DA-682A-DPP Windows Software User's Manual

Edition 1.0, March 2016

www.moxa.com/product

DA-682A-DPP Windows Software User's Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

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: +86-21-5258-5505

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

Moxa India
Tel: +91-80-4172-9088
Fax: +91-80-4132-1045

Table of Contents

1

Introduction

Thank you for buying Moxa's DA-682A-DPP rackmount computer. This model comes with the Windows 7 Embedded software platform.

- **Software Components**

Software Components

The following table details the software components of the Windows Embedded Standard 7 on this computer.

Windows Embedded Standard 7

Core OS:

- 32-bit support
- Remote Client
- Remote Procedure Call

Applications and Services Development:

- .Net Framework 3.5
- Remote Desktop Protocol 7.1
- COM OLE Application Support
- COM+ Application Support
- MSMQ

Internet Services:

- Internet Explorer 8.0
- IIS 7.0

File Systems and Data Store:

- Windows Data Access Components
- Windows Backup and Restore

Diagnostics:

- Common Diagnostic Tools
- Problem Reports and Solutions

Fonts: Chinese (Traditional . and Simplified), Japanese, Korean, Western, Middle Eastern, South East Asian, and South Asian Fonts

Graphics and Multimedia:

- MPEG DTV-DVD Audio Decoder (MPEG-2, AAC)
- MPEG Layer-3 Audio Codecs(MP3)
- MPEG4 Decoders
- Windows Media Video VC-1 (WMV) Codecs
- DirectX and Windows Device Experience
- Windows Media Player 12

Management:

- Group Policy Management
- Windows Management Instrument (WMI)
- Windows Update

Networking:

- Extensible Authentication Protocol (EAP)
- Internet Authentication Service
- Telnet Server
- Bluetooth
- Domain Services
- Network Access Protection
- Network and Sharing Center
- Quality of Service
- Remote Access Service (RAS)
- Telephony API Client
- Windows Firewall
- Wireless Networking

Security:

- Credential Roaming Service
- Credentials and Certificate Management
- Windows Authorization Manager (AZMAN)
- Windows Security Center
- Active Directory Rights Management
- Security Base
- Encrypted File System (EFS)

Embedded Features:

- Enhanced Write Filter (EWF)
- File-Based Write Filter (FBWF)
- Message Box Default Reply
- Registry Filter
- WSDAPI for .NET

Embedded Self-Health Diagnostic Software: SNMP-based remote scripting layer for monitoring, reporting, and control

2

System Initialization

This chapter describes how to initialize the system settings after booting up the computer for the first time.

The following topics are covered in this chapter:

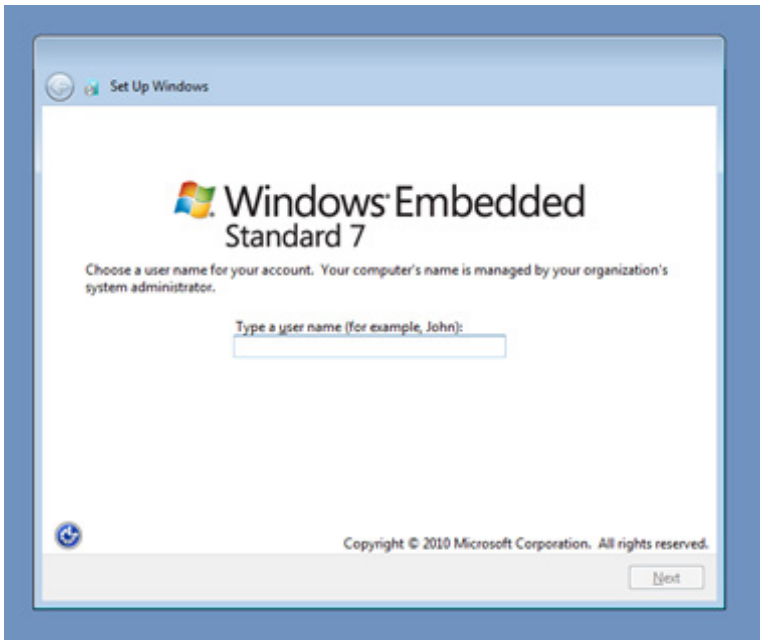
- **Setting Up an Administrative Account and Creating User Accounts**
  - Basic Administrative Setup

Setting Up an Administrative Account and Creating User Accounts

After booting into the computer for the first time, administrators will need to create their own account and set up user accounts.

Basic Administrative Setup

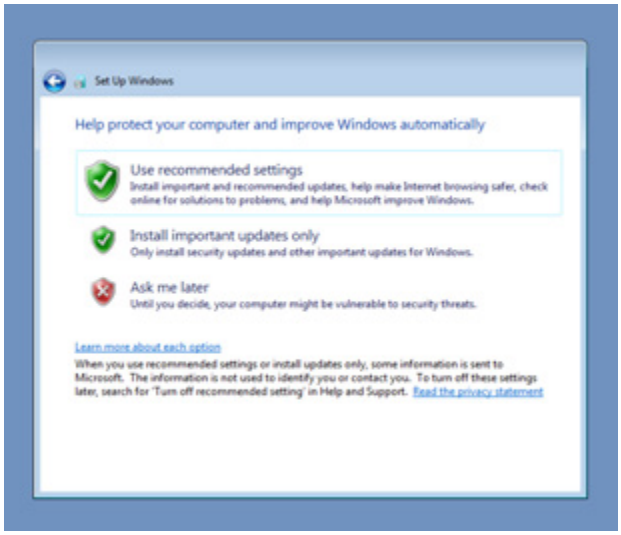
1. Boot up the computer.
2. In the **Set Up Windows** dialog, configure the user name and password for the system. Type in the user name first and follow the instructions of the set up wizard.

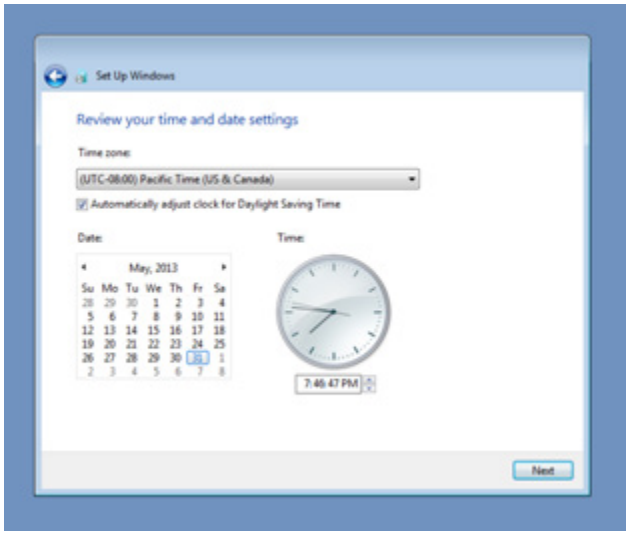


You can also configure a password hint, which will be shown in case you forget your password.

