DA-682 Series
WinXP Embedded User's Manual

Third Edition, January 2010

www.moxa.com/product

DA-682 Series
WinXP Embedded User's Manual

Copyright Notice

Trademarks

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use. This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information
www.moxa.com/support

Moxa Americas:
Toll-free: 1-888-669-2872
Tel: +1-714-528-6777
Fax: +1-714-528-6778

Moxa China (Shanghai office):
Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-10-6872-3958

Moxa Europe:
Tel: +49-89-3 70 03 99-0
Fax: +49-89-3 70 03 99-99

Moxa Asia-Pacific:
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

Table of Contents

1

Introduction

Thank you for purchasing a Moxa DA-682 Series x86 ready-to-run embedded computer. This manual introduces the software configuration and management for the Windows XP Embedded operating system. For hardware installation, connector interfaces, setup, and upgrading the BIOS, please refer to the DA-682 Series Hardware User's Manual.

Microsoft Windows XP Embedded is a specialized operating system with Windows XP Professional features that allows you to build a wide range of innovative, small footprint devices. Windows XP Embedded has the same binary files as Windows XP Professional. Applications that run under Windows XP can also run under Windows XP Embedded, and there is no additional cost required to migrate from XP to XPE. Windows XP Embedded enables you to develop reliable and full-featured connected devices, quickly and without spending an inordinate amount of effort.

The following topics are covered in this chapter:

- **Overview**
- **Product Features**
- **Software Specifications**
  - Application Development Environment
  - Networking and Communication Capabilities
  - Supporting Services and Daemons

Overview

The Moxa DA-682 Series of X86-based rackmount embedded computers are designed for industrial data acquisition applications. The state-of-art 2 expansion module design provides greater flexibility by allowing combinations of up to 16 RS-232/422/485 serial ports or up to 4+8 Ethernet ports. This friendly design gives users the advantage of being able to swap out modules quickly and easily. Additional expansion modules will be available soon to provide even greater flexibility.

The DA-682's main operating system is based on the Intel Celeron M processor and 915GME chipset, which supports standard X86 VGA, USB, PS/2 keyboard/mouse, 4 Gigabit LAN ports, and IDE/SATA disk interface. In addition, the DA-682 has a CompactFlash socket and comes with Linux, WinCE 6.0, Windows XP Embedded pre-installed. Programmers will find the full-function development kit a great benefit for developing software and building reliable communication applications.

The housing is a standard 2U, 19-inch wide rack-mounted rugged enclosure. This robust, rack-mountable design provides the hardened protection needed for industrial environment applications.

Product Features

The DA-682 Basic System has the following features:

- Intel Celeron M 1GHz processor with 400/533 MHz FSB
- Intel 915GME + ICH6M chipset
- 200-pin DDR2 SODIMM socket x1, supporting DDR2 400/533 up to 1 GB
- 4 Gigabit Ethernet ports for network redundancy
- 1 CompactFlash socket
- 1 SATA connector for storage expansion
- USB 2.0 ports for high speed peripherals
- 2 expansion module slots for I/O expansion
- 19-inch rackmount, 2U height form factor
- Fanless design
- 100/240 VAC/VDC power inputs
- Ready-to-run Linux, WinCE 6.0, or Windows XP Embedded platform pre-installed on the flash disk module

Special features for the DA-682-XPE Window XP Embedded model:

- Shipped with DDR2 512 MB memory
- Ready-to-Run Windows XP Embedded platform pre-installed on 1 GB flash disk module

Features supported by expansion modules:

- 8 or 16 software selectable RS-232/422/485 serial ports, with or without isolation protection
- Serial port baudrates from 50 to 921.6 Kbps, with support for most non-standard baudrates in this range
- Additional 4 or 8 10/100 Mbps Ethernet ports

ATTENTION

Refer to the "Non-standard Baudrates" section for instructions on how to calculate which baudrates are supported.

Software Specifications

The DA-682-XPE embedded computer provides the following common, popular application development features, networking/communications capabilities, and supporting services/daemons, making the Windows XP Embedded with SP2 environment an easy and convenient programming tool. The software features of the DA-682-XPE embedded computer are listed below:

Application Development Environment

- **Microsoft .Net Framework 2.0 with service pack 2**—Includes the common language runtime (CLR) and the .NET Framework class library.
- **Active Directory Service Interface (ADSI) Core**—Provides the basic functionality for ADSI by routing requests to the corresponding provider, based on the path that is provided.
- **Active Template Library (ATL)**—Supports ATL applications.
- **ASP.NET 2.0**—A unified Web application platform that provides the services necessary to build and deploy enterprise-class web applications.
- **Certificate Request Client & Certificate Autoenrollment**—This component includes the common language runtime (CLR) and the .NET Framework class library.
- **COM Base**—The Component Object Model (COM) includes a programming model and a set of application programming interfaces (APIs), but does not include a dedicated user interface.
- **Common Control Libraries**—Provides common user interface (UI) controls.
- **Common File Dialogs**—Provides support for common dialog boxes.
- **Direct3D**—Infrastructure for using two-dimensional and three-dimensional graphics.
- **DirectPlay**—Provides a networking API that can enable any application to operate over both a peer-to-peer and client/server topology.
- **DirectShow**—Base filter graph and device enumeration support for all DirectShow applications. This component also provides most DirectShow filters.
- **Distributed Transaction Coordinator (MSDTC)**—A distributed transaction facility for Microsoft Windows systems, which uses transaction-processing technology. MSDTC uses loosely coupled systems to provide scalable performance.
- **Enhanced Write Filter**—An upper filter in the storage device driver stack that redirects disk write operations to DRAM.
- **Event Log**—A dynamic-link library (DLL) that runs as part of Services.exe. This component stores and retrieves events that can be viewed in the event viewer.
- **Internet Explorer**—The Internet Explorer Web browser allows customers to connect to the Internet or to an intranet (see properties via inetcpl.cpl).
- **Mapi32 Libraries**—The infrastructure for e-mail support.
- **Message Queuing (MSMQ) Core**—Message Queuing is a messaging infrastructure and a development tool for creating distributed messaging applications for Microsoft Windows operating systems. It provides guaranteed message delivery, efficient routing, increased security, support for sending messages within transactions, and priority-based messaging.
- **Microsoft Visual C++ Run Time Libraries**—The Microsoft C++ Runtime Library.

- **NTFS**—The NTFS File System driver (NT File System). Use NTFS instead of FAT for optimum file system security.
- **Registry Editor**—The Registry Editor (regedit.exe, regedt32.exe).
- **RPC**—Facilitates local remote procedure calls (RPCs) using the ncalrpc and ncacn_np protocol sequences, and provides support for dynamic endpoint resolution. The RPC name service provides remote procedure call (RPC) named services functionality, such as the RPC Locator. The RPC Named Service component exposes all RpcNs* RPC functions. The RPC server provides a variety of RPC and Component Object Model (COM) services, including RPC Endpoint Mapper, COM Service Control Manager (SCM), and COM Object Resolver.
- **Smart Card Cryptographic Service Providers**—Supports features such as smart card logon and improved email security. Smart cards must be able to perform certain RSA public key cryptographic operations. The functions are exposed with CryptoAPI through a CSP. Each type of smart card requires a different CSP (provided by the card vendor).
- **USB 2.0**—The core drivers needed to communicate with an Enhanced Host Controller Interface (EHCI) that is compliant with USB .95 or 1.0.
- **Windows API**—Provides the user-mode component of the Windows operating system API.
- **Windows Media Player 10**—Playback functionality for digital media that includes music, videos, CDs, DVDs, and Internet Radio for end users and developers.
- **Windows Script Engines**—A complete scripting environment for Windows, including command-line scripting, script languages, and the ability to host script engines within your applications.
- **WMI**—Bundles the features that combine to create Windows Management Instrumentation (WMI).

Networking and Communication Capabilities

- **DHCP Client Service**—Registers and updates Internet Protocol (IP) addresses and Domain Name System (DNS) records for your target system.
- **IP Security Services**—This component provides IP Security (IPsec) services for all IP traffic.
- **Dial-Up Networking**—Provides the infrastructure necessary to implement a Remote Access Service (RAS) client.
- **Microsoft-Windows-HTTP**—Services that implement the functionality of the HTTP protocol on a server.
- **TCP/IP Networking**—Implements the core TCP/IP protocol stack, which includes the IPv4 version of the following protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), raw, Internet Control Message Protocol (ICMP), Internet Group Membership Protocol (IGMP), and Address Resolution Protocol (ARP). The component also includes Wshtcpip.dll, which is the Winsock provider for TCP/IP to enable socket-level communication over TCP/IP.
- **TAPI**—A Telephony API (TAPI) Telephony Service Provider (TSP).
- **Simple Network Management Protocol (SNMP)**—SNMP is an agent service that provides management systems with information about activities that occur in the Internet Protocol (IP) network layer. The SNMP agent monitors network traffic, and retrieves and updates local management information based on the requests from the SNMP manager. The agent also notifies registered managers with traps when significant events occur.
- **Time Service Core**—Synchronizes a workstation's clock with other computers using Network Time Protocol (NTP) version 3. This component increases accuracy by incorporating algorithmic enhancements from NTP 4.

- **Windows Firewall/Internet Connection Sharing (ICS)**—Windows Firewall provides a barrier between your device and network connections to reduce attacks by hackers, viruses, and worms across networks.
- **Wireless Zero Configuration**—Supports the Windows implementation of the IEEE 802.11 standard. This component performs automatic configuration and authentication for IEEE 802.11 wireless network adaptors.
- **Unimodem**—Provides the infrastructure necessary for applications to communicate with a modem.

Supporting Services and Daemons

- **COM+ Services**—The next evolution of Microsoft Component Object Model (COM) and Microsoft Transaction Server (MTS).
- **Computer Browser Service**—Computer browsing functionality that allows a client machine to browse its network neighborhood for available computers to find file and print sharing services.
- **Disk Management Services**—Support for disk and volume management operations. The component implements a Component Object Model (COM) interface that can be used to query and configure disks and volumes (both basic and dynamic). The component also monitors disk arrivals and removals and other changes in the storage subsystem.
- **IIS Web Server**—Allows you to create and manage Web sites.
- **Terminal Server**—Microsoft Terminal Server client application (mstsc.exe).
- **Remote Registry Service**—Enables remote users to modify registry settings on this computer.
- **Telnet Server**—Allows users to connect to Telnet servers from remote computers.

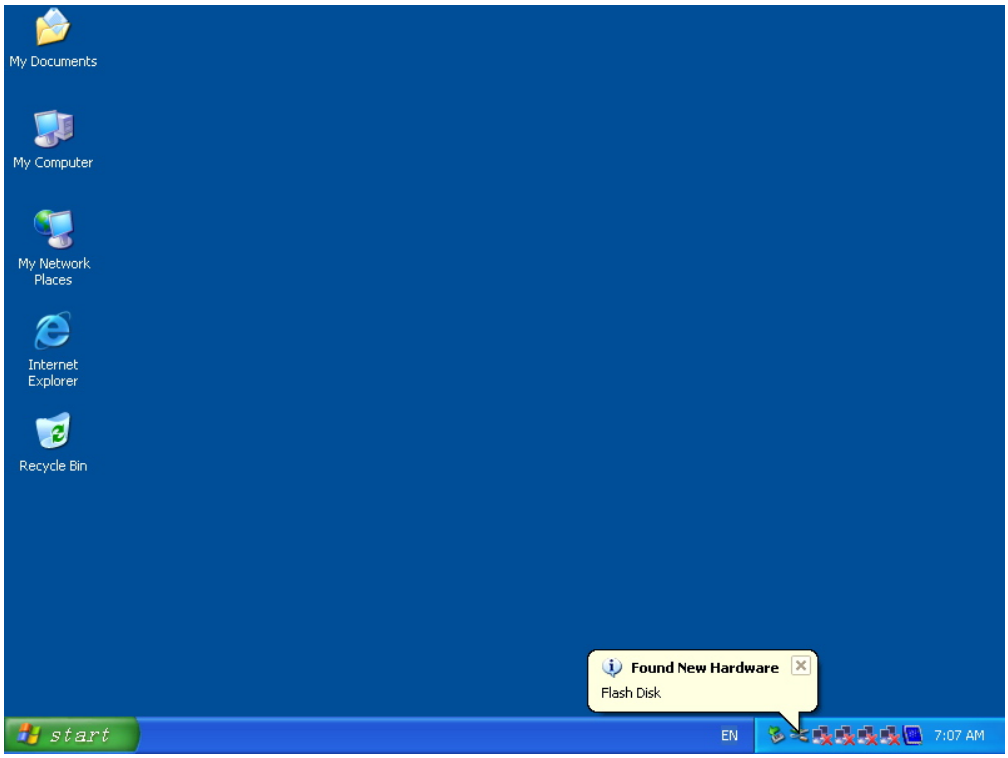
2

Software Configuration

In this chapter, we explain how to operate a DA-682-XPE computer directly or from a PC. In addition, we describe how to take care of system time adjustment, troubleshooting network connectivity, etc. Some of these operations can be done with system commands after gaining access to the computer, and others can be done with the “Control Panel,” which is described in a later chapter.

