SJJ600 Industrial Developers Kit (IDK) for Windows® XPe

User Guide

Developed by: SJJ Embedded Micro Solutions, LLC., San Diego Ca.

First Printing: January 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

Table of Contents

1 Welcome to the IDK

Windows XP Embedded is a special version of Windows XP Professional that allows Original Equipment Manufactures (OEMs) to customize the XP operating system (OS) for their embedded product. The IDK provides the software, components, and examples to quickly get a system-up and running with Windows XPe. Some of the features are:

- EMAC INC' PCM-9375
  - AMD Geode LX800
  - 4 serial ports
  - Parallel port
  - 4 USB Ports with 2 ports brought out externally
  - Dual Ethernet
  - VGA out
- IDK supports Windows XP Embedded Service Pack 2 Feature Pack 2007
- XPe FP2007 USB Boot 2.0 supported and is used for transfer of the XPe image via Ethernet
- XPe components for all platform device drivers
- Exercises for quickly building XPe images

1.1 About the IDK

The IDK provides the foundation to get started building OS images on the PCM-9375. The exercises in the documentation provide examples of images that can run on the hardware. Although the IDK is a teaching tool, more detail information on developing Windows XPe images with Target Designer and Component Designer can be found in several locations:

- Book: *Windows XP Embedded Advanced*
- Book: *Windows XP Embedded Supplemental Toolkit* - covers XPe SP2
- XPe Center @ seanliming.com – www.seanliming.com

1.2 Terms

Various acronyms will be used throughout this document. The table below provides information about some of the terms most commonly used in this document:

Term / Acronym	Description
ACPI	Advanced Computer and Power Interface
API	Application Programming Interface
CF	Compact Flash
FBA	First Boot Agent – Every XP Embedded image must go through FBA in order to complete the build processes
EWF	Enhanced Write Filter - EWF allows the XPe operating system to boot from read-only media such as CD-ROMs, write-protected hard disks, or flash media. Any write to the disk will be diverted to an overlay. Disk overlays will be used in this project.
FBWF	File Base Write Filter – Filter that allows for developer defined files to write through to an otherwise protected disk.
HORM	Hibernate Once, Resume Many – a combination of EWF and the hibernation capability within XPe to boot the OS as fast as possible.
IDE	Integrated Device Electronics
IE	Internet Explorer
OS	Operating System
RAM	Random Access Memory
SDK	Software Development Kit

Term / Acronym	Description
SID	Security Identification – Each XP OS image must have a unique SID for security.
SP1 and SP2	Service Pack 1 and Service Pack 2 – Service Packs are updates to the Windows XP Operating System that add features, functionality, and fixes to the original release of the Windows XP Operating System.
VB	Visual Basic
XPe	Microsoft Windows XP Embedded

1.3 Development System Requirements

Windows Embedded Studio for Windows XPe requires the following minimum hardware requirements for the development system:

- Pentium 4 - 2 GHz or higher is highly recommended
- 1GB RAM or higher
- 1GB HDD
- 10/100Base-T Ethernet card - for downloading images over an Ethernet connection.
- Monitor, Mouse, Keyboard
- Windows XP Pro SP2

1.4 Hardware Requirements

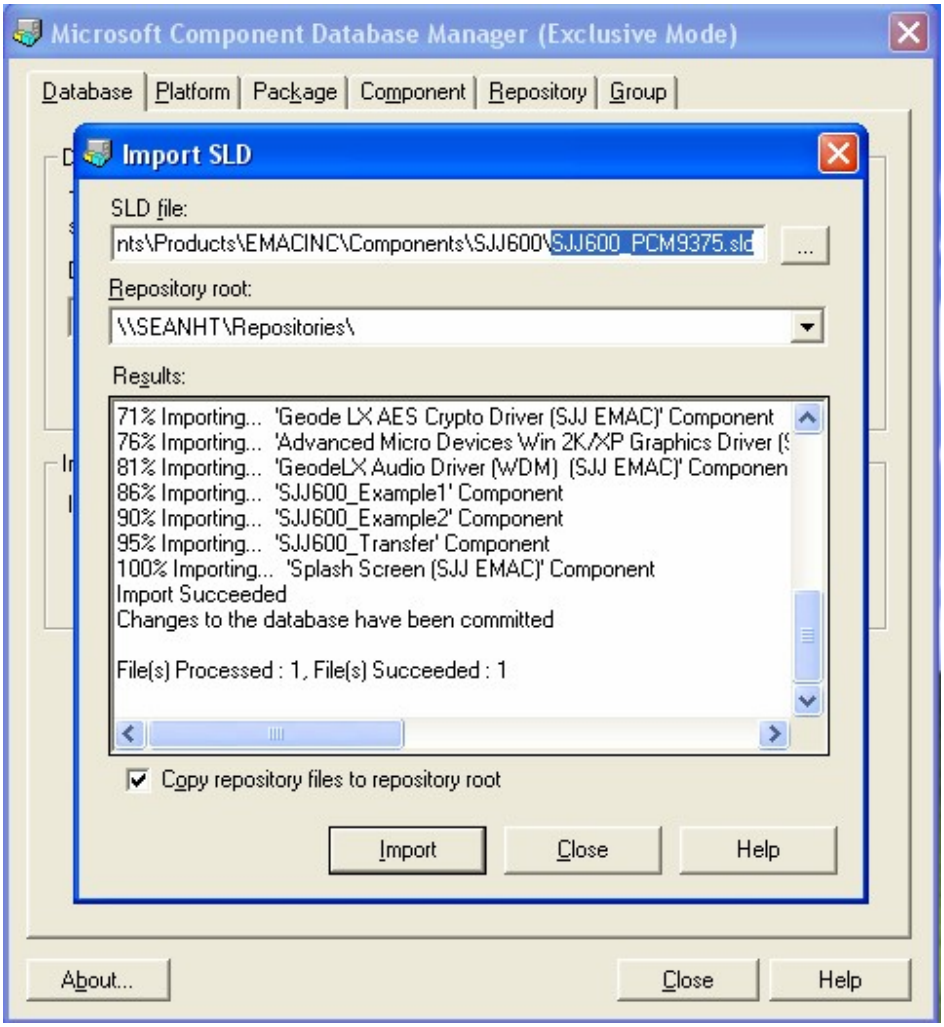
The SJJ600 IDK provides the target platform to run the XPe image, but you will need to provide the following:

1. VGA Monitor
2. PS/2 keyboard
3. PS/2 mouse
4. USB 2.0 Flash disk - used for transferring the image. Memorex and Sandisk flash drives recommended.