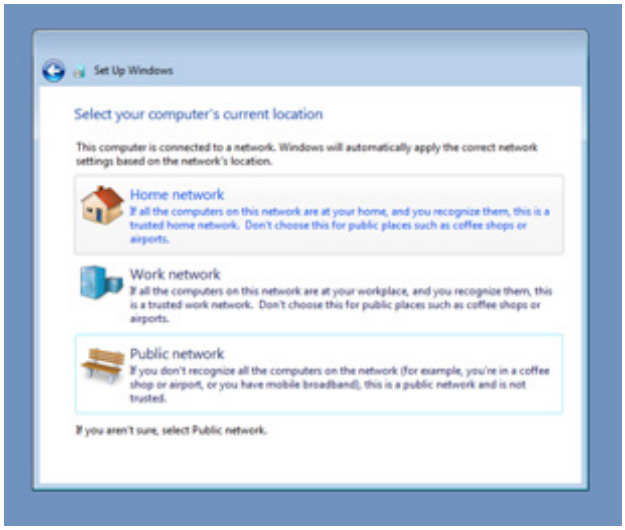
If you do not want to set a password for the administrative account (we strongly advise against doing so), leave the entry box for the password blank and click **Next**.

3. Select the **Windows Update** option, and then select your time zone and whether you will use daylight savings time.





4. Select the network environment for the computer. Windows will automatically apply a preset bundle of security settings based on the network environment in which your computer is located. The most restrictive environment is Public network, and the least restrictive is Home network.



You can start using the computer.

3

Enabling Windows 7 Write Protection

This chapter describes how to set up and configure Windows 7 bit-level and file-level write protection on block storage volumes.

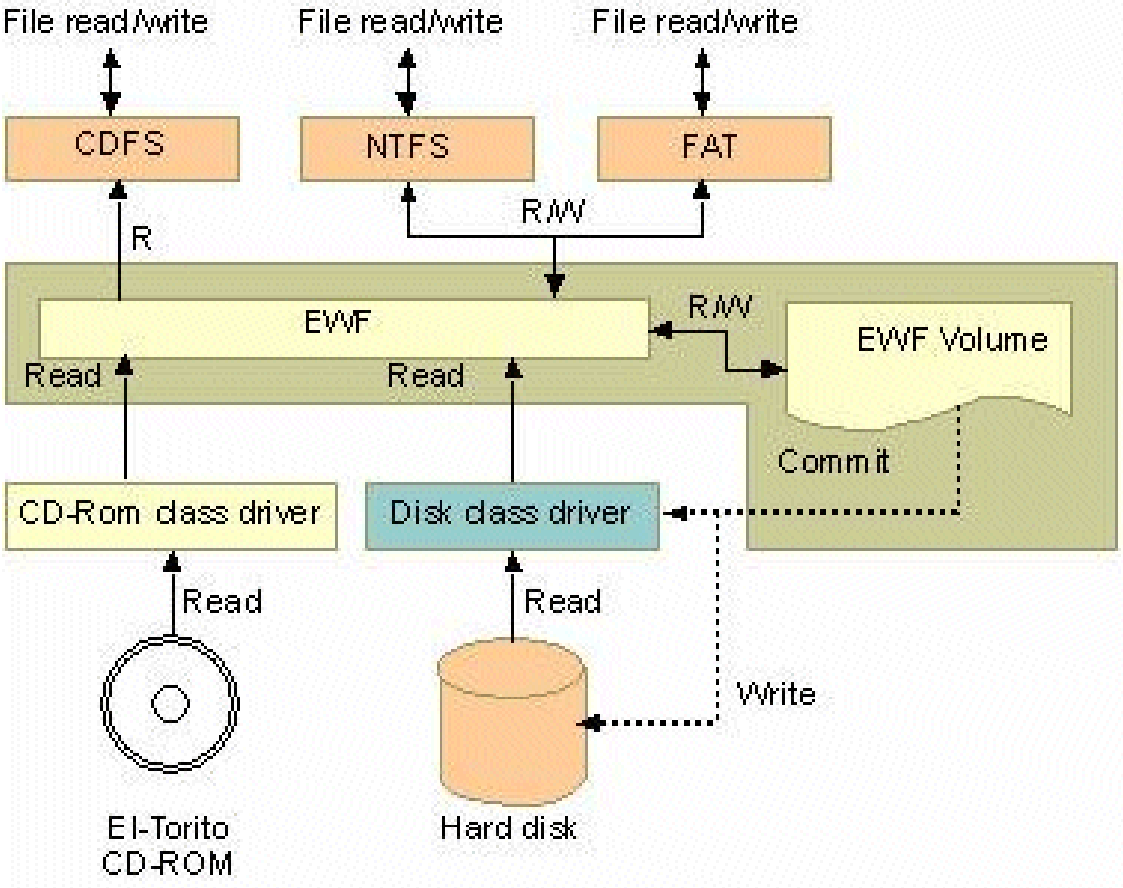
The following topics are covered in this chapter:

- **Enhanced Write Filter**
  - Overview
  - Enabling Enhanced Write Filter
- **File-Based Write Filter**
  - Overview
  - Enabling File-Based Write Filter

Enhanced Write Filter

Overview

The *Enhanced Write Filter (EWF)* allows Windows 7 users to protect the information on their storage drive from permanent changes of any sort. The write protection is implemented at the bit level, which is at the lowest level of the hardware and is therefore more secure. EWF allows the operating system (OS) to boot from the hard disk, but protects the system by creating a virtual file system called an *overlay*. All writes to a EWF-protected volume (the **Hard disk** in the figure below) are only recorded on this virtual overlay (the **EWF Volume** in the figure below), which is stored independently in random access memory (RAM). Because EWF does not write data directly to the hard disk but instead only records system writes to this virtual RAM overlay, any data that is "written" during system operation will not be stored upon the next re-boot. This approach allows the system to operate as if it is writeable when in reality the data is written to a virtual overlay. The OS and user-space file systems are stored in a permanent read-only state. You can commit the data from the overlay to the protected volume, but this requires additional effort. The structure of the enhanced write filter is illustrated in the following figure (source Microsoft):



Additional details on EWF configuration and usage are available on the Microsoft website at:

- EWF overview
- EWF Volume Configuration
- EWF modes
- EWF API

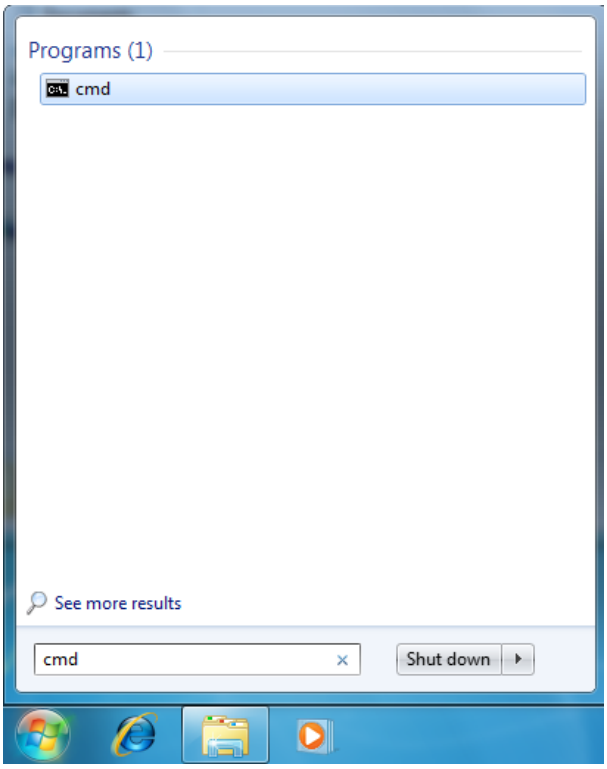
For EWF commands, refer to the MSDN web site:

http://msdn.microsoft.com/en-us/library/ms940853%28v=winembedded.5%29.aspx

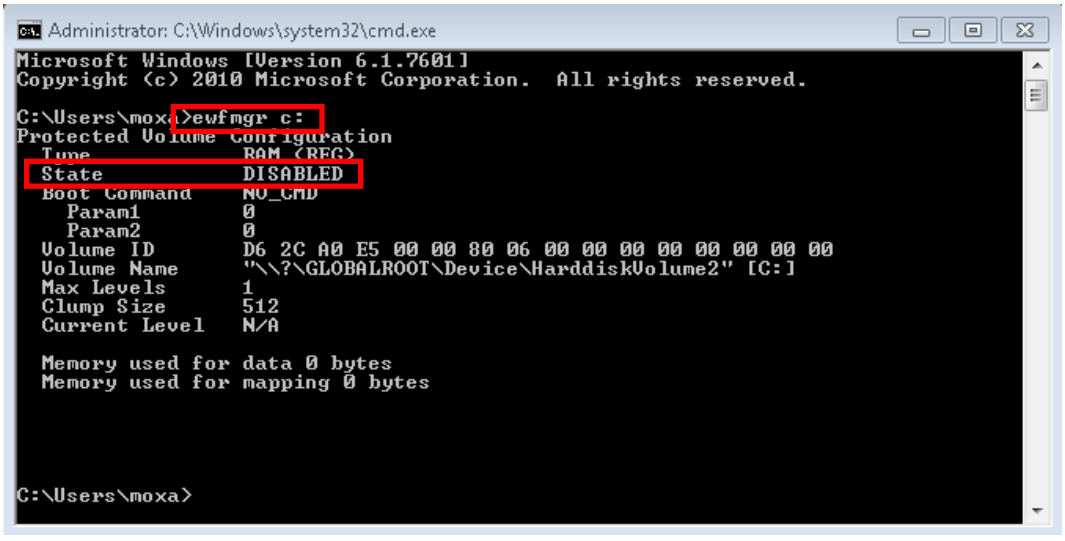
Enabling Enhanced Write Filter

Follow these steps to enable the Enhanced Write Filter (EWF):

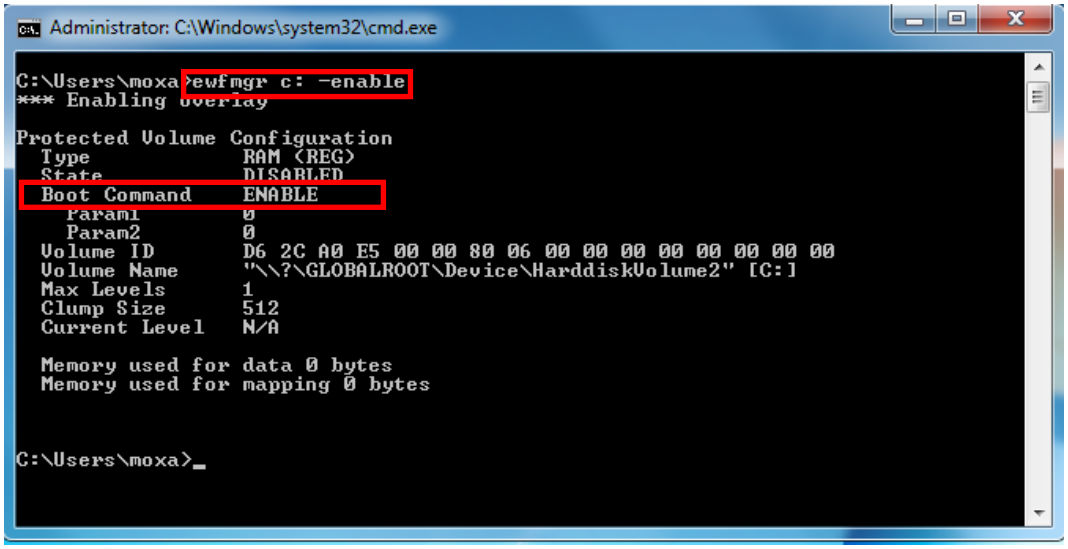
1. Open the command line console by running the **cmd.exe** file.



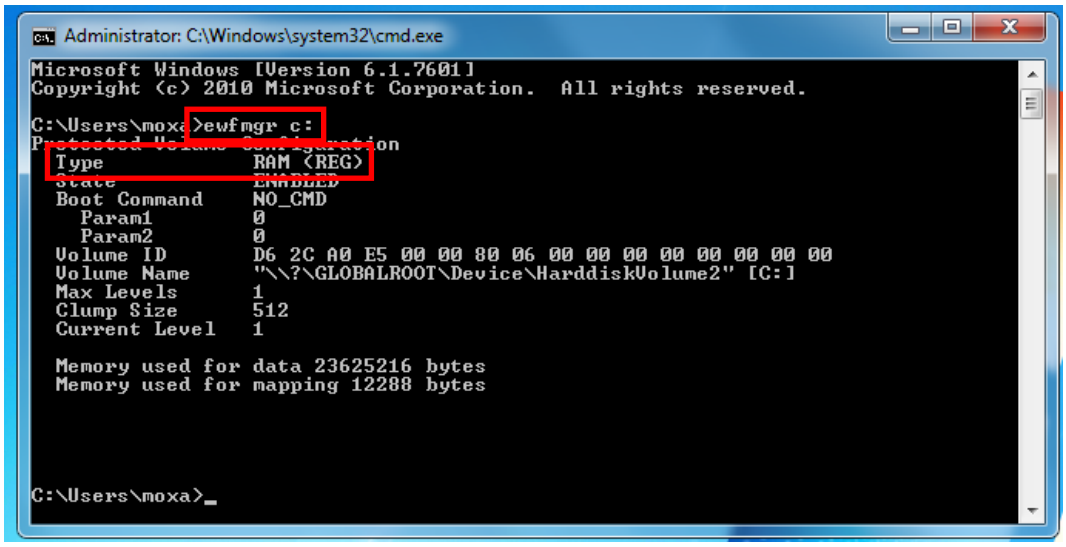
2. To verify that the EWF is disabled, type **C:\....>ewfmgr c:** at the prompt.