Starting Your DA-682-XPE Computer

Connect the CRT monitor or LCD monitor to the target computer, and then power it up by connecting it to the power adaptor. It takes about 30 to 40 seconds for the system to boot up. Once the system is ready, the desktop will appear on your monitor.



Resetting Your DA-682-XPE Computer

Reset Button

A **Reset** button is located on the front panel of the DA-682-XPE. You can shut down your DA-682 by pressing this button, just as you would do with a standard PC.

Software Shutdown/Reboot

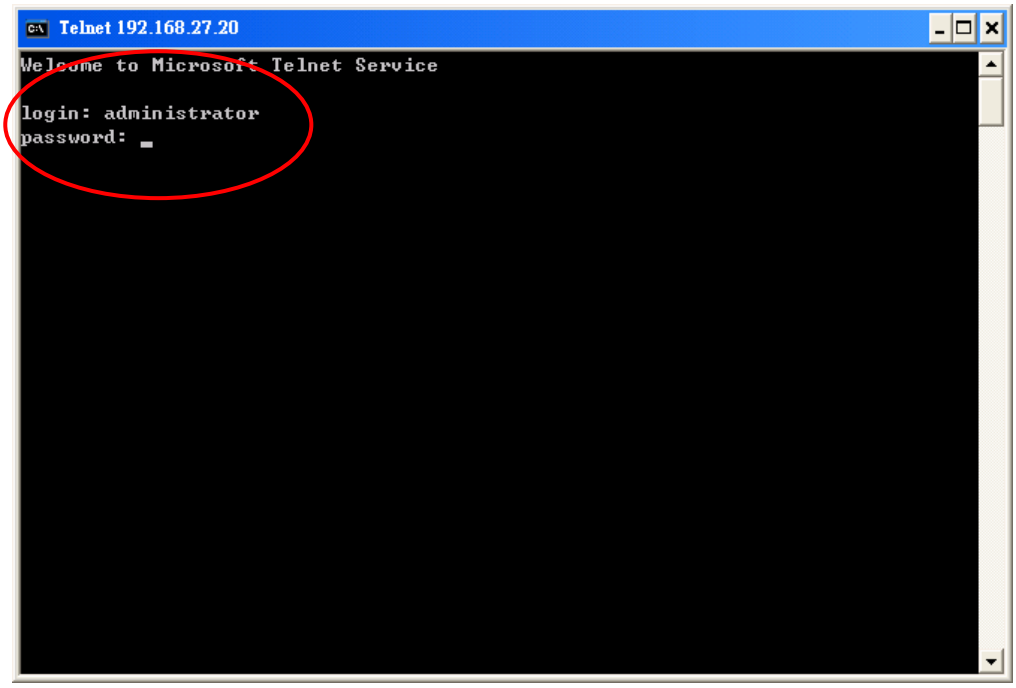
Click **Start → Shutdown** to reboot or shutdown the DA-682-XPE computer.

Operating Your DA-682-XPE Computer with a Telnet Client

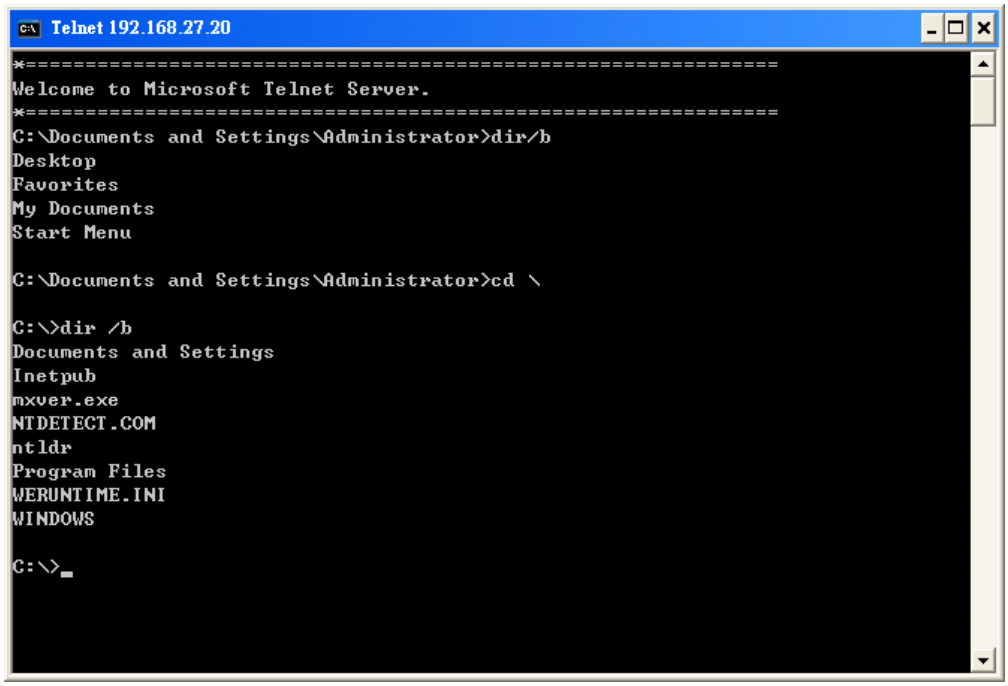
Use a crossover Ethernet cable to connect your development workstation directly to the target computer, or use a straight-through Ethernet cable to connect the computer to a LAN hub or switch. Next, use a Telnet client on your development workstation to connect to the Telnet console utility of the target computer. After a connection has been established, type the login name and password as requested to log on to the computer.

ATTENTION

The default user id is "**administrator**" and the default password is not set; you need to create a new password for this account to use the Telnet client.



After logging in through the Telnet client, a list of commands will be available for operating the computer. Use ***HELP*** to display all of the commands, or type ***HELP [command name]*** to display extended help for the selected command. Some of these commands, such as ***DATE*** and ***TIME***, are very useful for managing the computer's system time. Other commands, such as ***DIR*** and ***MKDIR***, are good utilities for file management. For example, to inspect the file structure of the root directory, type ***DIR.***



ATTENTION

You can only create two Telnet connections at the same time.

Adjusting the System Time

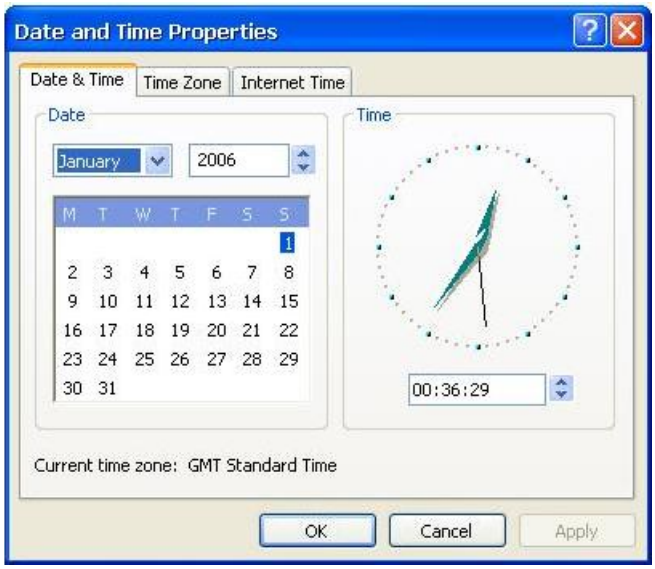
Two methods are available for adjusting the system Date/Time.

Setting the System Time Manually

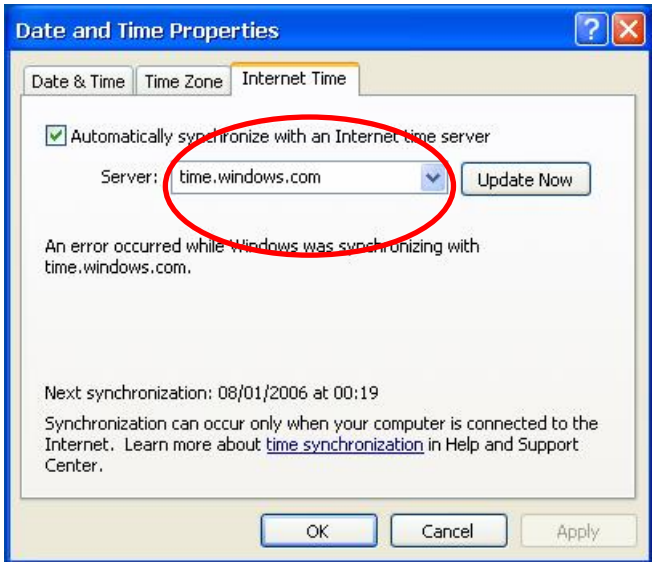
Use the **date** and **time** command line utility to query the current system date/time or set a new system date/time.

Date/Time Control Panel

Go to the Control Panel and double click the **Date/Time** icon, you can adjust the Date and Time with the **Date and Time Properties** Window.

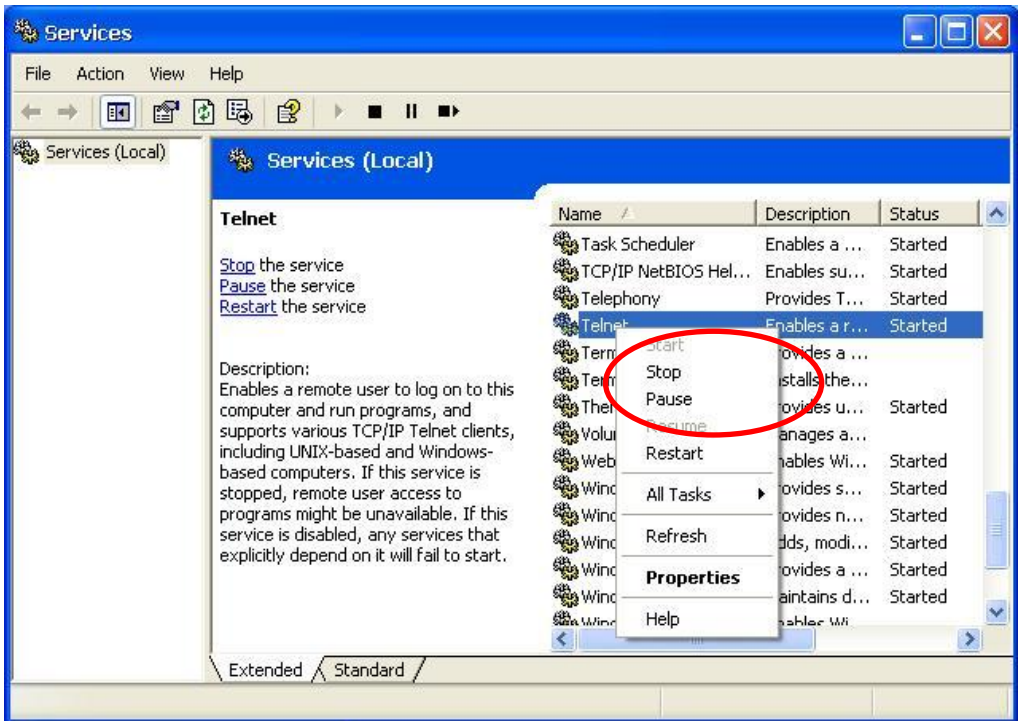


Go to **Date/Time properties → Intenet Time** to adjust the NTP server setting.



Starting and Stopping Services

To start or stop an XP service, select **Start → Control Panel → Administrative Tools** and double click **Services**. Then, use right-click to start or stop the service you want.



Simple Network Management Protocol (SNMP)

To check SNMP agent capabilities on a target DA-682-XPE computer (e.g., network IP at 192.168.3.127), log in to the workstation computer that the SNMP manager resides and type:

\> snmpwalk -v 2c -c public 192.168.3.127 system

You will see a series of messages from the SNMP agent on the DA-682-XPE computer. From here, you can monitor and manage the computer.

Serial Port Operation Mode

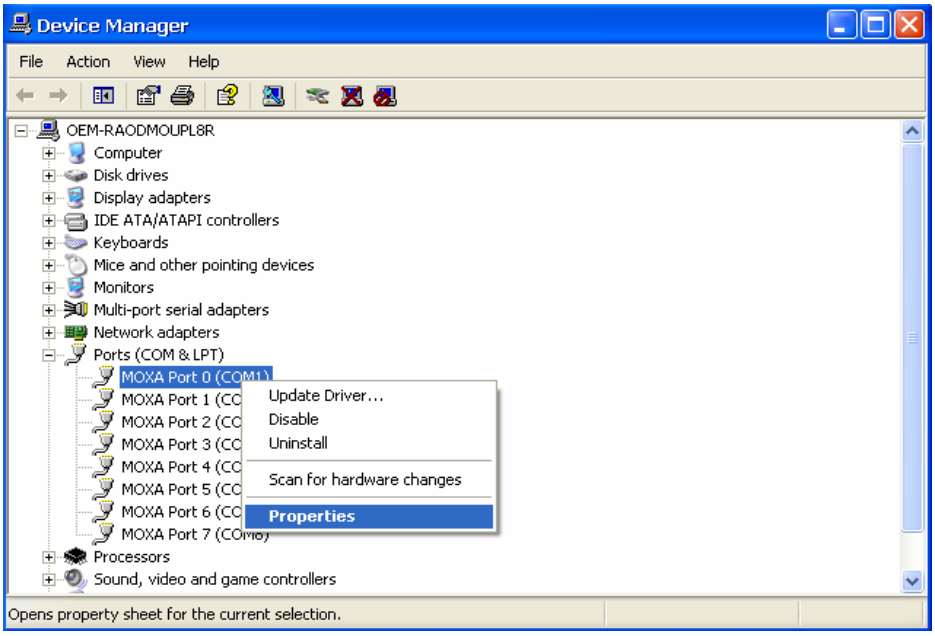
If you install the "Eight Serial Ports" module, the serial port names for the DA-682-XPE computer, going from left to right and bottom to top, use the most common COM names: COM1, COM2, …, COM8. The serial ports are designed to provide reliable, high-speed, 3-in-1 (i.e., RS-232, RS-422, and RS-485) operation modes for your diverse applications. Each of the ports support baudrate settings up to 921600 bps.

