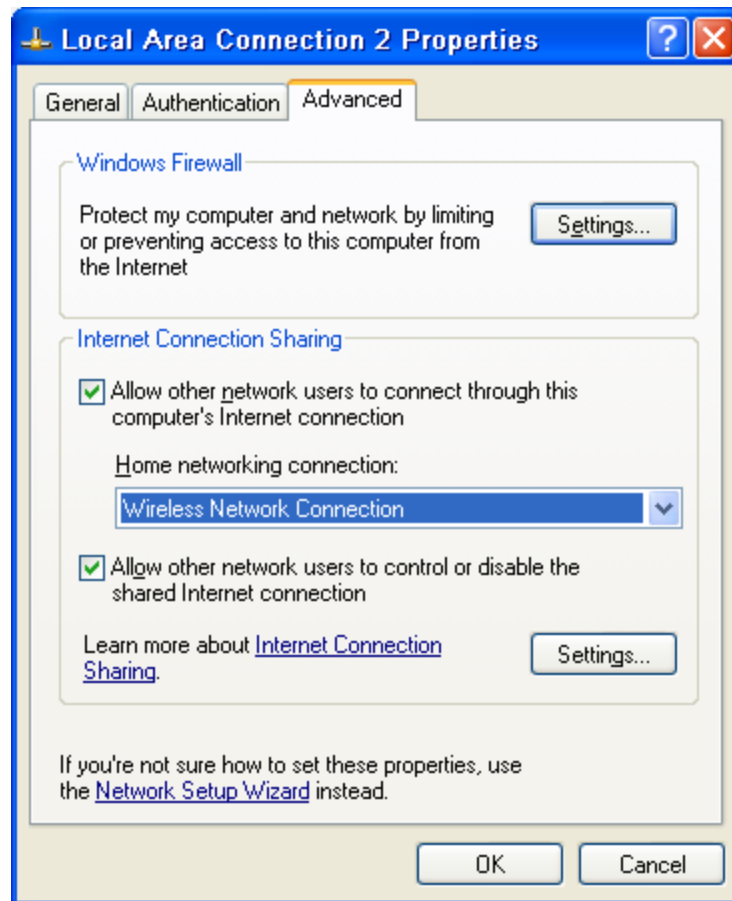


Fig 8.4 Enable Internet Connection Sharing in a Wired Connection

14. Right click on a wired connection and select properties.
15. Select Advanced tab
16. Set the ICS to check the box to "Enable others to connect through this internet connection".
17. Set Home networking connection to **Wireless Network Connection**

On another PC with wireless connectivity, you should now be able to connect to the ad hoc access point to access the Internet. You will need the wireless network key to complete the connection.

The same steps above can be applied to any Windows XP Pro system, but once a system is open to share the Internet any one can access the local files if things are not locked down. With XP Embedded, you can take additional steps to create custom security policies, custom firewall settings, unique splash screens, and custom shells to lockout users from certain files, folders, and system settings. Also, you can run XP Embedded headless without a video, keyboard, or mouse, and have it automatically reply to system messages. With XP Embedded, there is more flexibility and control that can be applied to an OS image running WAP.

There have been attempts to create software that makes a PC a WAP device. These applications have come and gone over the years. If you find any, please let us know, and we will update this documentation. By enabling an ad hoc implementation in XP Embedded, you can support future communication technologies.

9 Summary

This guide provided the basic setup, building steps, and deployment methods to get XPe running on the eBox-IV. The kit itself with pre-built components that address the core device drivers of the eBox-IV allows you to get a jump start into XPe development. With the base components already completed, you only need to create components for the peripheral devices that you want to attach to the eBox-IV to create these unique devices.

The exercises in this guide are a few brief examples of what can be done with the eBox-IV and XPe. There are many other concepts and features to explore with the eBox-IV and XPe, such as remote management, headless operation, mini-webserver, print servers, wireless connectivity, just to name a few.

A Bibliography

A.1 Books

Windows NT Embedded Step-By-Step, Sean D. Liming, Annabooks, 2000, ISBN: 0-929392-68-X

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, 2004, ISBN: 1-932373-96-9.

B SJJ Embedded Micro Solutions

SJJ Embedded Micro Solutions is dedicated to making learning embedded tools and solutions easier. Our books, toolkits, and training courses provide a hands-on approach to learning embedded systems. Our consulting services complement our products by offering support for all aspects of your embedded design hardware, software, manufacturing, and life-cycle of the product.

Developing an embedded system is more than just picking a few components and building an image. Developers have to take into account how the system is used, manufactured, and supported in the field. We take an architectural approach to deliver the right solution for your product. We have experience in a variety of markets; our previous XP Embedded projects have included:

- Thin clients
- Gaming Consoles
- Industrial Controls
- Voice over IP Systems
- Test Equipment
- Consumer Electronics
- Automotive

If you are looking for support on your Windows CE or XP Embedded project, we can support your project in different areas:

- Hardware design review and consultation
- OAL and Bootloader Development
- Develop CE or XP device drivers
- Create and integrate components for applications and device drivers
- Architect configurations for customer's hardware
- Implement security policies
- Enterprise connectivity and integration
- Implementation of XPe embedded features such as EWF, DUA, Cloning, Remote boot, CD-Boot, CF, SMS, and SUS, and USB flash boot.
- Application development including .NET, EWF APIs, and XPe Power Management APIs
- Develop architecture and build documentation
- Training for engineering staff

Please contact us for more information on consultation and availability.

Web: www.sjjmicro.com

E-mail: sales@sjjmicro.com

C Support

Please contact SJJ Embedded Micro Solutions (support@sjmicro.com) for Windows XP Embedded training, educational material, or support.

Please contact ICOP for any hardware questions or technical assistance.
<http://www.embeddedpc.net/>