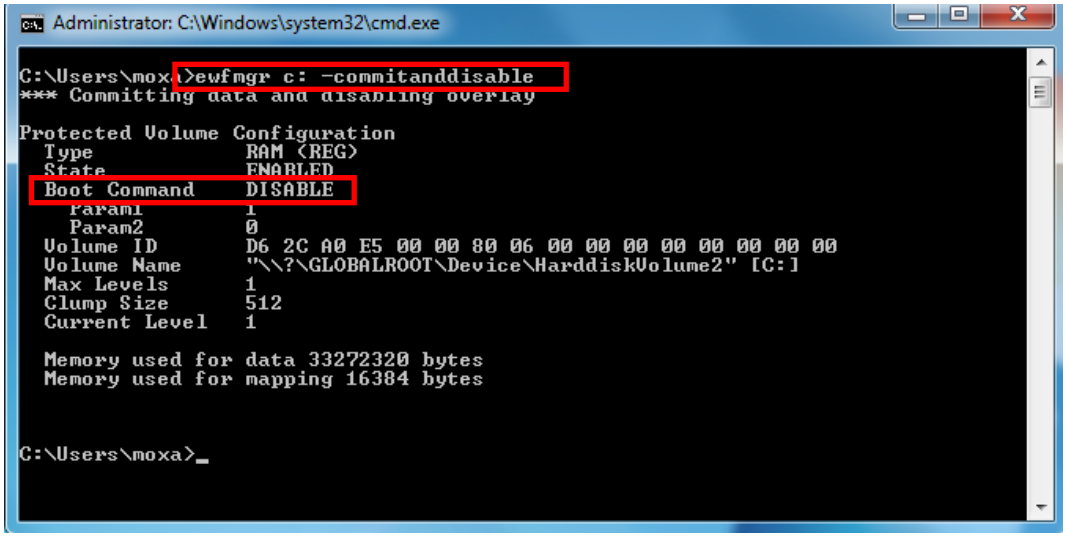
3. To enable the EWF, type **C:\....>ewfmgr c: -enable** at the prompt.



4. Reboot the system for the changes to take effect, and then verify that the EWF has been enabled by using the **C:\....>ewfmgr c:** command.



To disable the EWF, use the **C:\....>ewfmgr c: -commitanddisable** command.



File-Based Write Filter

Overview

The *File-Based Write Filter (FBWF)* is similar to Enhanced Write Filter (EWF), but is enforced at the file level rather than at the hardware (bit) level. Compared to the EWF, the File-Based Write Filter is less secure, but provides more features.

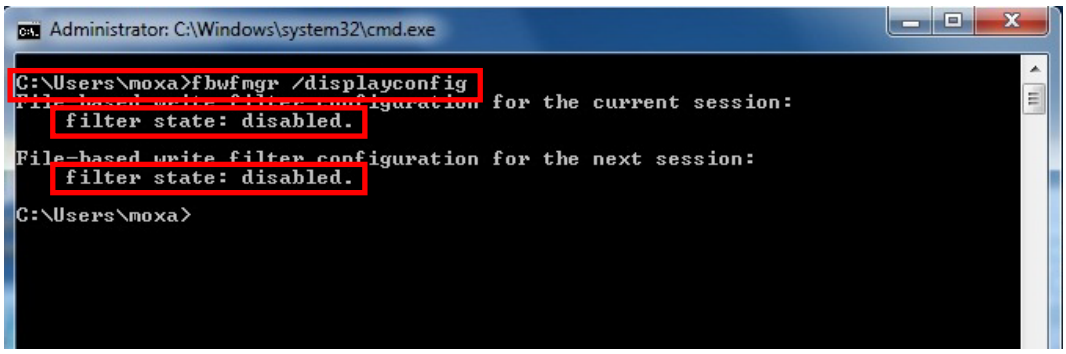
FBWF allows the Windows Embedded platform to maintain the appearance of read and write access on write-sensitive or read-only storage. FBWF makes read and write access transparent to applications. Writing to storage media may be undesirable or impossible in embedded devices. FBWF redirects all writes targeted for protected volumes to a RAM cache called an *overlay*. An overlay is similar to a transparency overlay on an overhead projector. Any change made to the overlay affects the picture as seen in the aggregate, but if the overlay is removed, the underlying picture remains unchanged.

One of FBWF's advanced features allows the user to specify a directory in the data drive to which the data can be written, which is not possible in the case of EWF.

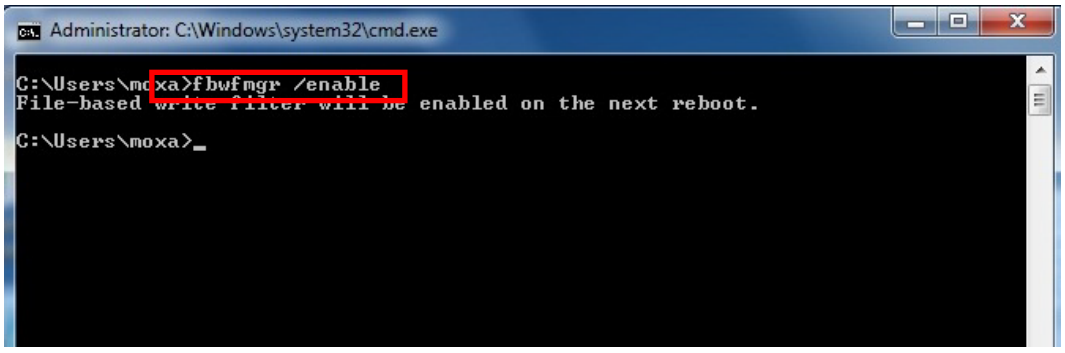
Enabling File-Based Write Filter

To enable file-based write filtering, do the following:

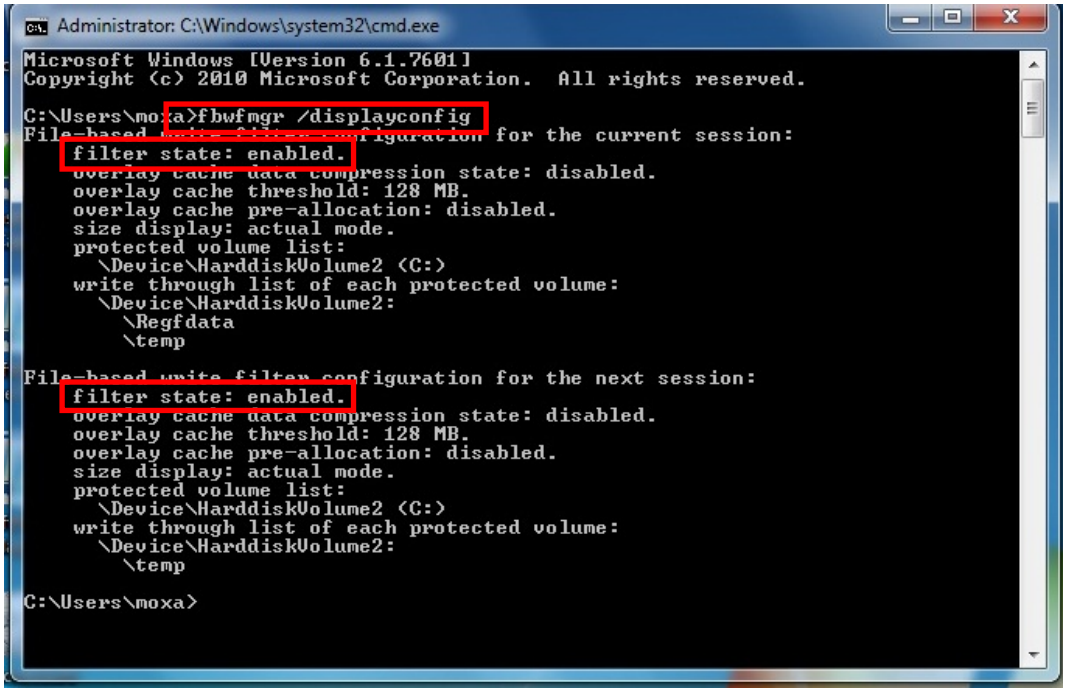
1. Check the current status of the Enhanced Write Filter using the **C:\...\>fbwfmgr /displayconfig** command.