Take the following steps to set the operation mode of each COM port:

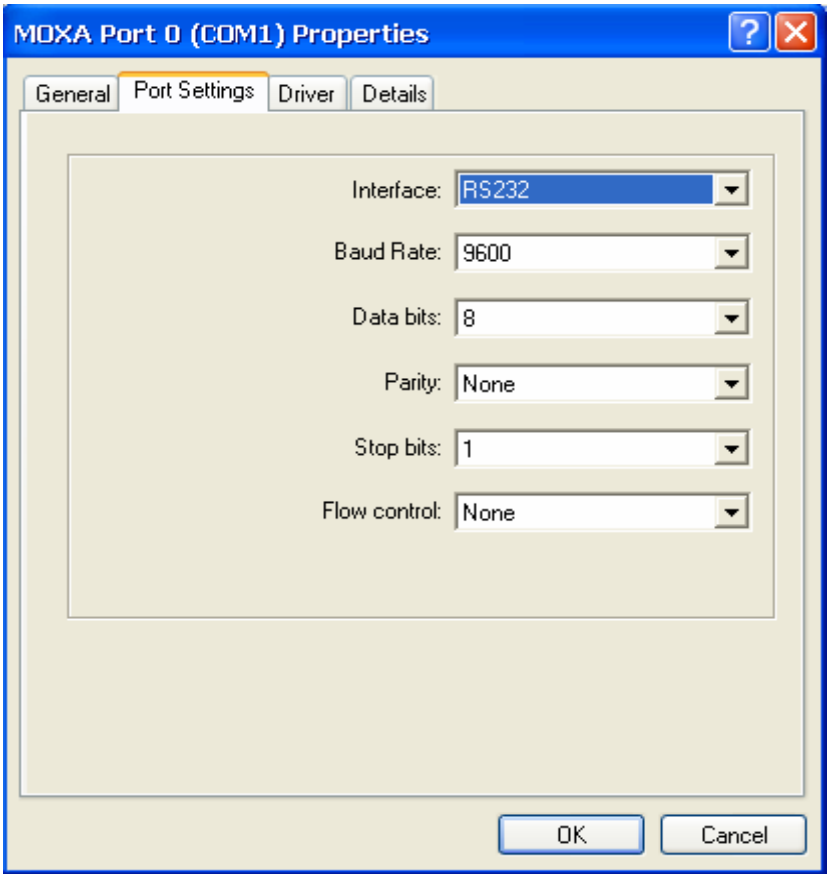
a. Go to the **Control Panel → Ports(COM & LPT)** and select the COM port (e.g., MOXA Port 0 (COM1)).

b. Right-click on the COM port and click **Properties**.

c. On the **Port Settings** tab, select which interface you want to use.

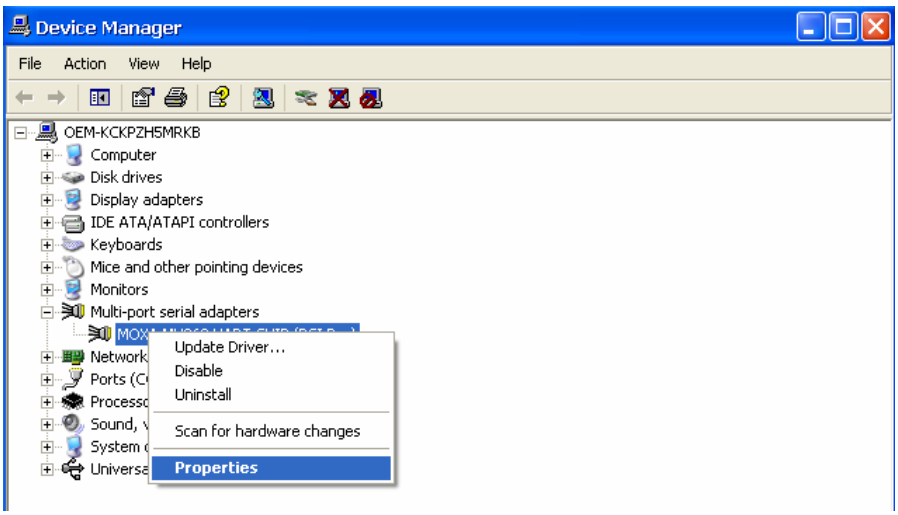


d. Click **OK** to activate the settings.

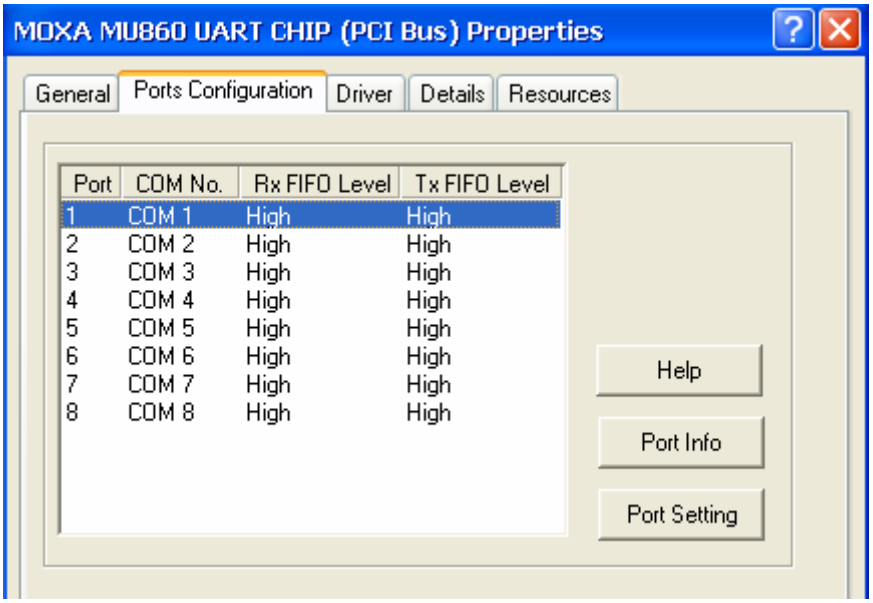


In some situations, you may want to change a port name to fit your own program. To do this, do the following:

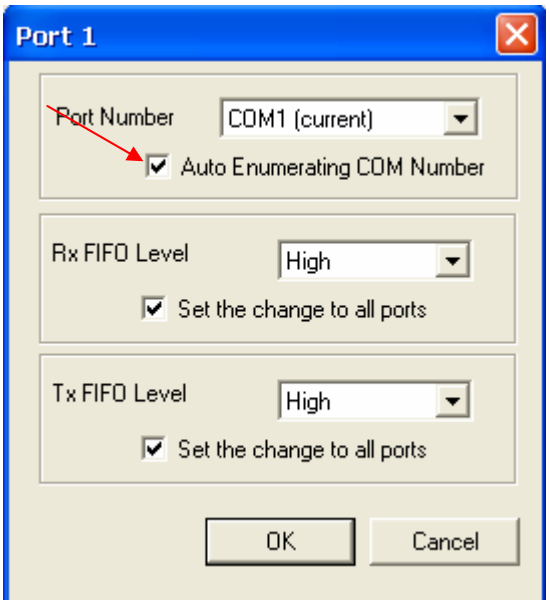
1. Go to the **Control Panel → Multi-port serial adapters** and select the adapter.
2. Right-click the adapter and click **Properties**.



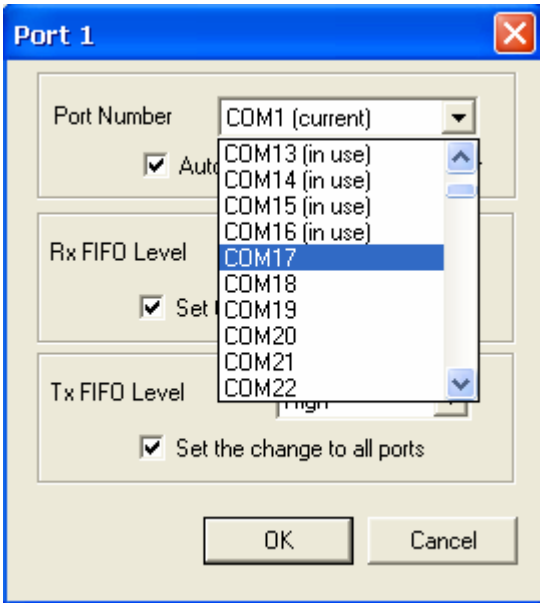
3. On the **Port Configuration** tab, select the port whose name you want to change and then click **Port Setting**.



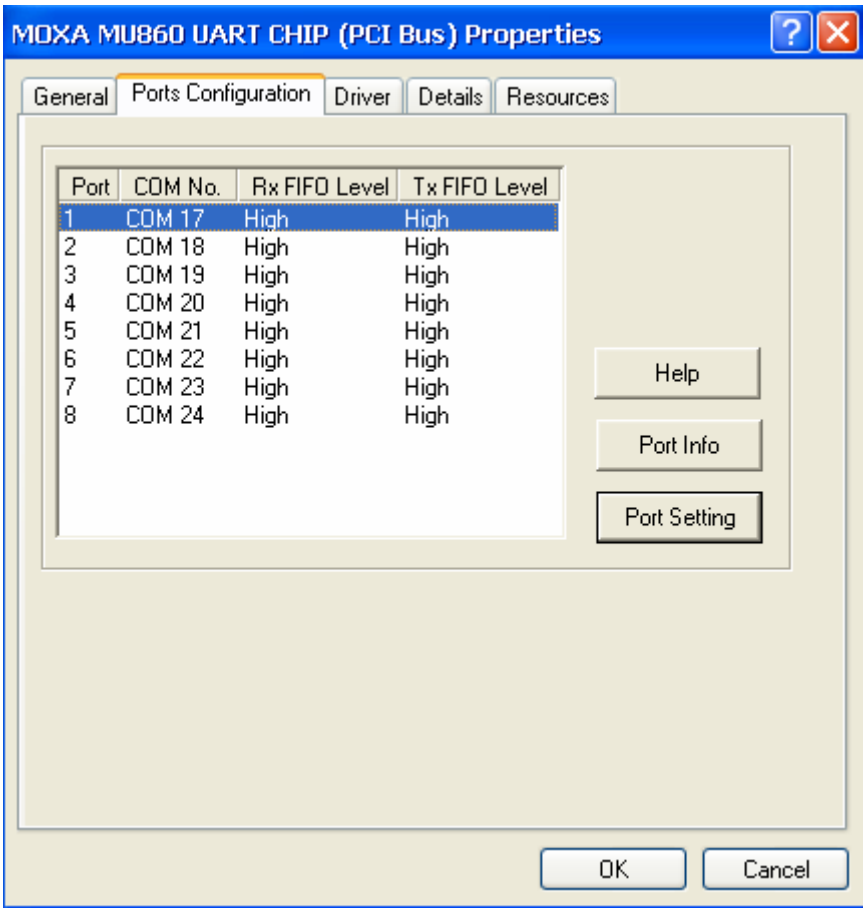
4. Uncheck "Auto Enumerating COM Number" if you want to change the port name separately.



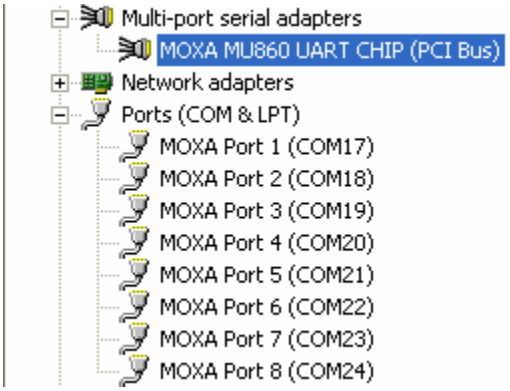
5. Select the port name you would like to use and then click **OK**.