2 IDK Installation

The CD contains a zip file for the components and files necessary to be imported into the XPe database. The following instructions explain how to install the IDK components.

1. Make sure that you have installed Windows XP Embedded tools and have the database active and running.
2. On the system with the XPe database, copy the SJJ600.ZIP file from the CD to your computer.
3. Extract the Zip file
4. Open Component Database Manager
5. Import the SJJ600_PCM9375.sld. This SLD contains all the components and exercises. The next section discusses the SJJ600_PCM9375.SLD file in more detail.

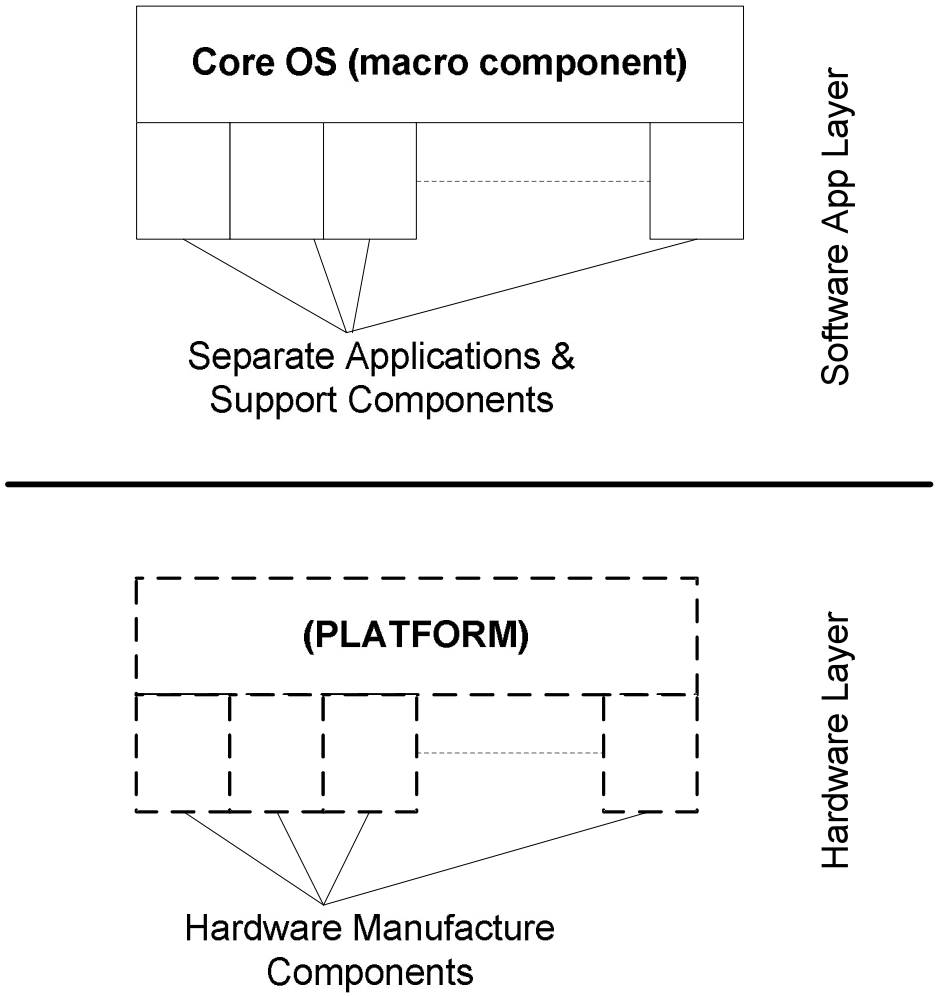


6. After the import process is completed, close Component Database Manager

3 SJJ600 Component Architecture and Descriptions

There are over 11,000 components in the XP Embedded database. To make rebuilding the OS image simpler and to achieve the goals of the project, macro components are used to group multiple individual components into a single block. The database categories are divided into two major groups: hardware and software. The hardware category contains all the device driver and hardware layer components needed to boot the OS on the hardware. The software category contains the Core OS feature components that add functionality to the image such as file system support and various application support components.

By grouping individual components into a few macro components, the architecture is flexible so the Core OS can support different hardware platforms simply by changing the platform macro component.