2. Type **fbwfmgr /enable** at the command prompt to enable the FBWF. Reboot the system for the changes to take effect.



3. Run the **C:\...\>fbwfmgr /displayconfig** command to confirm that the Enhanced Write Filter is enabled and has been started on system boot up.



Additional details on FBWF configuration and usage is available on the Microsoft website at:

- FBWF overview
- FBWF Installation and Configuration
- FBWF features
- FBWF API

4

Customizable Sample Code

This chapter uses sample code to show how scripting can be used to add customized capabilities to the DA-682A-DPP computing platform.

The following topics are covered in this chapter:

- **Sample Code for Customizing the DA-682A-DPP**
  - LED Control
  - Watchdog Control
  - Relay Control

Sample Code for Customizing the DA-682A-DPP

The DA-682A-DPP comes with several pieces of sample code that you can use to customize the behavior of LED notifications, temperature, vibration, and GPS events, or to initiate emergency reboots when critical system applications come to a halt.

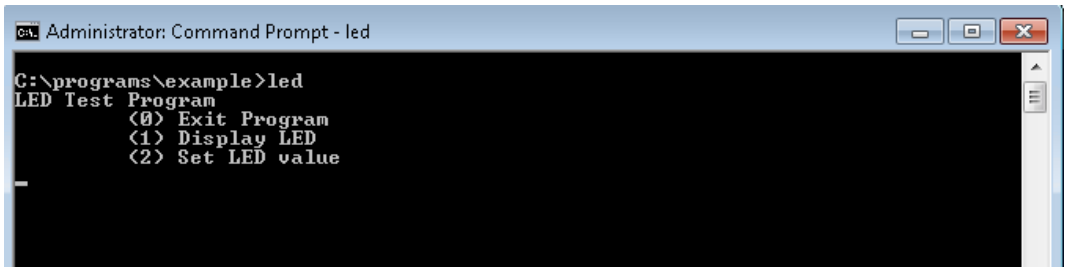
LED Control

The DA-682A-DPP is designed with 8 programmable LEDs that integrators and system administrators can customize for their notification needs. The source code for controlling the LED behavior is available on the software DVD under **\examples\Example\C++\WatchDog** and the compiled executable **LED.exe** is available at **\examples\Release**.

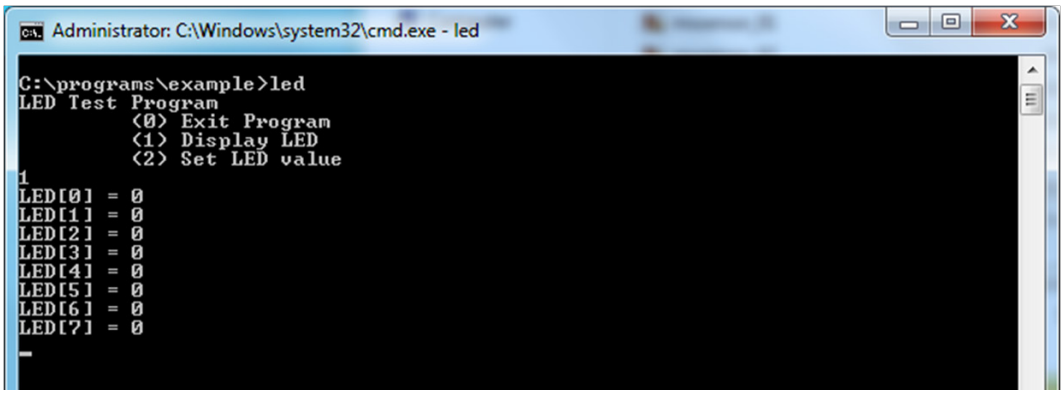
The LED Control Walkthrough

You can follow the steps below to test the LED control script or modify the control script to create customized patterns that are associated with specific system events. Refer to **LED Control Code** for the complete code.

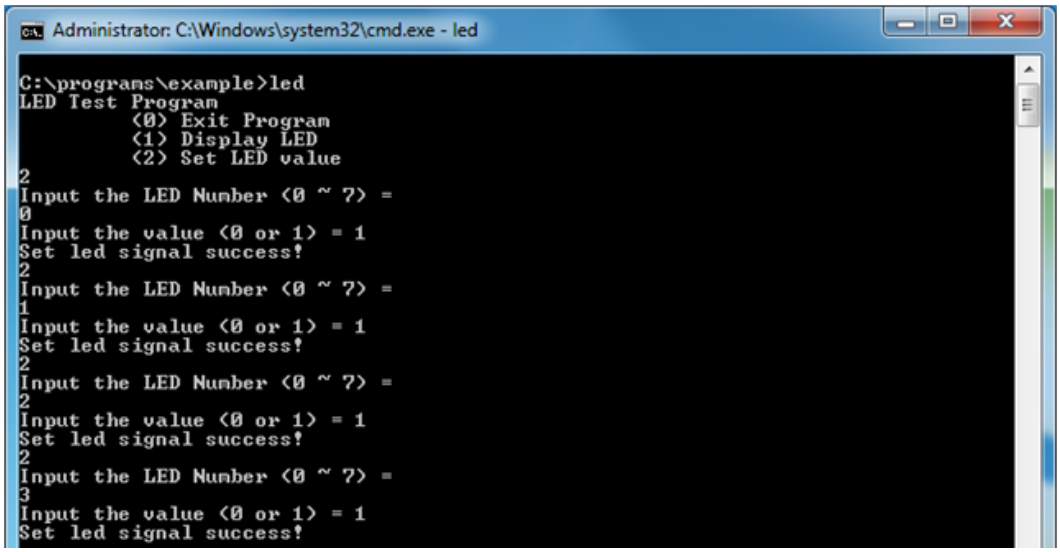
1. Create the **c:\programs\examples** folder on your computer and copy the **LED.exe** file from the software DVD into that folder. Run **LED.exe** on the computer.
2. Select **1** to display the LED's current status.



In the screenshot below, the user has selected **1** and is viewing the current status of all LEDs. The value **0** shown next to each LED entry indicates that all of the LEDs are currently turned off. Visually verify this by examining the LEDs on the DA-682A-DPP's front panel.



3. Run the **LED.exe** program again, but this time, select the option **2** to set the LED value. At the next prompt, enter the ID number (0 to 7) of the programmable LED that you want to activate, and when prompted enter either **1** (to turn it on), or **0** (to turn it off).