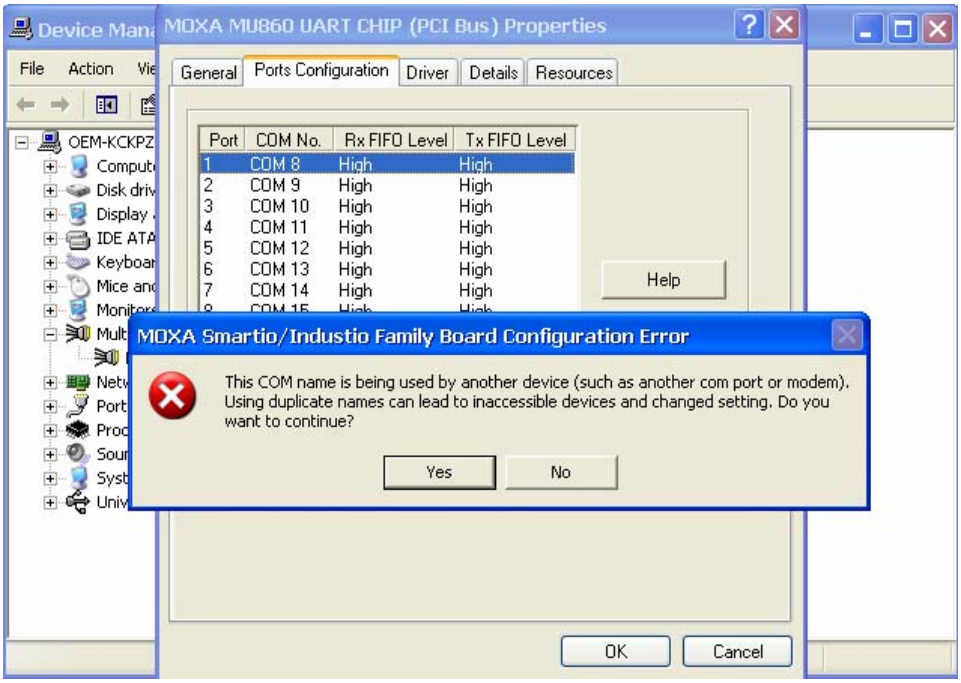
6. Make sure the port names are correct, and then click **OK** to activate the settings.



7. At this point, you should be able to see that the port names have been changed under **Ports (COM & LPT)**.



NOTE: Make sure each port name is unique, since using duplicate names will cause some devices to be inaccessible.



You can also set the operation mode by running a program. For example, the code "UartMode" is under \examples\C++\ on Software DVD.

The code snippet is as follows:

```
#include "stdafx.h"
#include <windows.h>
#include "devices.h"

#define _USE_IOCTL 1

#if _USE_IOCTL
   #include "UartMode.h"
#endif
/*
   Port[n]: 1  ; COM1
         2  ; COM2
         3  ; COM3
             ....
   Mode[n]: 0  ; RS-232
          1  ; RS-485 2W
          2  ; RS-422
          3  ; RS-485 4W
*/
```

```
int _tmain(int argc, _TCHAR* argv[])
{

   UCHAR    nPort;
   UCHAR    nMode;
   int      curMode;


   /*Port handle*/
   HANDLE   hCOM;

   /*Port information variables*/
   int      portNum=0;
   int      portStart=0;
   int      portEnd=0;

   /*Registry variables */
   HKEY hKey;
   DWORD    ulRet;
   WCHAR   sPort[20];


   if(argc==3)
   {
        nPort=0x0+(UCHAR)_wtol(argv[1]);
        nMode=0x0+(UCHAR)_wtol(argv[2]);

#if _USE_IOCTL
        portMode=nMode;
        if((nPort>=1) && (nPort<=16) && (nMode<=3))
        {
             printf("Port=COM%d, Mode=%x \n",nPort,nMode);

             wsprintf(sPort,L"\\\\.\\COM%d",nPort);
hCOM=CreateFile(sPort,  GENERIC_READ | GENERIC_WRITE,0,NULL,OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL,NULL);

             if(hCOM != INVALID_HANDLE_VALUE) //open device success
             {
```

```
                SetPortVal(hCOM);
            }
            GetPortVal(hCOM);
            CloseHandle(hCOM);


        }
        else
        {
            printf("Invalid parameter! Please try again.\n");
            Usage();
            exit(0);
        }


        /*Set Registry Value*/
      ulRet = RegOpenKeyEx( HKEY_LOCAL_MACHINE, TEXT("SOFTWARE\\MOXA\\COM"),
NULL, KEY_ALL_ACCESS, &hKey);
        if ( ulRet == ERROR_SUCCESS )
        {
            wsprintf( sPort, L"COM%d", nPort);
ulRet = RegSetValueEx( hKey, sPort, NULL, REG_DWORD, (LPBYTE)&portMode,
sizeof(DWORD) );
            if ( ulRet != ERROR_SUCCESS )
            {
                printf("Set registry value error!\n");
            }
        }
#else
        /* Set Mode Configuration */
        if(setUrMode(nPort,nMode))
        {
            printf("set OK\n");
        }

        /* Get Mode Configuration */
        curMode=getUrMode(nPort);
        if(curMode==(-1))
        {
            printf("error");
        }
```

```
        else
        {
            printf("get OK\n");
            printf("port:%d, get current mode %d\n",nPort,curMode);
        }


#endif
   }
   else
   {
        printf("Invalid parameter! Please try again.\n");
        Usage();
   }


   return 0;
}



#if _USE_IOCTL

BOOL SetPortVal(HANDLE hndFile)
{
   IoctlCode = IOCTL_SET_MODE;
   IoctlResult = DeviceIoControl(
            hndFile,             // Handle to device
            IoctlCode,            // IO Control code for Write
            &portMode,                // Buffer to driver.  Holds port & data.
            sizeof(portMode),      // Length of buffer in bytes.
            NULL,               // Buffer from driver.   Not used.
            0,                // Length of buffer in bytes.
            &ReturnedLength,      // Bytes placed in outbuf.  Should be 0.
            NULL                // NULL means wait till I/O completes.
                        );

   if (!IoctlResult)
```

```
   {
        printf("Ioctl failed with code %ld\n", GetLastError() );
   }
   return IoctlResult;
}

BOOL GetPortVal(HANDLE hndFile)
{
   IoctlCode = IOCTL_GET_MODE;

   IoctlResult = DeviceIoControl(
            hndFile,        // Handle to device
            IoctlCode,      // IO Control code for Write
            NULL,           // Buffer to driver.  Holds port & data. Not used.
            0,                        // Length of buffer in bytes. Not used.
            &portMode,                // Buffer from driver.
            sizeof(portMode),      // Length of buffer in bytes.
            &ReturnedLength,       // Bytes placed in outbuf.  Should be 0.
            NULL                // NULL means wait till I/O completes.
                        );

   if (!IoctlResult)
   {
        printf("Ioctl failed with code %ld\n", GetLastError() );
   }

   return IoctlResult;
}
#endif
void Usage()
{
   printf("\n\n");
   printf("Usage: \n");
   printf("\tSetInterface  [Port] [Mode]    ; Port=1~16  Mode=0,1,2,3\n");

   printf("\t Mode 0: RS232 \n");
   printf("\t Mode 1: RS485-2W \n");
   printf("\t Mode 2: RS422 \n");
```

```
    printf("\t Mode 3: RS485-4W \n");
    printf("\n");

    printf("e.g.\tSetInterface 3 2    ; Change the COM3: to RS422\n");
    printf("    \tSetInterface 4 3    ; Change the COM4: to RS485-4W\n");
}
```

Non-standard Baudrates

Moxa's UART ASIC, which is used for both the DA-SP08-I-DB and DA-SP08-I-TB serial expansion modules, supports most non-standard baudrates in the range 50 bps to 921.6 Kbps. In fact, supported baudrates are much denser towards the lower values. For example, no baudrates are supported between the integers 5320 and 5323, but 49 baudrates are supported between the integers 387 and 388. Of course this is the way it should be, since serial devices that require using non-standard baudrates generally use slower baudrates.

Before using a serial device that requires using a non-standard baudrate, you must first check that the DA-682 supports a baudrate within the tolerance specified by the serial device manufacturer.

Use the following formula to calculate which baudrates are supported by the DA-283:

(A) Baudrate = 921600/(N+M/8) bps, for N = 1, 2, …, 18431, M = 0, 1, 2, …, 7

or

(B) Baudrate = 8 x 921600/K bps, for K = 8, 9, …, 147456

If you are a programmer and you need to write a driver for your serial device, then you may need to use formula A. If you have a serial device that requires using a non-standard baudrate, then you can use formula B to determine if the DA-682 supports a baudrate within the tolerance specified by the serial device manufacturer.

Example: Your serial device requires using a baudrate of 5340 bps and has a tolerance of 2 bps. Can the DA-682 be used with this device?

Solution: Set formula B to the desired baudrate and then solve for K.

8 x 912600/K = 5338 ==> K = 1367.703259…

This shows that the supported baudrate closest to 5340 comes from setting K=1367 or K=1368.

K=1368 ==> Baudrate1 = 5336.842105...

K=1367 ==> Baudrate2 = 5340.746159…

Since 5338 – Baudrate1 < 2, we can see that the DA-682 supports the serial device.

Note that we can use formula A to generate the so-called "standard" baudrates, which come from setting M=0, and setting N equal to certain integers.

Standard Baudrates					
Baudrate	**N**	**M**	**Baudrate**	**N**	**M**
921600	1	0	4800	192	0
460800	2	0	2400	384	0
230400	4	0	1800	512	0
115200	8	0	1200	768	0
57600	16	0	600	1536	0
38400	24	0	300	3072	0
19200	48	0	150	6144	0
9600	96	0	75	12288	0
7200	128	0	50	18432	0

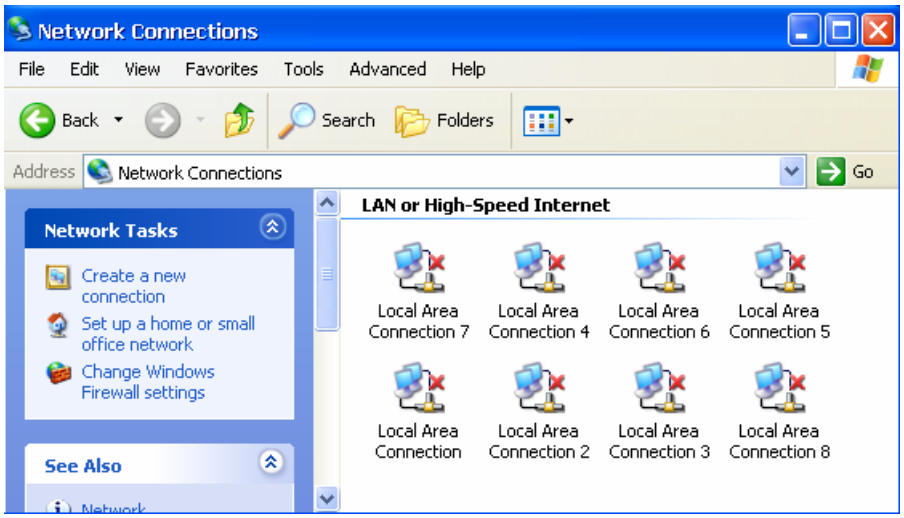
WARNING

Communication between a serial device and a Moxa UART port may not work correctly if the serial device uses a baudrate that it not within the correct tolerance of a baudrate calculated from either formula A or formula B.

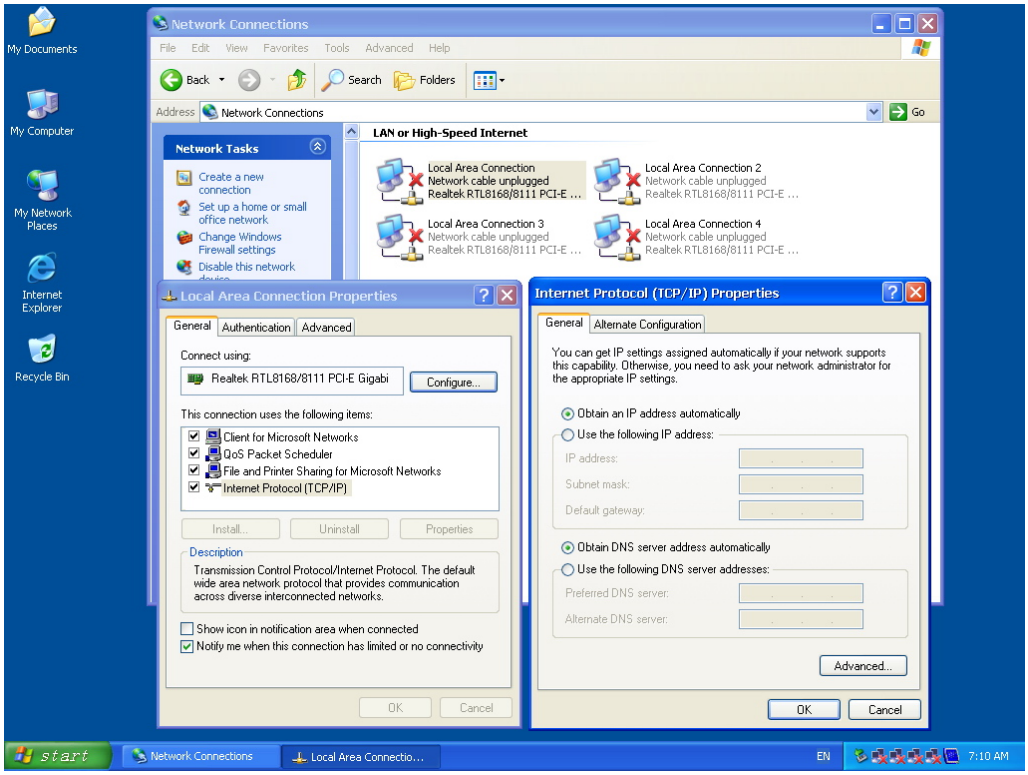
Changing the Network Settings

The DA-682-XPE computer comes with four basic Gigabit Ethernet ports labeled LAN1 to LAN4. The LAN Port Expansion Module supports an additional four 10/100 Mbps Ethernet ports labeled LAN5 to LAN8.