Software / Hardware Implementation

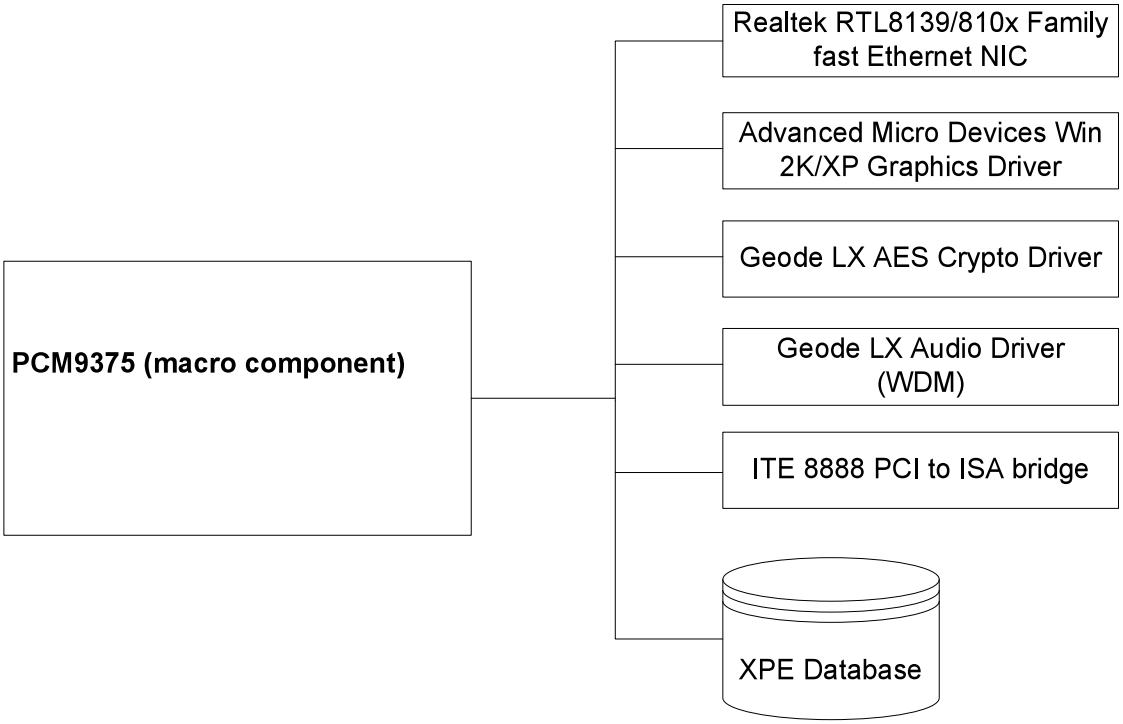
There is also the added benefit, when developing a new configuration, that only 2 Macro components are needed instead of tracking down a large number of different components to start the new XPe configuration.

3.1 SJJ600_PCM9375.SLD

The SJJ600_PCM9375.SLD file contains all the device driver components, platform macro component, and example components for the SJJ600 IDK. All device driver components have a (SJJ EMAC) in the name to help distinguish these custom components from those that come in the database.

3.1.1 PCM9375

The PCM9375 is the platform macro component for the PCM9375. It was created with the help of Target Analyzer. The missing device driver components are part of the SJJ600_PCM9375.SLD and are component dependencies for the PCM9375.



PCM9375 Platform Component

3.1.2 Realtek RTL8139 and RTL8169 Components

There are 5 Realtek Ethernet device driver components. They were created from the INF file that came with the driver set for the PCM9375. Only one component is needed for the PCM9375. The others are extras.

3.1.3 ITE 8888 PCI to ISA Bridge

PCI to ISA bride controller driver created from INF file.

3.1.4 GeodeLX AES Crypto Driver

Device driver component for the AES encryption / decryption hardware created from INF file.

3.1.5 Advanced Micro Devices Win 2K/XP Graphics Driver

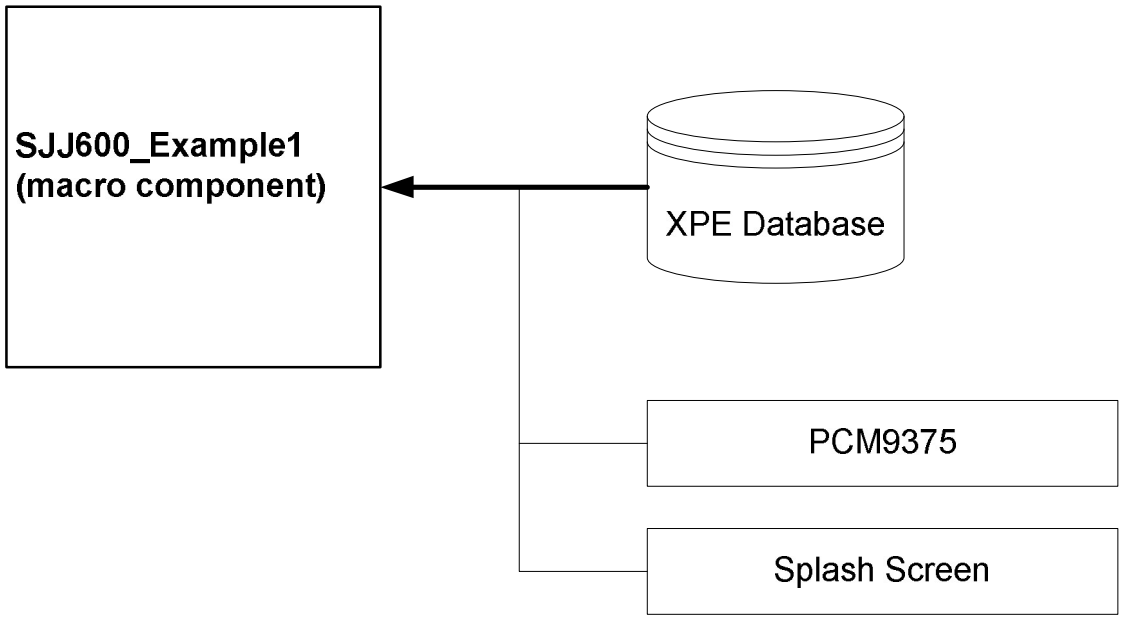
Video driver component for the built in video graphics engine created from INF file

3.1.6 GeodeLX audio Driver (WDM)
Audio driver component for built in audio controller created from INF file.

3.1.7 Splash screencomponent
Replaces the Windows Logo with a custom splash screen. The component has a special application that runs during FBA that switches out the Boot.INI file built with Target Designer with a new BOOT.INI file that has the boot switches to replace the splash screen. The boot.bmp must be 640x480 4BPP (16 colors)

3.1.8 SJJ600_Transfer
This macro component will build a USB Boot 2.0 bootable image. The image is intended for booting the PCM9375, preparing the CF, and sharing the internal CF drive so an image can be downloaded.

3.1.9 SJJ600_Example1
This is a macro component that builds an XPe image with Explorer shell, Telnet Server, and Terminal Server Remote Desktop support. The SJJ_Example1 follows the model described at the beginning of this section with one difference. PCM9375 is a dependency component, which makes it easier to build the image.