Watchdog Control

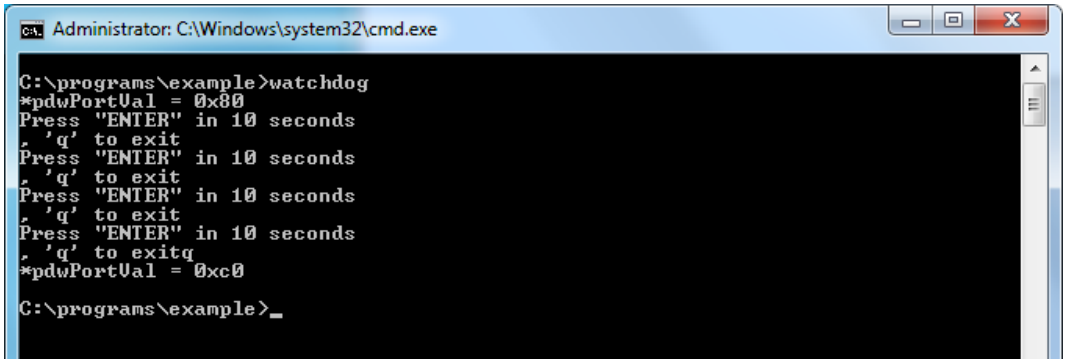
The code for controlling the watchdog/COP timer is the simplest and least-customizable of the sample scripts. The source code is available on the software DVD under **\examples\Example\C++\WatchDog** and the compiled executable **Watchdog.exe** is available at **\examples\Release**.

You can modify this code to integrate the watchdog timer with specific applications. Using this code you can set up programs where the watchdog timer will provide a last-line failsafe mechanism during application crashes. For instance, the DA-682A-DPP may be set up so that whenever a mission-critical application fails the watchdog timer will send a message to a system administrator and then initiate an automatic reboot.

The Watchdog Control Code Walkthrough

To test the watchdog executable, do the following:

1. Create the **c:\programs\examples** folder on your computer (if you have not already done so) and copy the **Watchdog.exe** file from the software DVD into that folder. Run **Watchdog.exe** on the computer.
2. To keep the system from rebooting automatically, you need to press **Enter** at least once every 10 seconds.
3. To stop the watchdog and exit the program, press **q**.



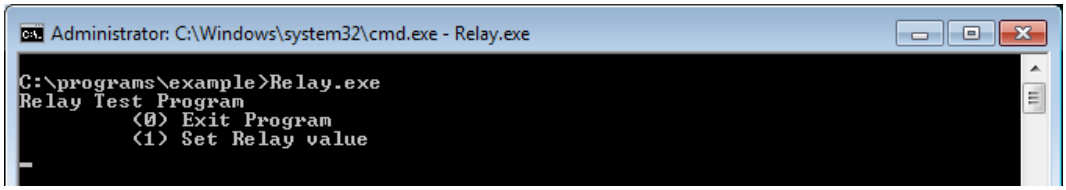
Relay Control

The DA-682A-DPP is provided with one programmable relay that integrators and system administrators can customize for their notification needs. The source code is available on the software DVD under **\examples\Example\C++\WatchDog** and the compiled executable **relay.exe** is available at **\examples\Release**.

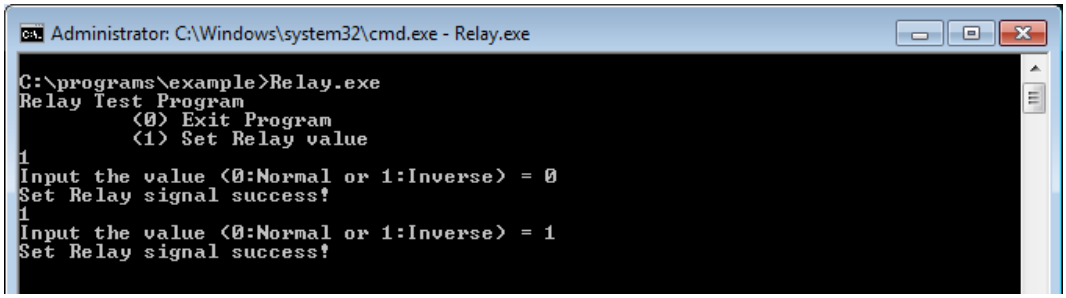
The Relay Control Walkthrough

You can follow the steps below to test the relay control script, or you can modify the control script to create customized patterns that are associated with specific system events. Refer to **Relay Control Code** for the complete code.

1. Create the **c:\programs\examples** folder and copy the **relay.exe** file into that folder.
2. Run **relay.exe**.
3. In the **Relay Test Program** menu, select **1** to set the status of the relay.



In the example below, the user has selected **1 (Set Relay value)** and is setting the current status of the relay. The value **0** indicates that the relay is currently in normal status as per the jumper setting. The value **1** shown next to the relay entry indicates that the relay is currently in inverse status.



The following table shows the relay status according to the jumper setting in the computer:

Jumper Setting	NO	NC
System Boot up	Close	Open
Set value 0	Open	Close
Set value 1	Close	Open

5

System Recovery

The DA-682A-DPP ready-to-run embedded computers are Windows Embedded Standard 7 platforms. This chapter describes the recovery process for these computers in the event of a system crash.

The following topics are covered in this chapter:

Overview: Setting Up the Recovery Environment

You will require a DA-682A-DPP computer, a 4-GB (minimum) USB drive, and a copy of the recovery suite to set up the DA-682A-DPP's system recovery environment.

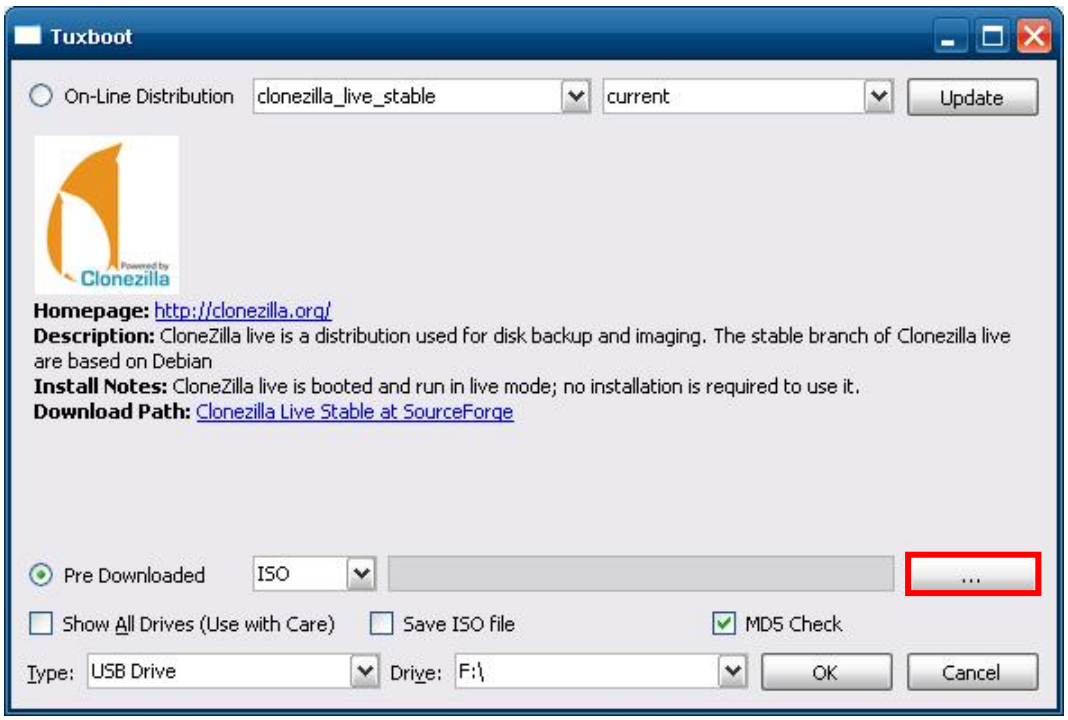
The recovery procedure itself requires only a DA-682A-DPP computer and a bootable USB drive.