The default IP addresses are DHCP supported. Choose **My Device → Control panel → Network Connections** to enter the network settings page.

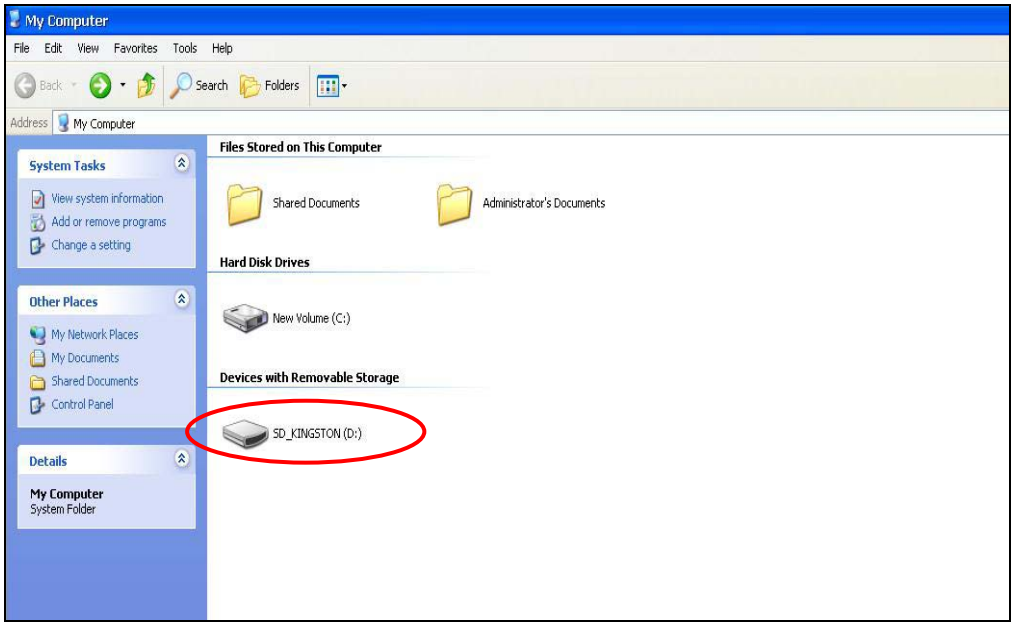


Select the connection and then right-click and choose an option from the pop-up menu. You can specify the IP address manually or by DHCP. In addition, you can disable or enable either one or any combination of connections with the pop-up menu.



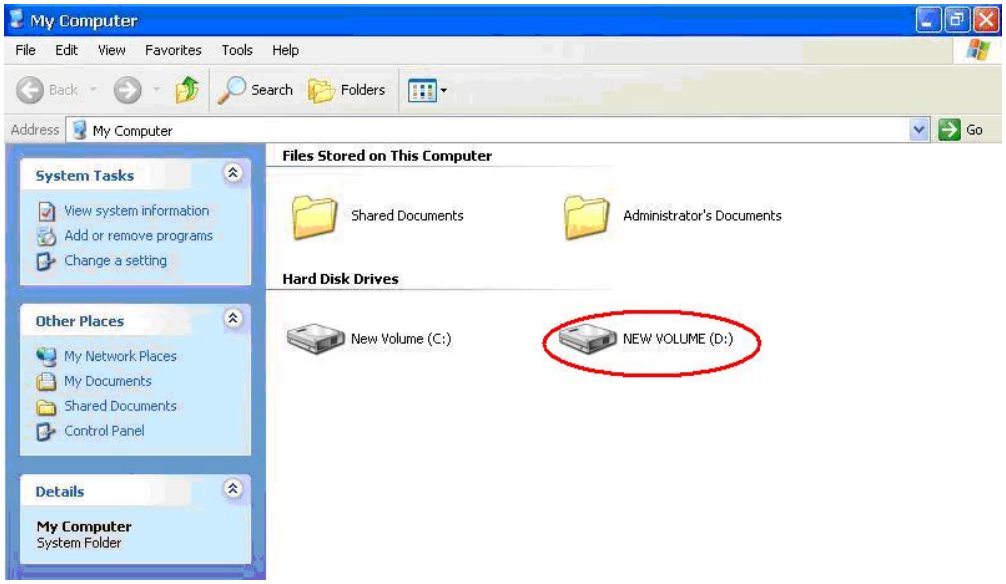
Inserting a USB Mass Storage Device into the Computer

Inserting a USB mass storage device will generate a new drive on the DA-682-XPE. The new drive should be visible in the File Explorer.



Inserting a SATA Hard Disk into a DA-682 Computer

Inserting a SATA hard disk device will generate a new drive in the DA-682-XPE. The new drive will appear in File Manager.



NOTE: The DA-682 computer supplies only 5 VDC of power for a 2.5 inch SATA hard disk. If you want to use a 3.5 inch hard disk, you will need to use an external power supply to power the SATA hard disk.

How to Determine the Firmware Build Version

Use the mxver command to obtain the firmware version. This information is particularly important for identifying which features your embedded computer supports.

Execute **mxver.exe** from the command line:

C:\> mxver

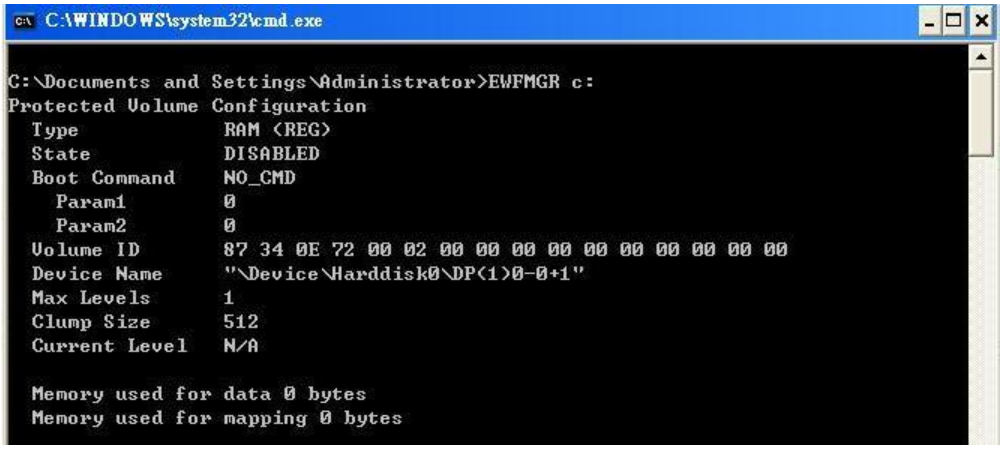
The response should appear similar to the following:

Model Name: DA-682-XPE
Version: 1.0
Build Date: 08040711

Enhanced Write Filter

The Enhanced Write Filter protects the contents of a volume on the target media by redirecting all writes to another storage location called on overlay. You can control EWF either by using the EWF APIs or embedded command-line utility. To use the EWF APIs, you need to include "**ewfapi.h**" and "**ewfapi.lib**" from the software CD to your project. To use the command-line utility, take the following steps to enable the Enhanced Write Filter.

1. Type **EWFMGR C**: to check if the Enhanced Write Filter is disabled.
2. Type **EWFMGR C**: **-enable** to enable the Enhanced Write Filter.
3. Reboot the system to activate the Enhanced Write Filter.
4. Delete a file on your protected volume and reboot the system. The file you just deleted will appear.



3

Management Tools

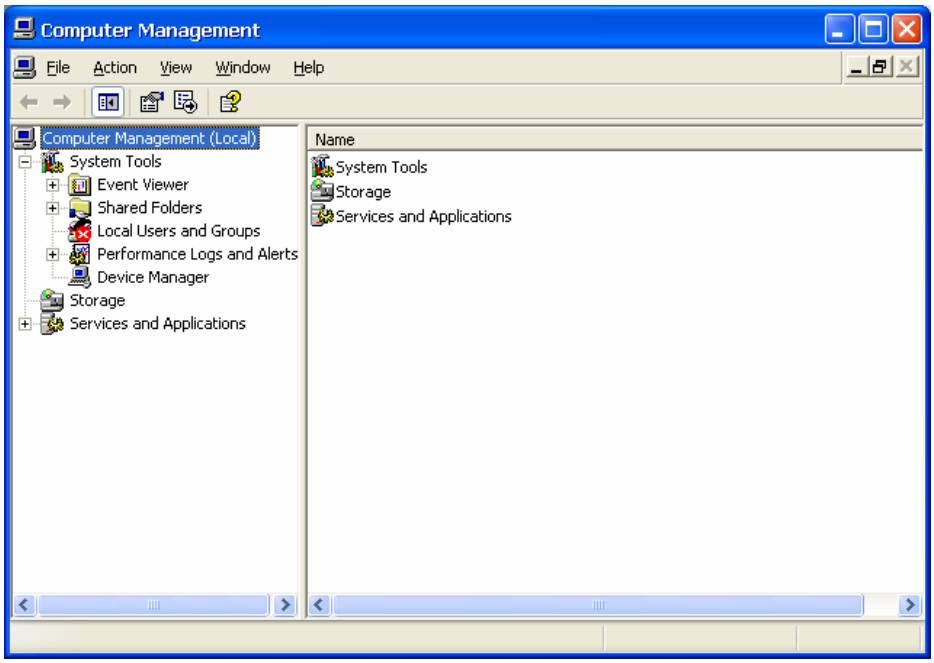
The DA-682-XPE ready-to-run embedded computer is shipped with the Window XP Embedded operating system pre-installed. This network-centric platform is designed to serve as the front-end for data acquisition and industrial control applications. A set of Windows XP management tools is installed on the DA-682-XPE computer to help with management issues.

Computer Management

[Control Panel] → [Administrative tools] → Computer Management

You can use the tools for a variety of tasks, such as disk partition, disk mount/dismount, and create/remove users.

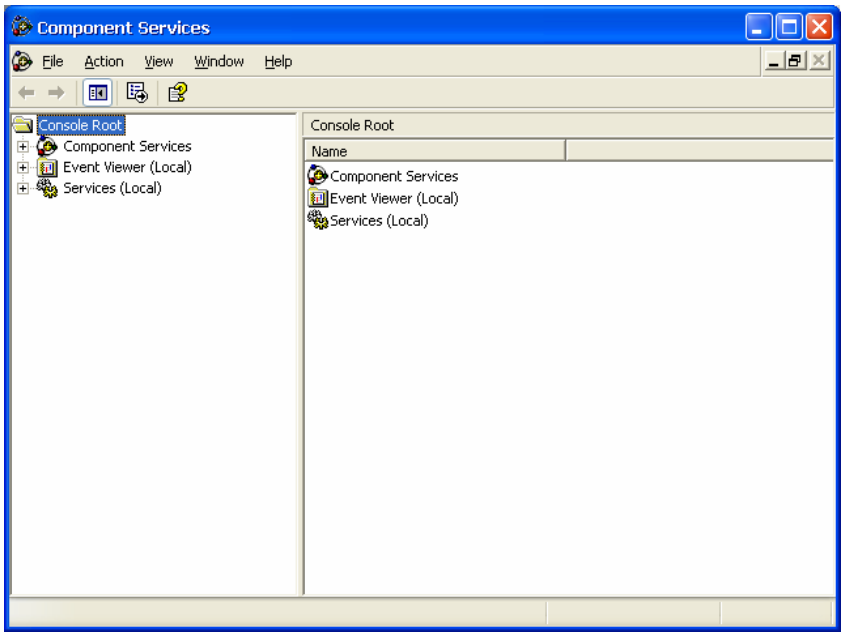
You can also check services in the Computer Management window.