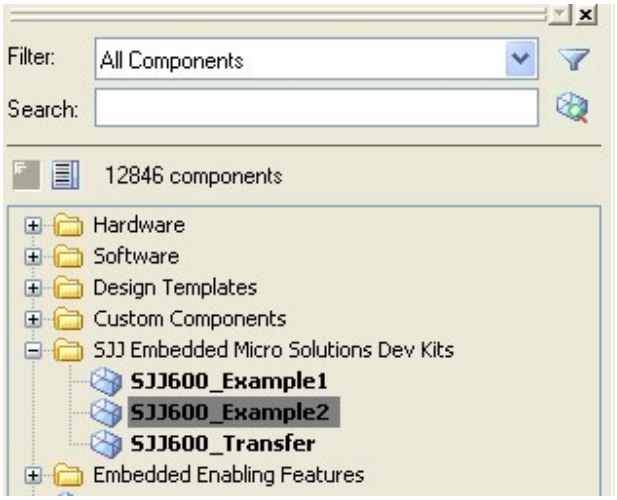


SJJ600_Example1 Component Dependencies

3.1.10 SJJ600_Example2
This component contains all the components needed to build the image. Unlike SJJ600_Example1, this macro component is an example of a smaller XPe image. Audio support has been removed, and Command Shell and Minlogon are used in place of Explorer shell and Windows Logon Standard. The network driver component has been disabled

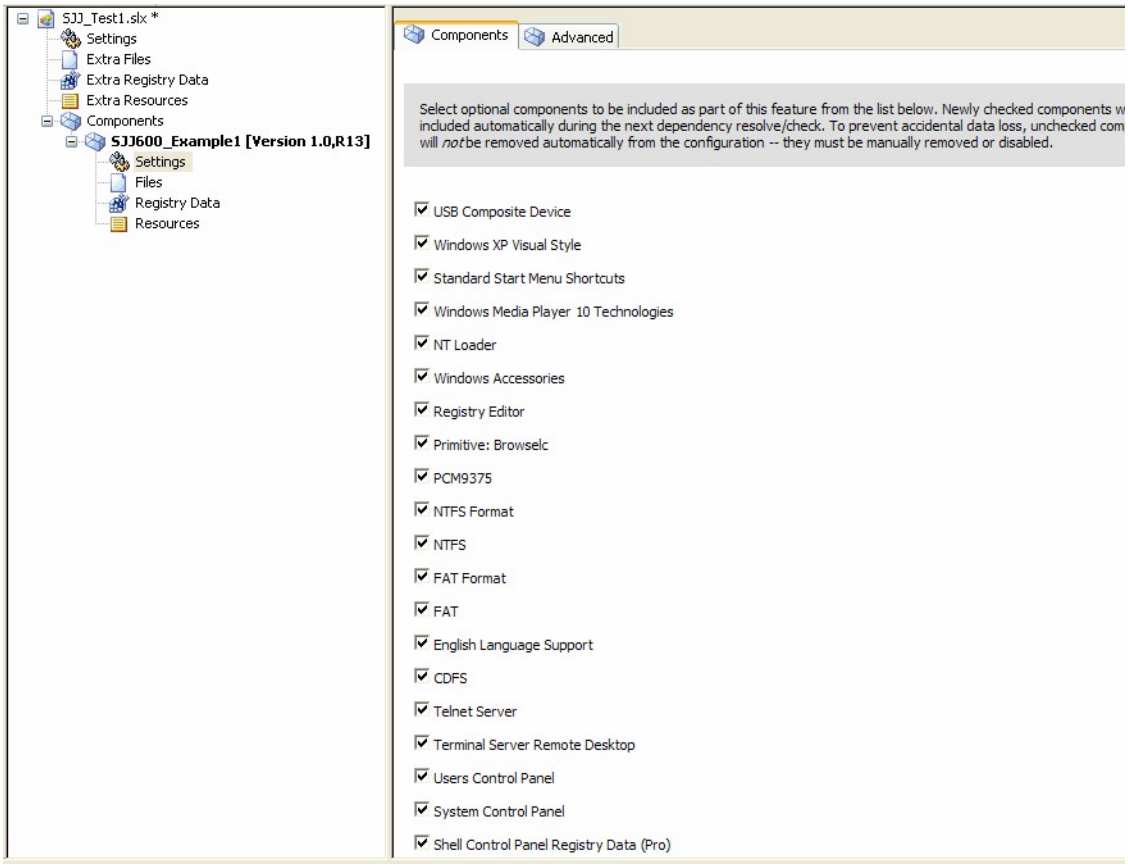
3.2 Special Category

SJJ600_Transfer, SJJ600_Example1, and SJJ600_Example2 component can be found in the component browser under the SJJ Embedded Micro Solutions Dev Kits category. The category was created by opening Component Designer with the /C option switch.



3.3 Macro Components Use Selector Prototype

The PCM9375, SJJ600_Transfer, SJJ600_Example1, and SJJ600_Example2 all have the Selector Prototype Component defined as the prototype. The Selector Prototype enables a picker list in the Target Designer settings for the component dependencies associated with the macro component.



The picker list allows you enable or disable a feature from being pulled into the configuration during a Dependency Check with Auto Resolve for the macro component. It doesn't mean that the feature will not be pulled in from some other component. You can use a tool like Component Hunter to track down dependency relationships.

4 Creating the SJJ600_Transfer Image

In order to transfer images from the desktop development system to the CF drive in the unit, a USB flash disk with a bootable XPe image will be used as a transfer mechanism. Windows XPe SP2 Feature Pack 2007 now adds the support to boot XPe from USB 2.0 flash disks.