The following steps describe the basic process of setting up the system recovery environment:

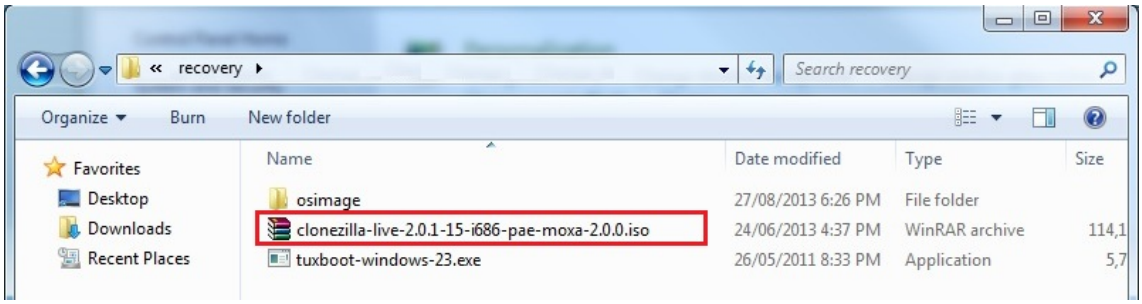
1. **Preparing the USB drive**: Involves copying the bootable recovery environment, which comes in the form of an ISO (*.iso) image, from the software CD to the USB drive.
2. **(optional) Restoring the System to the Stock OS**: Here, you can choose to create a bare-bones stock OS recovery image. If you choose this option as your recovery method, keep in mind that any applications or scripts you install later cannot be recovered during the system restore process.
3. **Setting up the BIOS to Boot via the USB**: Here, you will reset the BIOS so that the USB port is the first boot priority. **If you are initiating a recovery from a key you have already configured, this will be your starting point.** The system will re-booted into the CloneZilla recovery environment found on the USB
4. **(optional): Creating a Custom System Image**: This step describes how to create an exact copy of a fully configured system on the USB drive. This is the alternative to the stock OS recovery offered in Step 2.
5. **Performing a Trial Recovery**: This step describes how to perform a recovery; you can use this to run a trial system recovery and test your setup.
6. **Resetting the BIOS to its Original State:** This step explains how and why to return the BIOS to its original state.

Step 1: Preparing the USB drive

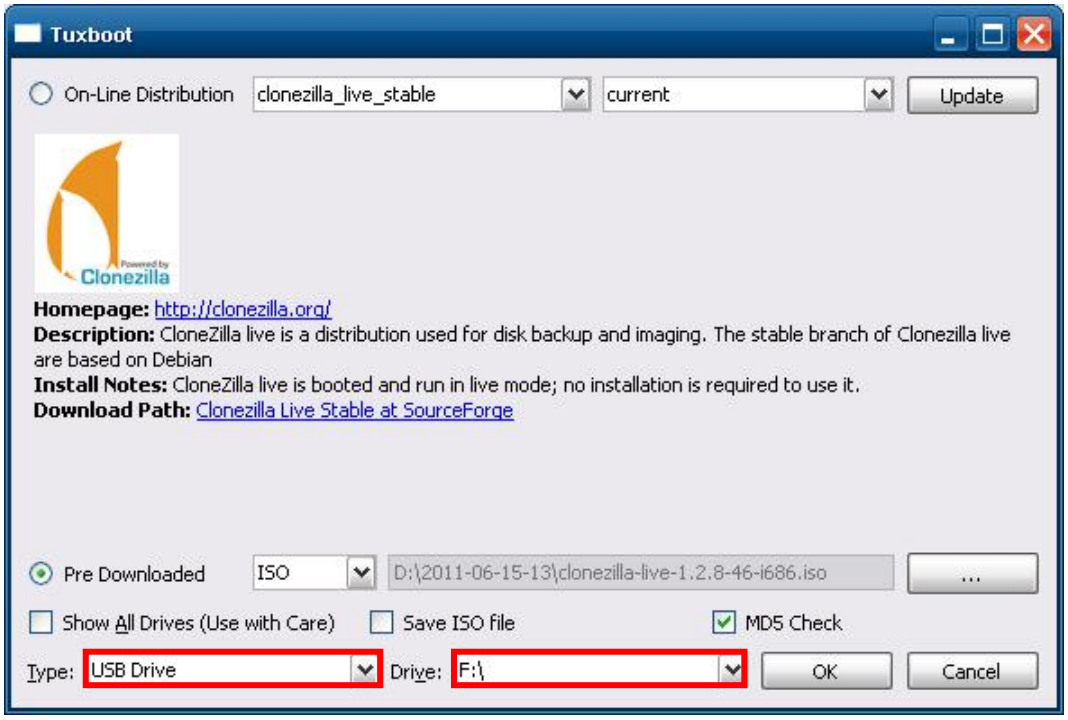
1. From the software DVD that came with your computer **start the Clonezilla imaging program** (within the current OS) by starting `tuxboot-windows-23.exe`, which is found in the `\recovery\` folder.
2. At the right, select **Pre-Downloaded** and set the dropdown to **ISO**.
3. **Browse the CD to locate the Clonezilla ISO image** by clicking the button with an ellipsis (**...**).



4. In Windows Explorer, navigate to the `\recovery\` folder on the software DVD and select the ISO image for the CloneZilla recovery environment.



5. Set the **Device Type** (lower left-hand corner) as **USB Drive**, then set the **Drive** dialog to the letter under which the USB is currently mounted.



6. Click **OK**, and the CloneZilla recovery environment (plus bootloader) will be copied to your USB drive.

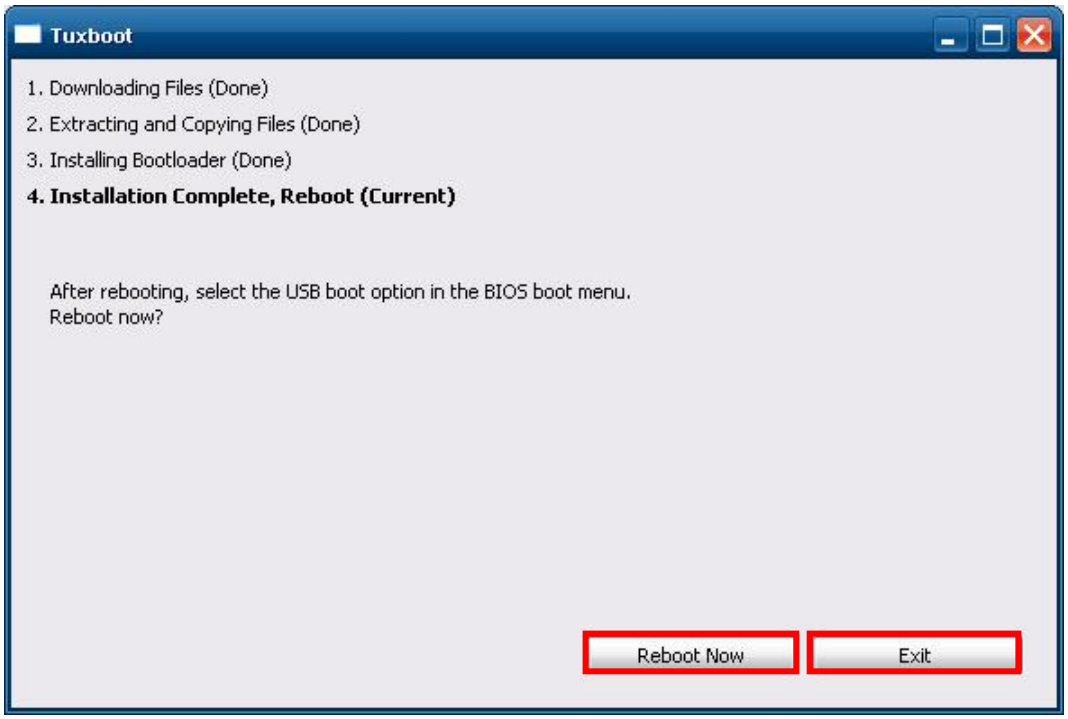
The Two Types of Recovery: Base Install and Fully Configured