Component Services

[Control Panel] → [Administrative tools] → Component Services

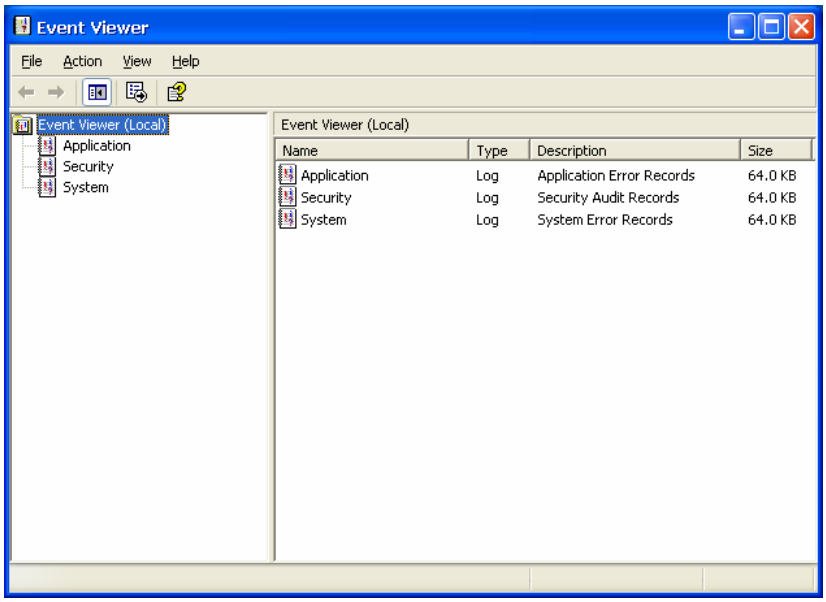
You can install/view/remove COM components with this tool.



Event Viewer

[Control Panel] → [Administrative tools] → Event Viewer

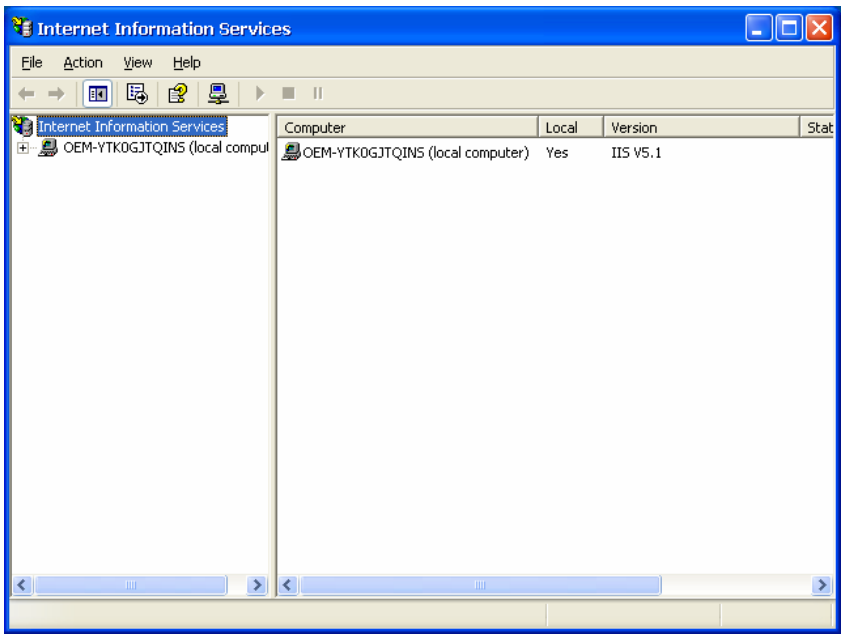
Every DA-682-XPE event, including system, applications, and security events are logged in this event database.



Internet Information Services (Web/FTP)

[Control Panel] → [Administrative tools] → Internet Information Services

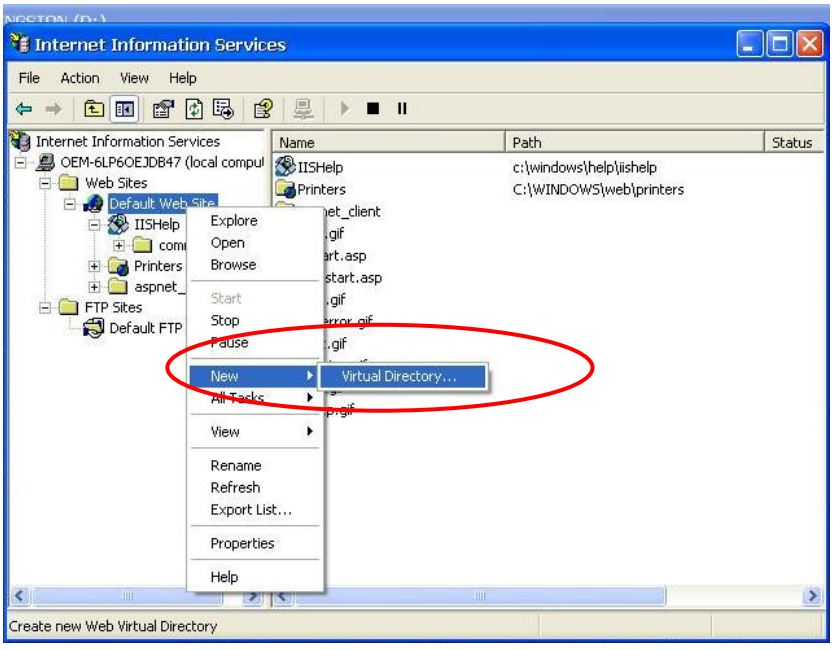
You must use this tool to configure web or FTP services. You can also use the tool to start and stop HTTP/FTP services.



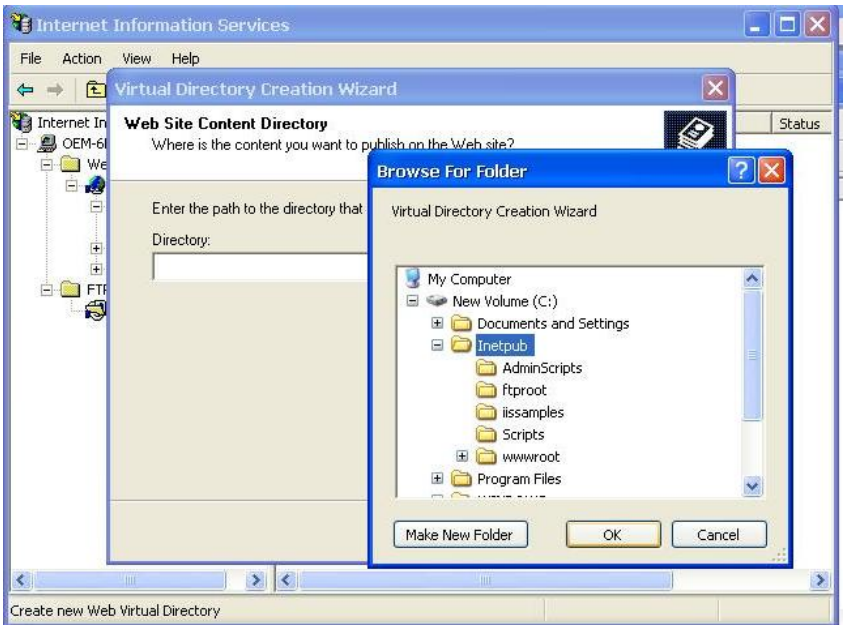
A default web page is located in the directory **c:\Inetpub**. Use this default page to test your web server.

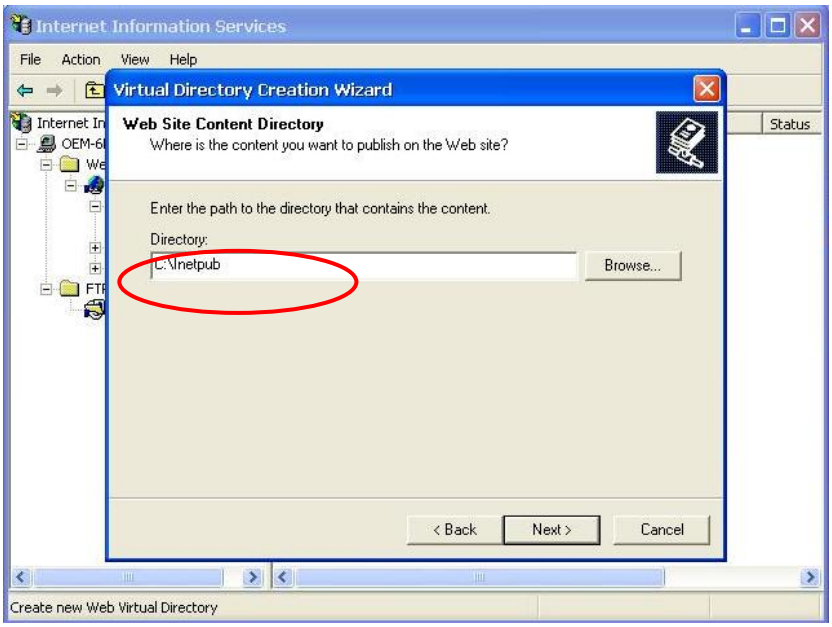
Follow the steps shown below to create the virtual directory.

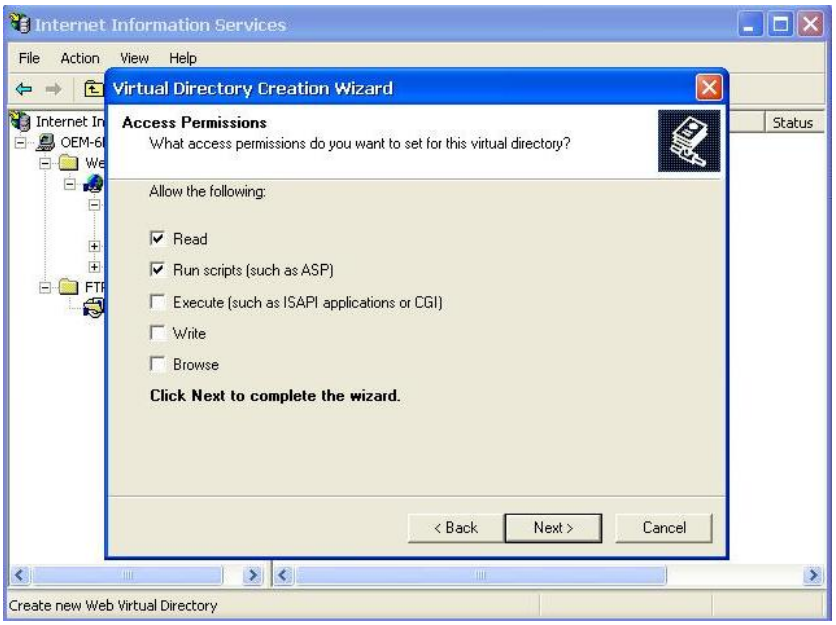
1. Create a virtual directory by selecting **Default Web Site → New → Virtual Directory.**



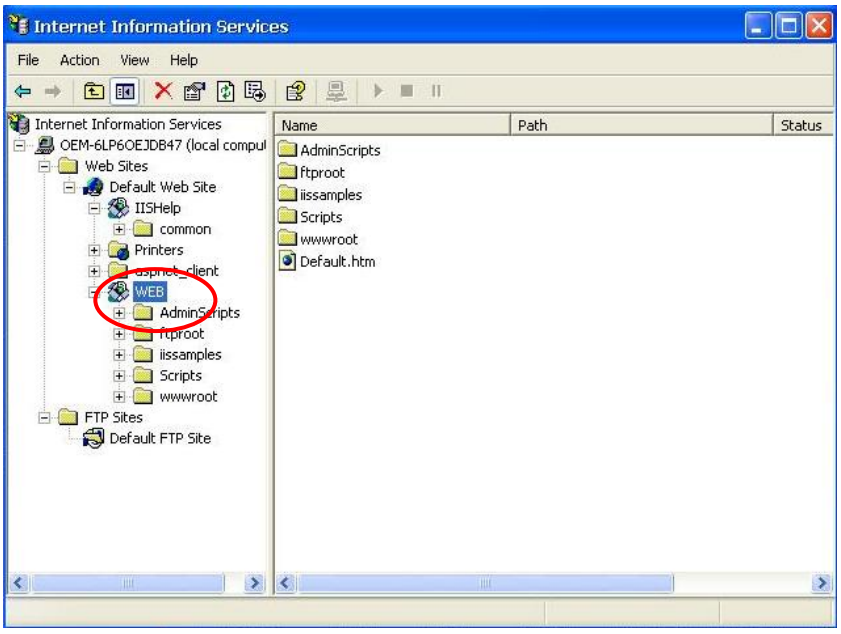
2. Follow the virtual directory creation wizard and complete the steps to create the virtual directory **c:\Inetpub**, as indicated in the following series of screen shots.