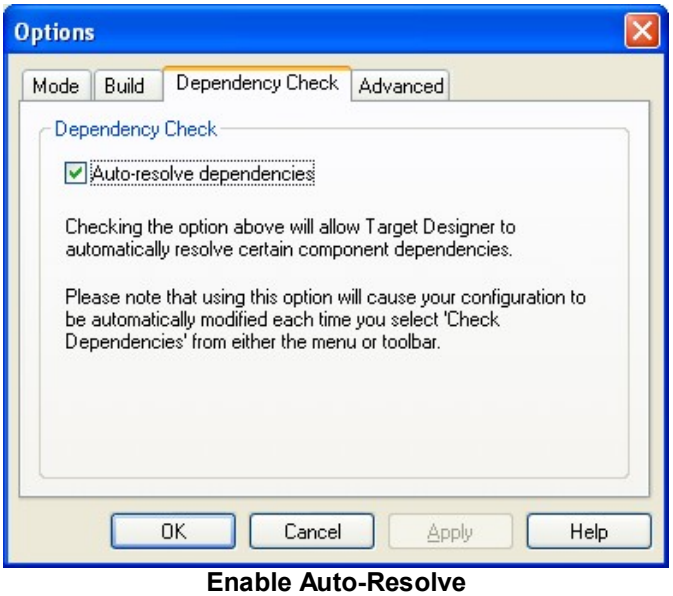
If you do not have FP2007, an alternative method must be used to transfer the image:

- Remove the CF from the target and placed in the development machine using a USB for CF adapter.
- Winternal's Remote Recover - boot the Remote Recover boot floppy and connect over the Ethernet. Remote Recover is now owned by Microsoft.

4.1 Create a new configuration

If you have FP2007, you can build a USB image as a transfer solution. Please note that this image is only for booting on the PCM9375.

1. **Open** Target Designer
2. From the menu, select **File->New**
3. The New configuration dialog appears. **Type** in the new configuration name: ***XPETransfer IMAGE***.
4. Add the **SJJ600_Transfer** macro component to the configuration.
5. Make sure the Auto-Resolve Dependencies option in the Tools->Options->Dependency Check is selected.



Enable Auto-Resolve

6. Run the **Check Dependencies** on the configuration. The Auto-Resolve will pull in all the dependent components into the configuration. The dependency check should complete without error and further component issues to resolve.
7. **Save** the configuration to the hard drive.
8. Build the transfer solution. On the Configuration menu, click Build Target Image.
9. In the Build type box, click Release.
10. In the Destination box, type the full path where you want to store the run-time image. If an image already exists in the same folder, it is overwritten when you build the new image.
11. In the Log file box, type the full path and file name for the log file.
12. Click Build.

4.2 Prep and Boot from USB 2.0 Flash Disk

1. Insert the USB flash disk in to the development PC.
2. **Open** a command window
3. Run the **UFDprep.exe** utility 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

4. Once the format has completed, **copy** the XPe image to the USB flash disk.
5. **Undock** the USB flash disk.
6. Insert the USB flash disk into the PCM9375.
7. Set the BIOS in the target to boot USB 2.0 (or USB HDD), and let the system boot from the flash disk.

The system will run through FBA and the XPe build will boot. Once completed, you can use Disk Management to format the CF drive, and use File Explorer to share the drive on the network.

5 SJJ600_Example1

The SJJ600_Example1 macro component is an example of an image with an Explorer desktop and remote connectivity. The macro component contains all the dependencies to boot an image on the PCM9375.

5.1 Create a new configuration

Once the SLD files have been imported, the configuration can be created.

1. **Open** Target Designer
2. From the menu, select **File->New**
3. The New configuration dialog appears. **Type** in the new configuration name: **Example1**,
4. Add the software component **SJJ600_Example1** macro component to the configuration.
5. Make sure the Auto-Resolve Dependencies option in the Tools->Options->Dependency Check is selected.

Enable Auto-Resolve

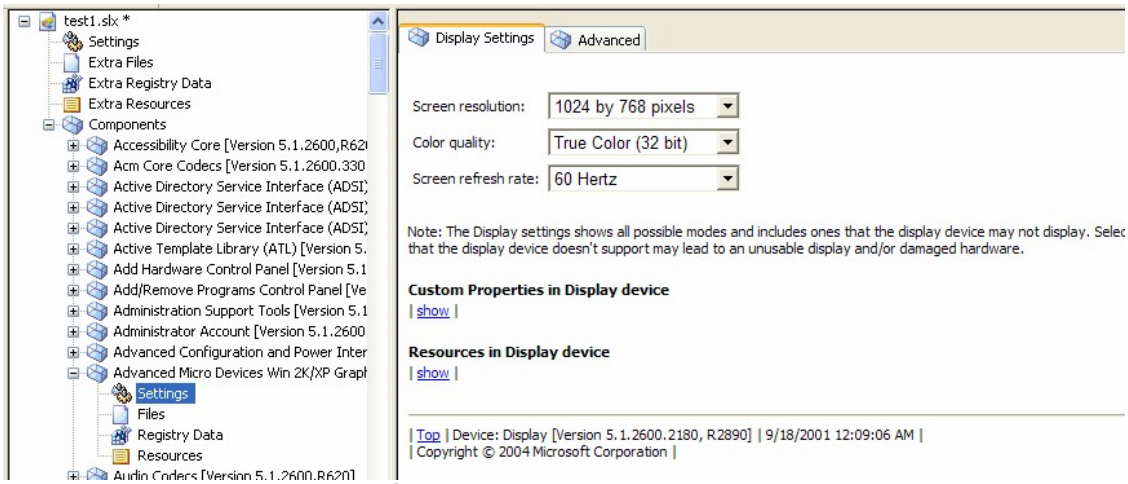
6. Run the **Check Dependencies** on the configuration. The Auto-Resolve will pull in all the dependent components into the configuration. The dependency check should complete without error and further component issues to resolve.
7. **Save** the configuration to the hard drive.

5.2 Configuration Settings

The Dependency Check with auto-resolve enabled will pull in all the components required for the target image. The next step is to manually configure the image by modifying the settings for various components. The following settings changes can be found either under a settings page or by clicking on the “Advanced” button found on the component information page.

A. Advanced Micro Devices Win 2K/XP Graphics Driver.

   Change the screen resolution: Screen Resolution: **1024x768 pixels**.