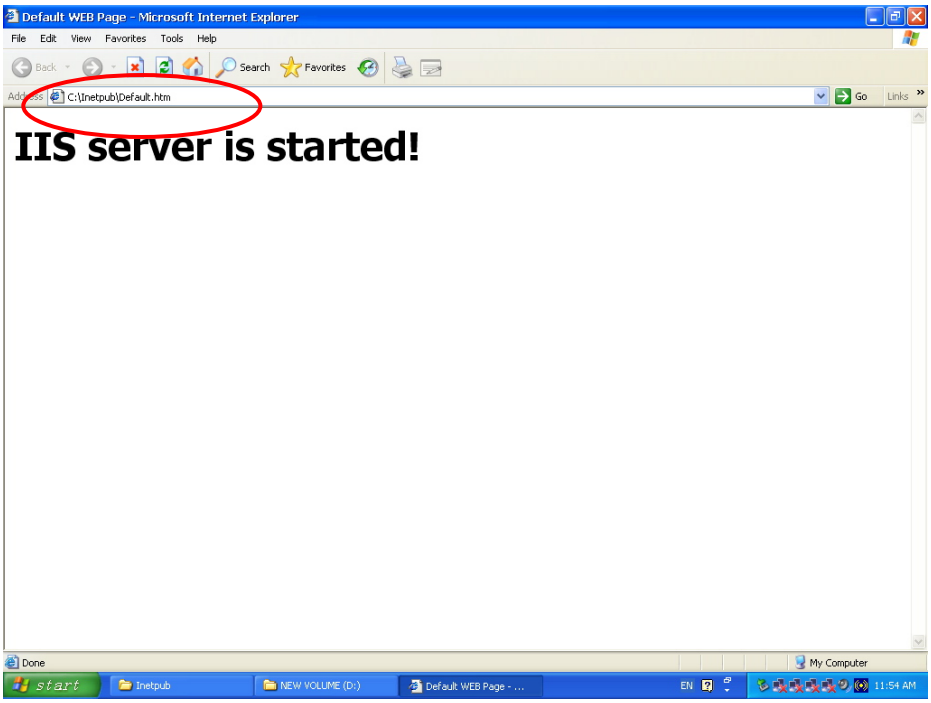

Internet Information Services
File Action View Help
Virtual Directory Creation Wizard
Web Site Content Directory
Where is the content you want to publish on the Web site?
Enter the path to the directory that contains the content.
Directory:
C:\Inetpub
Browse...
< Back
Next >
Cancel
Status
Create new Web Virtual Directory


Internet Information Services
File Action View Help
Virtual Directory Creation Wizard
Access Permissions
What access permissions do you want to set for this virtual directory?
Allow the following:
Read
Run scripts (such as ASP)
Execute (such as ISAPI applications or CGI)
Write
Browse
Click Next to complete the wizard.
< Back
Next >
Cancel
Status
Create new Web Virtual Directory

3. When you complete the steps, the virtual directory **WEB** will appear under **Default WEB Site**.

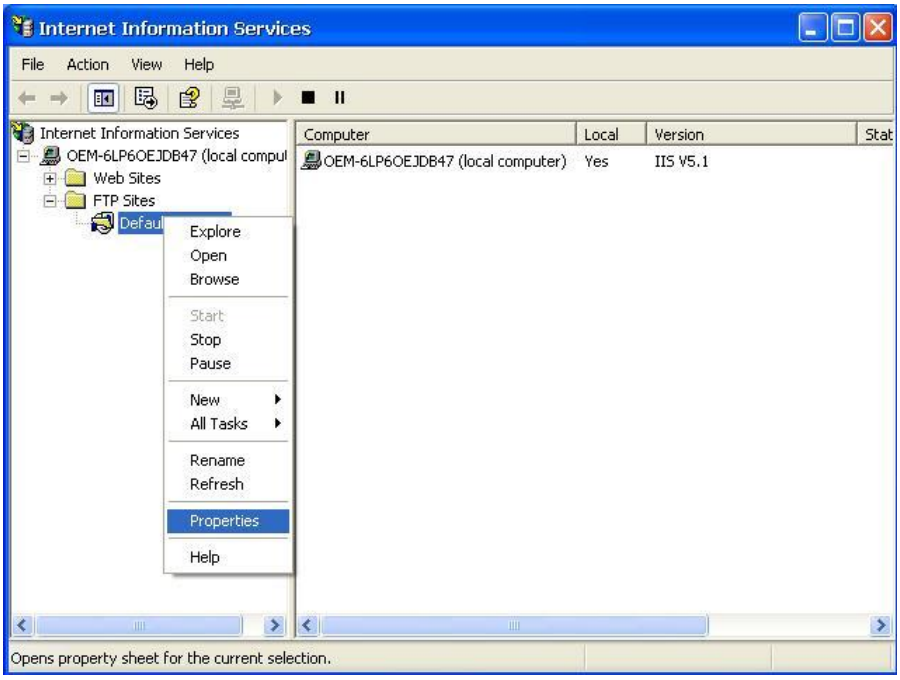


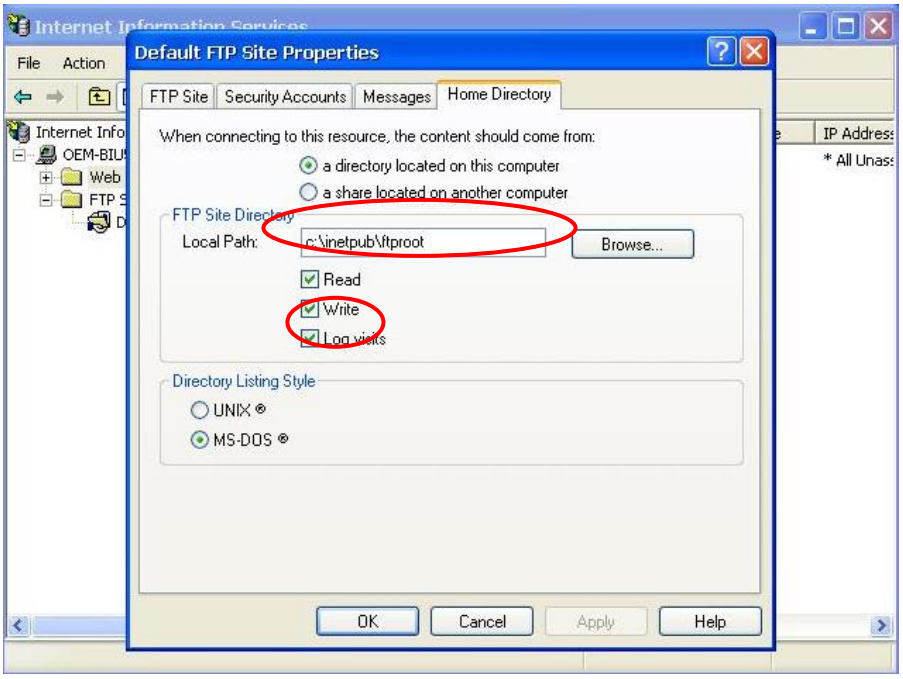
4. Open IE, type **C:\Inetpub\Default.htm** or **[*IP Address*]/WEB/Default.htm** (for example, 192.168.3.127/WEB/Default.htm). The following page should appear.



If you need to use the FTP server, you must create the default password for your account and turn on the write permission on your home directory, which is located in **c:\intepub\ftproot**.

Select **FTP Sites → Default FTP Site → Properties → Home Directory**, and checkmark the **Write** checkbox. You should now be able to transmit files through the ftp server.

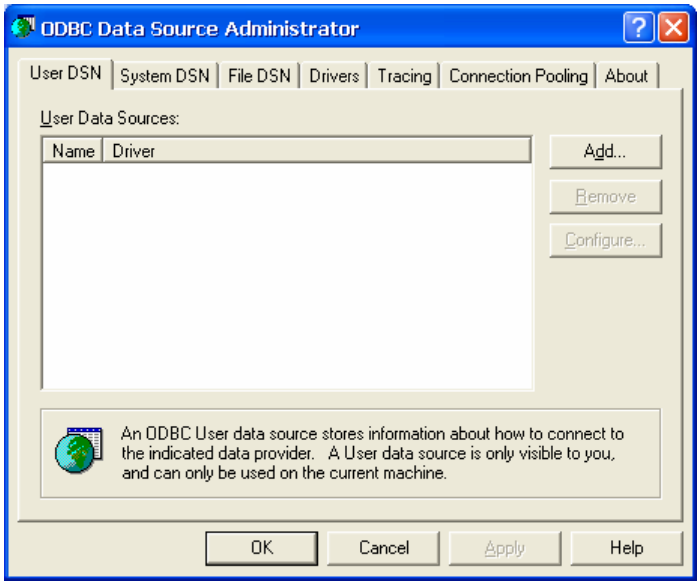




ODBC Data Source Administrator

[Control Panel] → [Administrative tools] → ODBC.

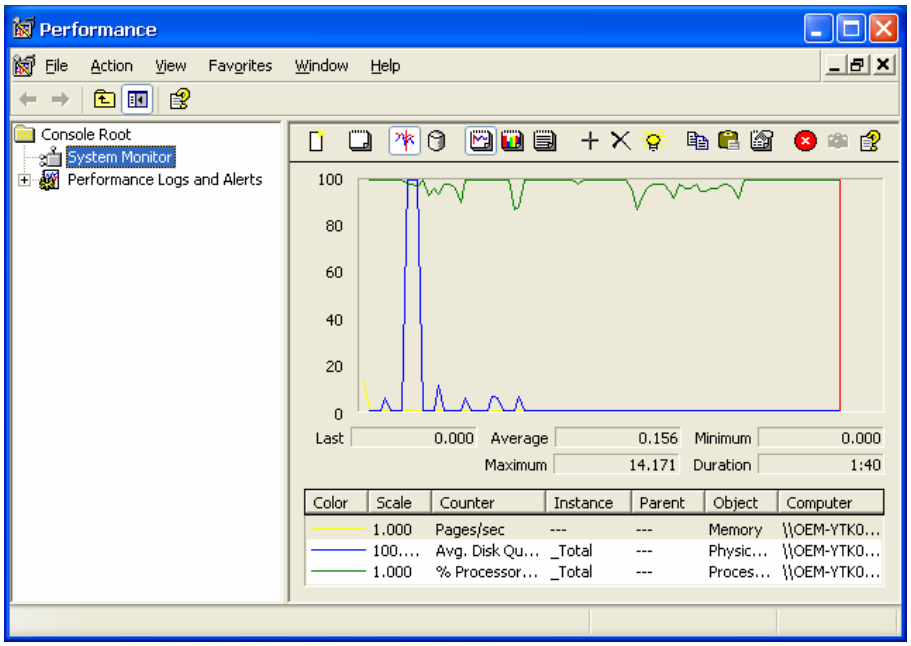
This is a database source configuration tool for users to add, delete, or set up a data source and display information about the installed ODBS drivers. You can create a new data source or trace the calls to ODBC functions.



Performance Monitor

[Control Panel] → [Administrative tools] → Performance

You can use this tool to monitor system and network resources.



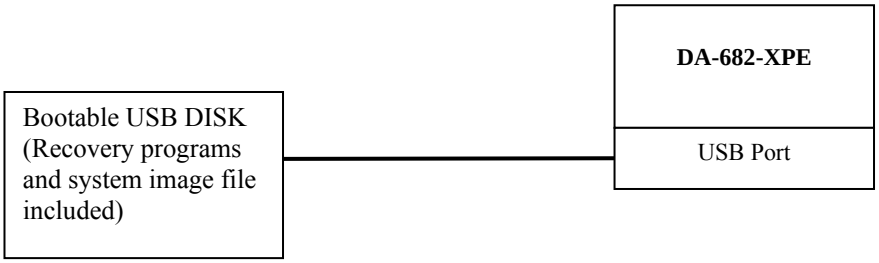
4

System Recovery

The DA-682-XPE ready-to-run embedded computers are Windows XP Embedded platforms. Although it rarely happens, you may find on occasion that operating system files are damaged. This chapter describes how to recover your DA-682-XPE to a normal status.

Recovery Environment

The recovery environment includes the DA-682-XPE embedded computer and a bootable USB disk with the recovery programs and system image file.



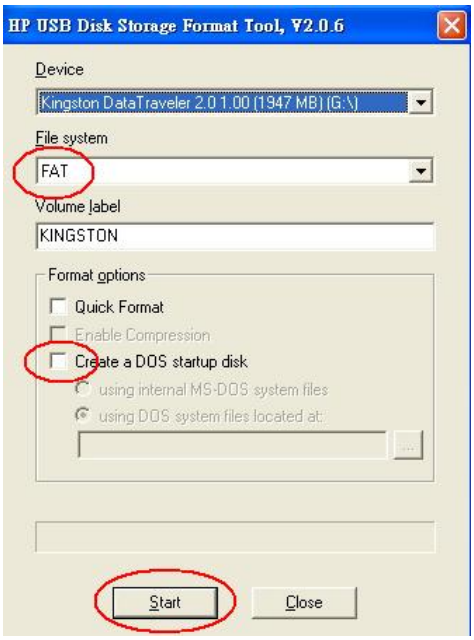
Recovery Procedure

This section describes the recovery procedure.

Step 1: Format an Empty USB Disk.

a. Format your USB disk with the **HP USB Disk Format Tool**, open the utility, and then select the device and FAT file system. Use an empty disk, and DO NOT check the option **Create a DOS startup disk**.

b. Click **"Start"**.