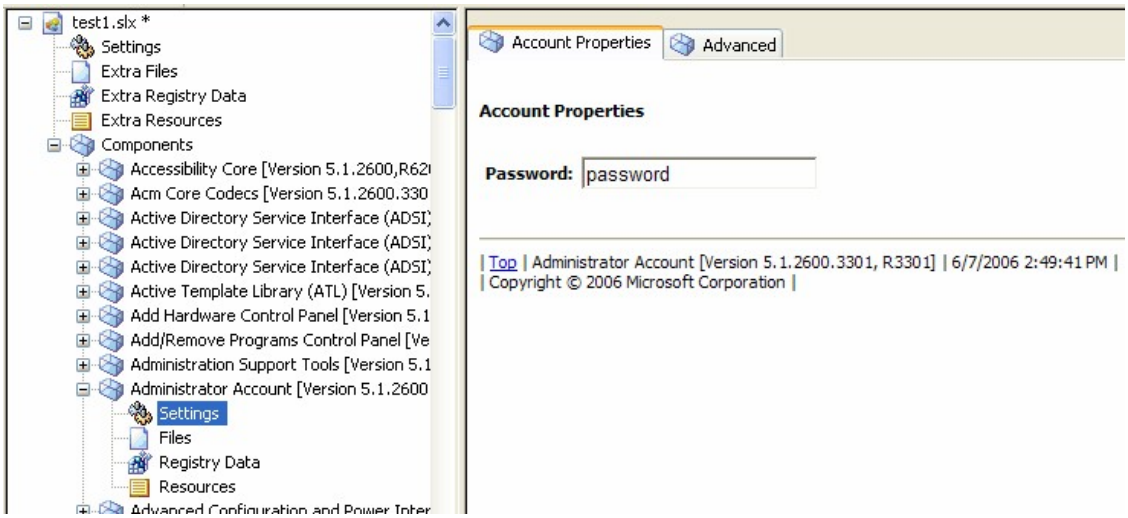


Setting the Video Resolution

B. Administrator Account

A password is need if you are going to connect to the shared drive.

Password: **password**



Setting the password

C. User Interface Core

If you are NOT using FP2007 than change the User Interface Core component settings to enable items on the start bar:

a. Show Desktop icons: **Checked**
b. Show Control Panel on Start Menu: **Checked**
c. Show Network Connections on Start Menu: **Checked**
d. Show Run on Start Menu: **Checked**
e. Show Log Off on Start Menu: **Checked**
f. Show Shut Down on Start Menu: **Checked**
g. Show All Programs list on Start menu: **Checked**
h. Show context menu on Shell folders: **Checked**
i. Show context menu on Task bar: **Checked**
j. Lock Task bar: ***Unchecked***

k. Use Windows Classic folders: **Checked**
l. Enable Drag and Drop on Start Menu: **Checked**

5.3 Re-Run Dependency Check

1. Re-run the **Dependency Check**.
2. **Save** the Configuration

5.4 Build and Deploy the Image

We are now ready to build and deploy the image once the configuration has completed the dependency check.

5.4.1 Build the OS Image

1. **Open** the configuration if it is not open already.
2. Either from the menu or the Tool Bar, select **Build the Target Image….** The Build dialog will appear.
3. Make sure the **Destination** directory is set to **c:\Windows Embedded Images**. By default a log file will be created.
4. **Click** the Build button.
5. A warning box will appear asking if you are sure about deleting the contents of the destination directory, clicking yes.

   ***Note**: You might be asked to re-run the Dependency Check, click **yes**, and make sure there are no errors or tasks once the dependency check completed.*

The build will take a few minutes. Once complete there should be no errors or warnings.

5.4.2 Deploying the OS and Running FBA

Now, we will use the USB flash drive to transfer the image.

1. Boot the PCM9375 using the USB flash containing the XPETransfer Image.
2. Go to Control Panel->Administrative Tools-> Computer Management
3. Go to Disk Management and partition and format the CF disk.
4. There are two options to transfer the image:
   a. Share the drive on the network, connect to the share via the development machine, and drag-n-drop the image to the share.
   b. Copy the image to the USB flash disk prior to booting and copy the image over to the flash disk.
5. Once the image has been deployed to the target CF drive, shutdown the transfer image. Leave USB flash disk in the unit until shutdown is complete.
6. Remove the USB flash disk.
7. Reboot the unit to the CF drive so FBA can complete the boot process.

***Note**: When booting from the USB flash disk, the flash disk must remained connected to the system until shutdown. Otherwise, the OS will blue screen.*

5.5 Remote Connection

Finally test the connection to the remote target via Ethernet.

1. On the target, open a command window.
2. Run ipconfig /all to get the TCP/IP address.

3. On the development machine open Remote desktop connection and use the TCP/IP address to connect to the remote system.
4. Once connected, the target will log-off the local session because only one log-on session is permitted.

If you want to connect via telnet, you must first either disable the firewall or add the telnet sever to the Exceptions list.

6 SJJ600_Example2

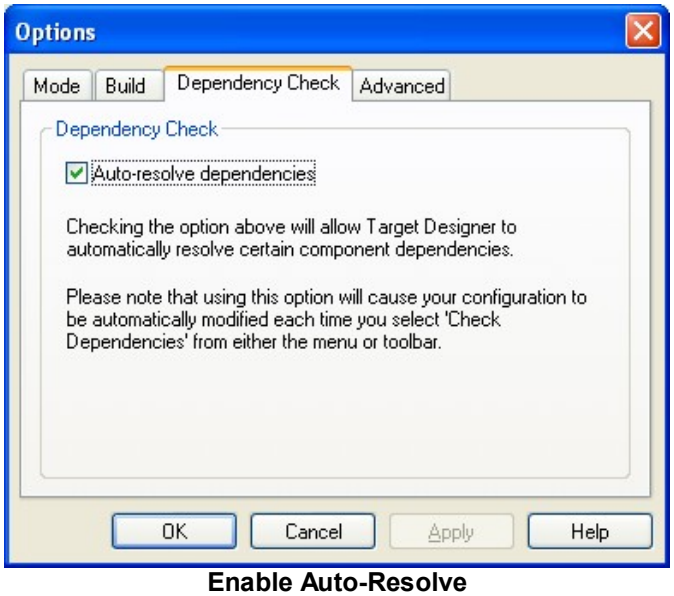
The SJJ600_Example2 macro component is an example of how to build a small OS image. The components for audio and Ethernet have been removed. The resulting image size is about 130MB. This is idea for those just wanting to use the serial ports for communication.

Flash technology continues to improve, but constant write cycles still have a possibility of shortening the life of the flash. Some CF cards have internal wear-leveling some do not. When booting from flash, you should consider using EWF or FBWF to protect the life of a flash disk. For those using FP2007, we will also add the new FBWF capability. FBWF adds islands of safety so files can be written directly to the disk instead of the overlay.

6.1 Create a new configuration

Once the SLD files have been imported, the configuration can be created.

1. **Open** Target Designer
2. From the menu, select **File->New**
3. The New configuration dialog appears. **Type** in the new configuration name: **Example2**,
4. Add the software component **SJJ600_Example2** macro component to the configuration.
5. If you are using FP2007, add the **File Based Write Filter** component.
6. Make sure the Auto-Resolve Dependencies option in the Tools->Options->Dependency Check is selected.



Enable Auto-Resolve

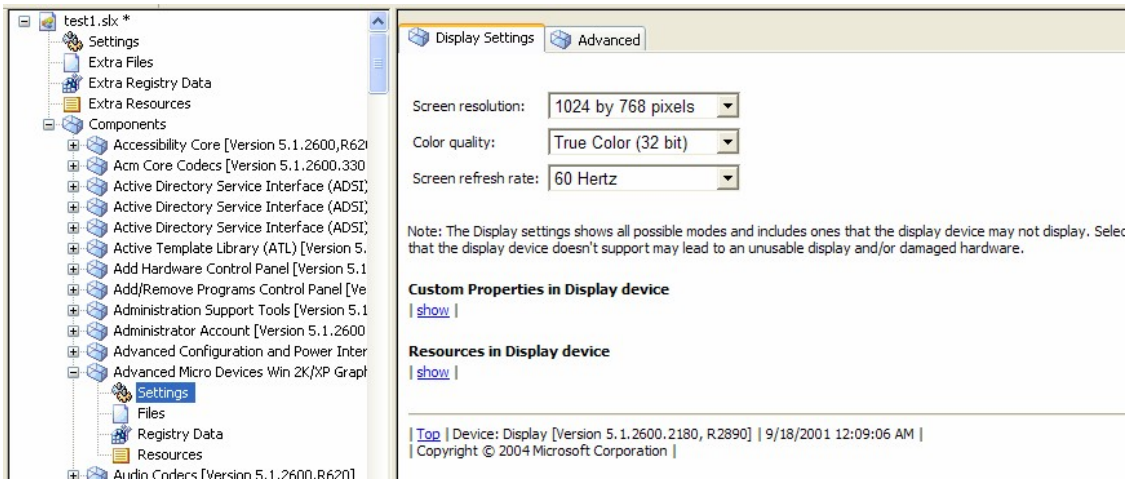
7. Run the **Check Dependencies** on the configuration. The Auto-Resolve will pull in all the dependent components into the configuration. The dependency check should complete without error and further component issues to resolve.
8. **Save** the configuration to the hard drive.

6.2 Configuration Settings

The Dependency Check with auto-resolve enabled will pull in all the components required for the target image. The next step is to manually configure the image by modifying the settings for various components. The following settings changes can be found either under a settings page or by clicking on the “Advanced” button found on the component information page.

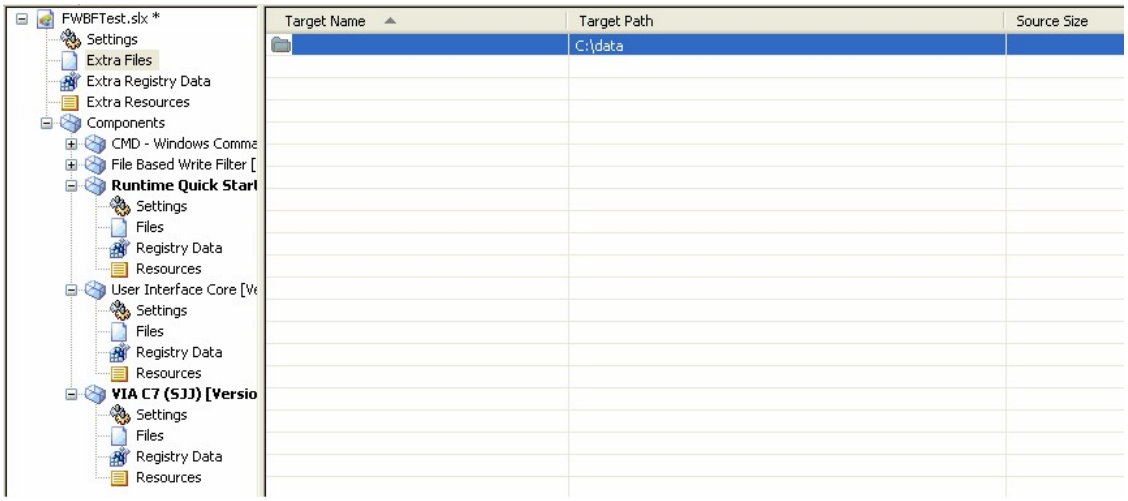
A. Advanced Micro Devices Win 2K/XP Graphics Driver.

Change the screen resolution: Screen Resolution: **1024x768 pixels**.



B. Add a Folder

Under the Extra Files, add a new folder called Data. The Folder will be used as our write-through test.