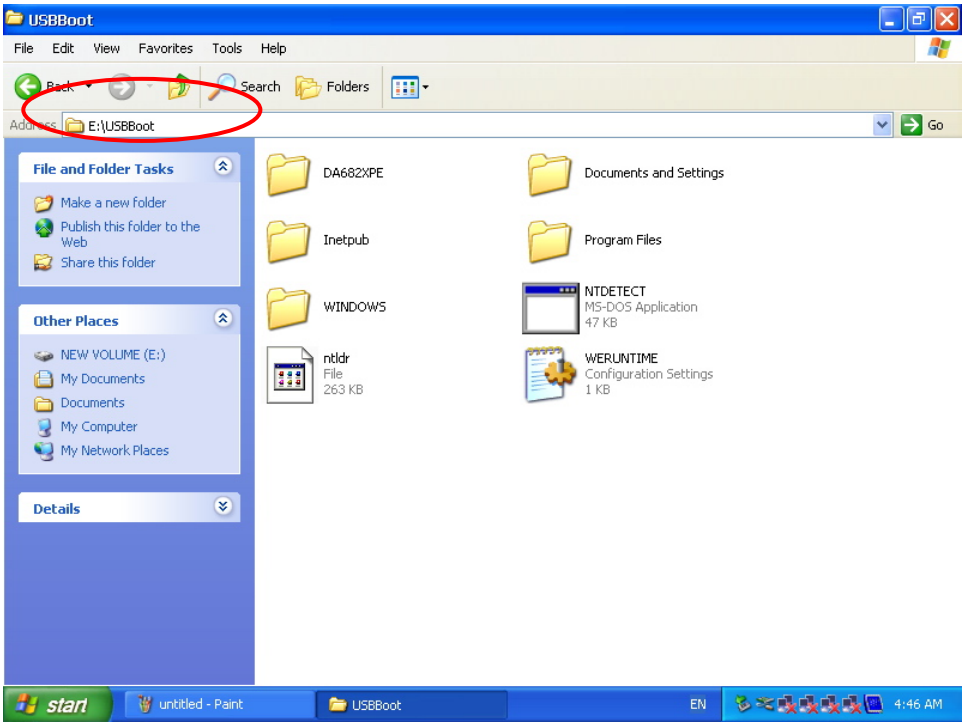
ATTENTION

HP USB Disk Storage Format Tool can be downloaded from many web sites, type "**HP USB Disk Storage Format Tool**" to search internet and download it.

Step 2: Create a Windows XP Embedded Bootable USB Disk.

a. Configure Windows Explorer to show hidden files (including protected operating system files).

b. Copy all files in the **USBBoot** directory from the DVD to the root directory of your USB disk.



Step 3: Set up the BIOS to Boot from the USB Disk.

a. Insert the USB disk.

b. Power on the computer and press **DEL** on the keyboard to enter the bios setup menu.

c. Select **Advanced → Hard Disk Boot Priority** and then press **Enter**.

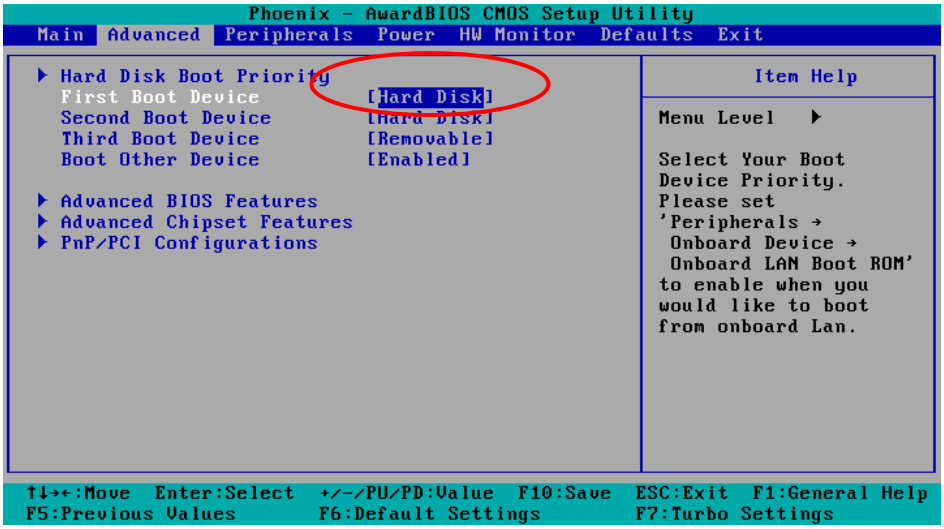
d. From the setup menu, use "↑" or "↓" to select the USB device.

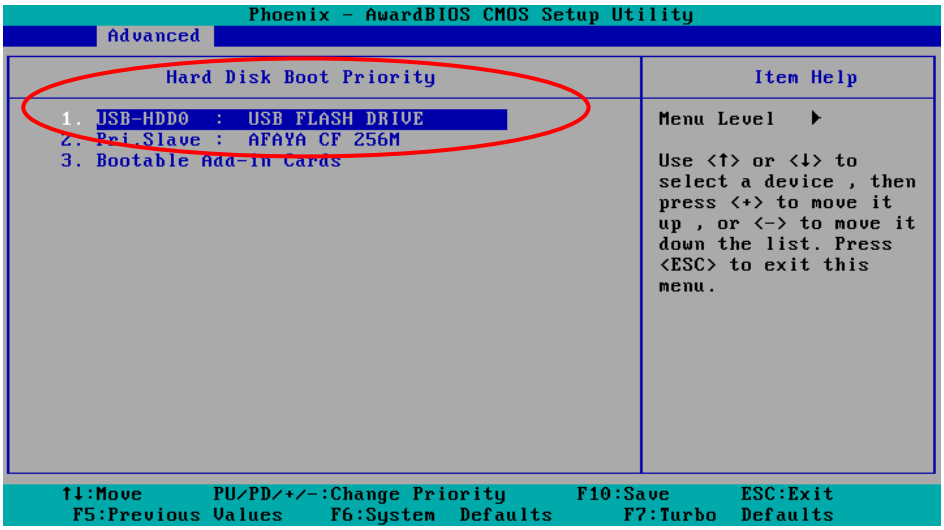
e. Press "+" to move the selection up to the first priority, and then press "**Esc**" to exit the setup menu.

f. Make sure the first boot device is "Hard Disk". If not, press Enter to change it.

g. Select **Exit → Save & Exit Setup** and then press **Enter**.

h. Choose **Y** to save to CMOS and then exit.





Step 4: Copy Windows XPE system files to the DOM or CompactFlash Disk.

a. If the BIOS setup is correct, it will restart and boot from the USB disk.

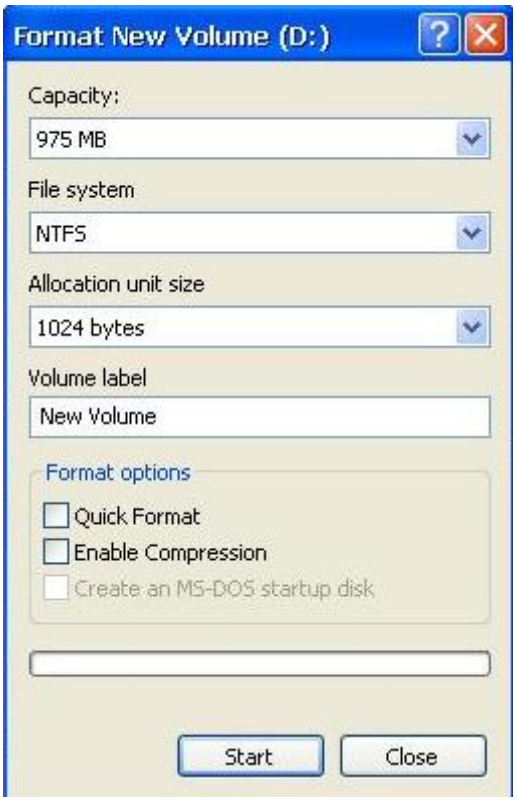
b. Select and right-click the hard disk you want to recover (e.g., New Volume D: for DOM or New Volume E: for CF card), and then select **Format**.

c. Select the NTFS file system option and click **Start**.

d. Configure Windows Explorer to show hidden files (including protected operating system files).

e. Close the format utility window and copy all files in the **DA682XPE** directory from the USB disk to the hard disk.

f. When the operation is complete, turn off the computer and remove the USB disk.



Step 5: Reset the BIOS to boot from the DOM or CompactFlash disk.

a. Power on the computer and press **DEL** to enter the bios setup menu.

b. Select **Advanced → Hard Disk Boot Priority** and then press **Enter**.

c. From the setup menu, use "↑" or "↓" to select the DOM or CompactFlash device.

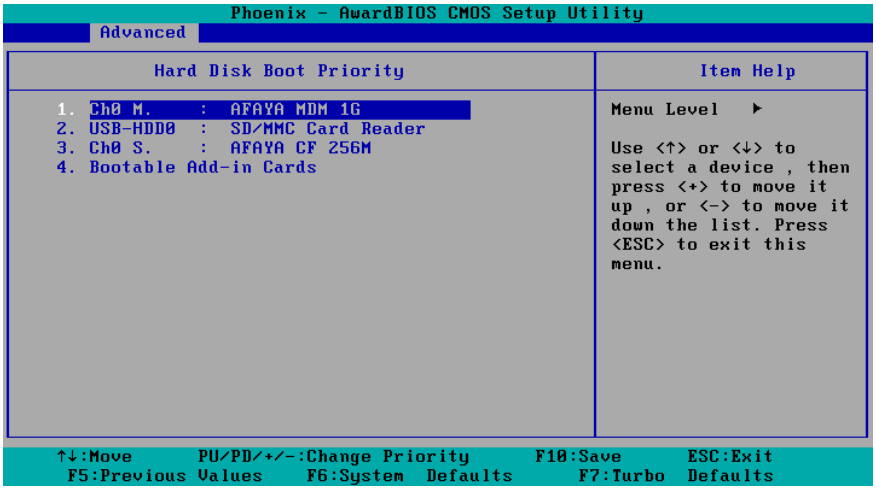
d. Press "+" to move the selection up to the first priority, and press then **Esc** to exit the setup menu.

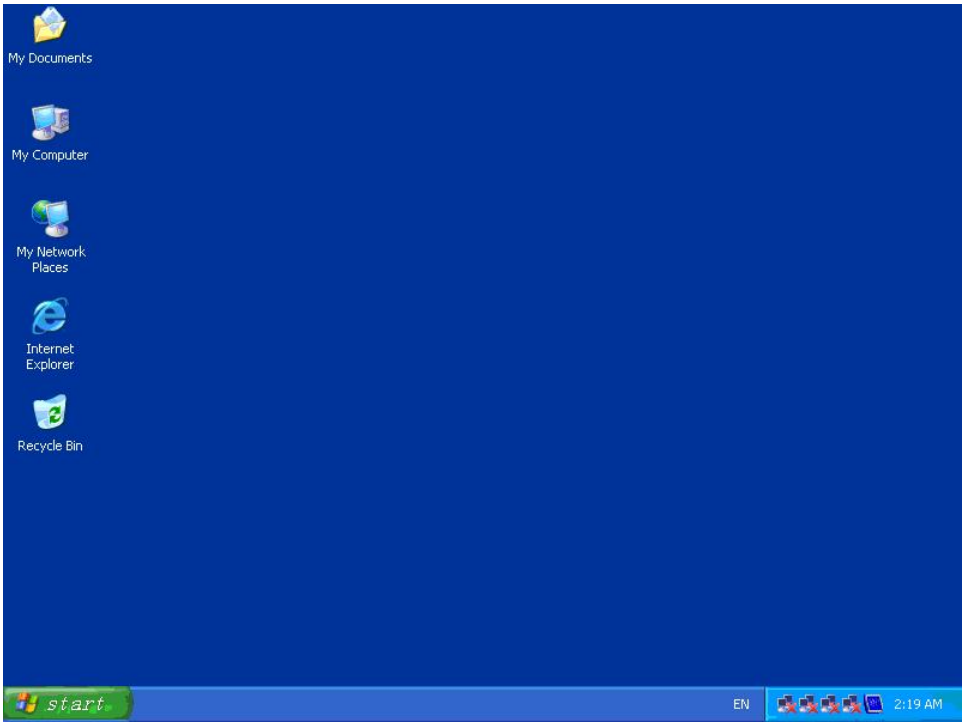
e. Select **Exit → Save & Exit Setup** and then press **Enter**.

f. Choose **Y** to save to CMOS and then exit.

g. Wait about 10 or 15 minutes for the system to recover. When the recovery process is finished, you should be able to see the Windows XP Embedded desktop. At this point, restart your computer for the new settings to take effect. Otherwise, some of the services (e.g., MSMQ) will not start automatically.

NOTE: DO NOT turn off the power during this operation, or the system might crash.


Phoenix - AwardBIOS CMOS Setup Utility
Advanced
Hard Disk Boot Priority
1. Ch0 M. : AFAYA MDM 1G
2. USB-HDD0 : SD/MMC Card Reader
3. Ch0 S. : AFAYA CF 256M
4. Bootable Add-in Cards
Item Help
Menu Level ▶
Use <↑> or <↓> to select a device , then press <+> to move it up , or <-> to move it down the list. Press <ESC> to exit this menu.
↑↓:Move PU/PD/+/-:Change Priority F10:Save ESC:Exit
F5:Previous Values F6:System Defaults F7:Turbo Defaults


My Documents
My Computer
My Network Places
Internet Explorer
Recycle Bin
start
EN
2:19 AM