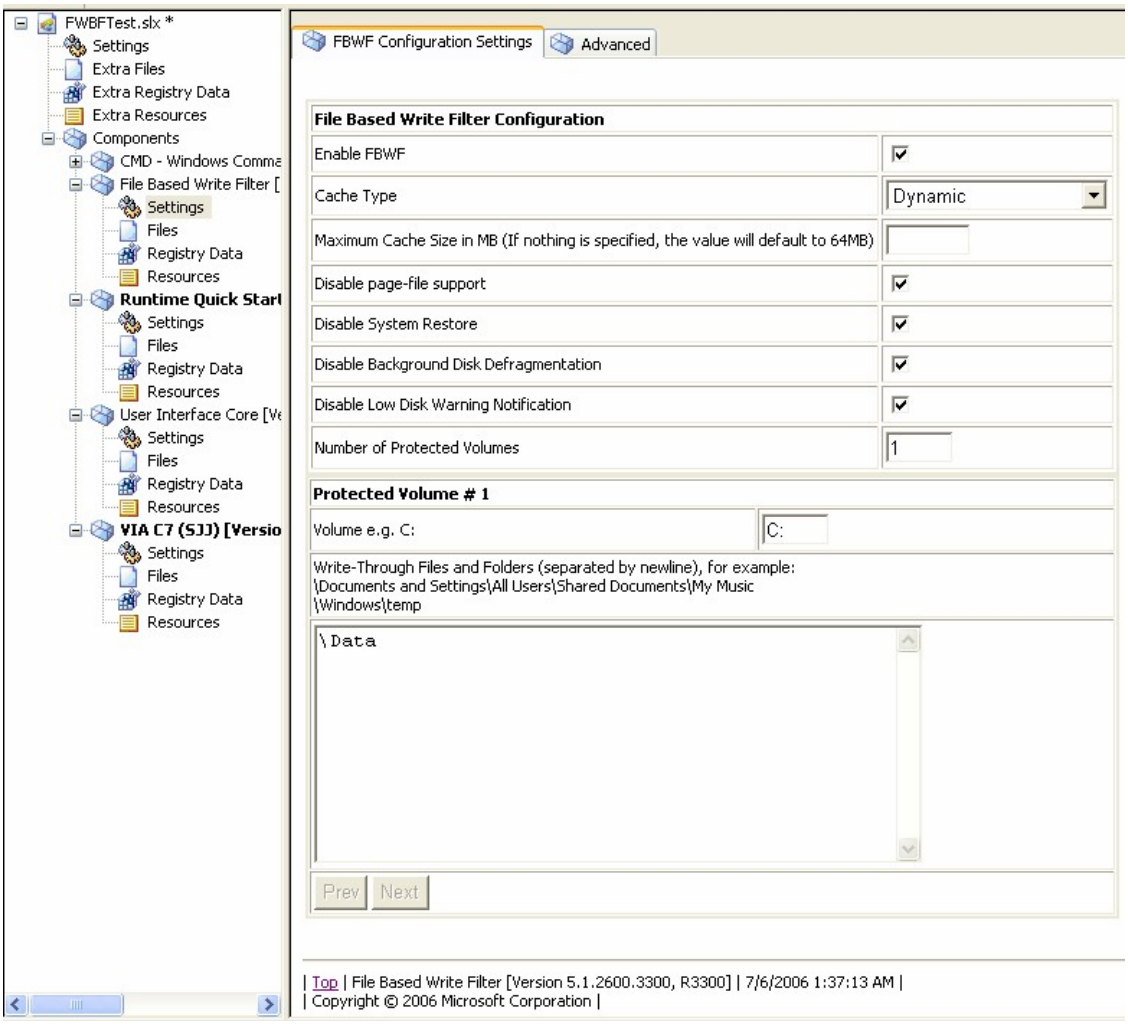


Adding a Directory

Now that we have a folder to write to, let's modify the File Based Write Filter settings. Like EWF, FBWF's Settings page is split into an upper half (File Based Write Filter Configuration) and a lower half (settings for each Protected Volume).

C. File Based Write Filter

Keep the File Based Write Filter Configuration settings as the defaults. We will only be protecting the C drive so add "C:" to the Protect Volume #1. Add the \Data directory to the Write-Through Files and Folders list. This means all other files and folders under C:, including the OS files, will be protected from writes.



FBWF Settings

6.3 Re-Run Dependency Check

3. Re-run the **Dependency Check**.
4. **Save** the Configuration

6.4 Build and Deploy the Image

We are now ready to build and deploy the image once the configuration has completed the dependency check.

6.4.1 Build the OS Image

6. **Open** the configuration if it is not open already.
7. Either from the menu or the Tool Bar, select **Build the Target Image….** The Build dialog will appear.
8. Make sure the **Destination** directory is set to **c:\Windows Embedded Images**. By default a log file will be created.
9. **Click** the Build button.
10. A warning box will appear asking if you are sure about deleting the contents of the destination directory, clicking yes.

***Note**: You might be asked to re-run the Dependency Check, click **yes**, and make sure there are no errors or tasks once the dependency check completed.*

The build will take a few minutes. Once complete there should be no errors or warnings.

6.4.2 Deploy the OS and Running FBA

Now, we will use the USB flash drive to transfer the image.

1. Boot the PCM9375 using the USB flash containing the XPETransfer Image.
2. Go to Control Panel->Administrative Tools-> Computer Management
3. Go to Disk Management and partition and format the CF disk.
4. There are two options to transfer the image:
    a. Share the drive on the network, connect to the share via the development machine, and drag-n-drop the image to the share.
    b. Copy the image to the USB flash disk prior to booting and copy the image over to the flash disk.
5. Once the image has been deployed to the target CF drive, shutdown the transfer image. Leave USB flash disk in the unit until shutdown is complete.
6. Remove the USB flash disk.
7. Reboot the unit to the CF drive so FBA can complete the boot process.

***Note**: When booting from the USB flash disk, the flash disk must remained connected to the system until shutdown. Otherwise, the OS will blue screen.*

6.5 Testing FBWF

Let's test to see if FWBF really works. We will use the FBWFMGR utility that comes with the FBWF component to control the state of FBWF.

1. The system should be booted to the command shell.
2. Type the following to see the status of FBWF:

```
C:\>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
```

3. Create a new folder called test1 under c:\data and another folder called test2 under c:\.

4. Reboot the system. After the system reboots. Test1 folder should still exist. Test2 should be gone.

5. Now let's use the FBWFMGR utility. Open a Command Window.
6. Enter the following at the command line to disable FBWF:

   C:\>**Fbwfmgr /disable**

7. Reboot the system.
8. Create a new folder called test2 under c:\
9. Reboot the system. Test2 should still be there.

7 Technical Support

For any technical questions about the IDK, please connect support:

E-mail: support@sjjmicro.com
Tel: 858-485-1059