Because of the naming conventions used, for any given computer only a single system image may be stored on any given USB drive. Consequently, at this point, users need to make a decision about which sort of system recovery is preferred:

A. A recovery image of **a fully configured OS**, with user-installed software applications and scripts, or
B. A recovery image of only the **basic, newly-installed root OS.**

A: To configure the recovery environment to copy over a fully configured system, users should click **Reboot Now** to close the installation environment and restart the computer. They should then proceed to the next section, **Step 3: Setting Up the BIOS to Boot Via** the USB and continue the installation of the recovery environment by continuing to **Step 4 (optional): Creating a Custom System Image**.

B: To configure the recovery environment to boot into a clean OS image with no applications, users should instead click **Exit** here to complete the installation and return to the OS. At this point, **Step 1** has been completed, and you should proceed to **Step 3: Setting Up the BIOS to Boot Via** the USB, and then go directly to **Step 4: Configuring a Stock OS** Restoration



Step 2 (optional): Restoring the System to the Stock OS

The instructions which follow describe how to set up the recovery environment that will ***restore the operating system to a pristine post-install state***. If you have installed any software on your system, then following these directions will result in ***all custom applications and code being wiped from the operating system***. If the computer has already been heavily customized with user applications and local scripts, then skip this section and instead go to the next section, **Error! Reference source not found.**. There, you will begin the process of copying over a full system image.

Creating a post-install rescue drive involves just two steps: preparing the USB drive, and then copying the rescue image (found on your Moxa software CD) to it.

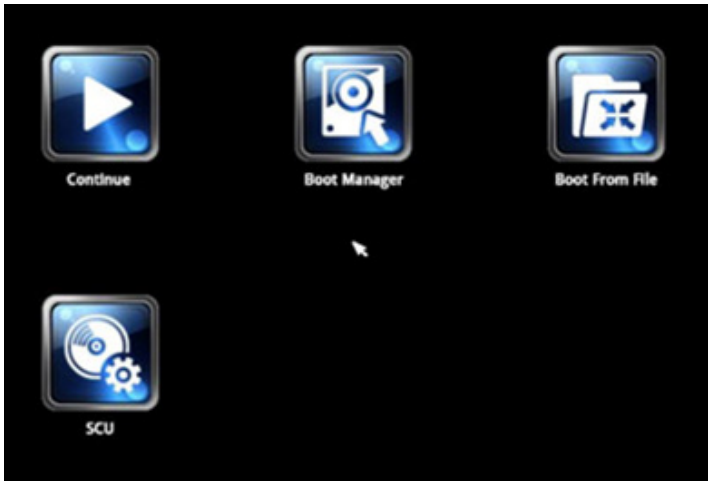
1. Prepare a USB drive.
2. Copy the stock OS image from the software DVD to the USB drive. The image is available in the **#:\<SoftwareDVD>\recovery\DA-682A-DPP** directory and should be copied to the partition image directory, **#:\home\partimag** on the USB drive.

   That's it. You have now configured a USB recovery key that will restore your computer to the stock operating system it was shipped with. You can test this by doing a trial recovery process on your computer. To do this, continue on to **Step 3: Setting Up the BIOS to Boot Via the** USB, skip step 4, and then go on to **Step 5: Performing a Trial Recovery.**

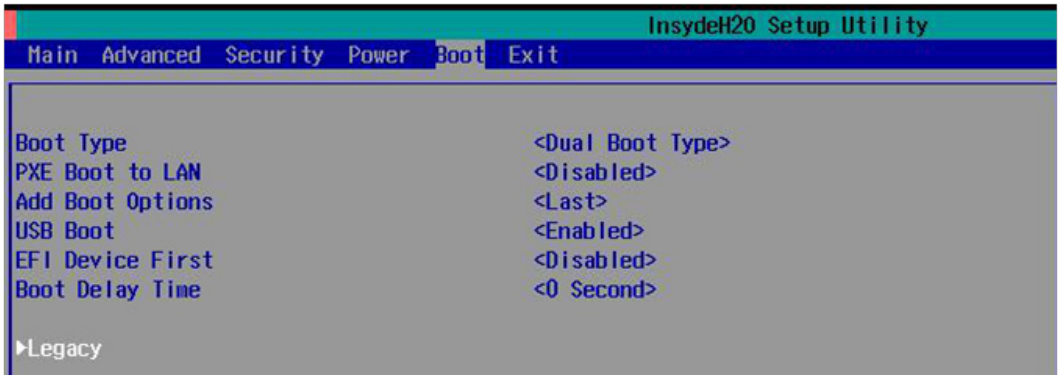
Step 3: Setting Up the BIOS to Boot Via the USB

At this stage, users will set the BIOS so that the system boots directly from the USB. This must be done before the rest of the system recovery environment may be configured

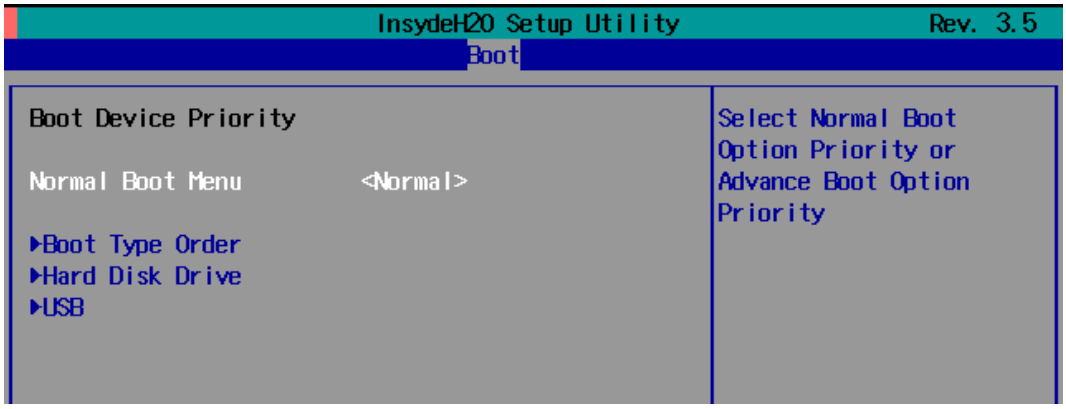
1. Turn on the computer and, during the POST process, press F2 until you hear a long beep. You should then enter the BIOS setup menu. Select **SCU** to enter the BIOS setup menu.



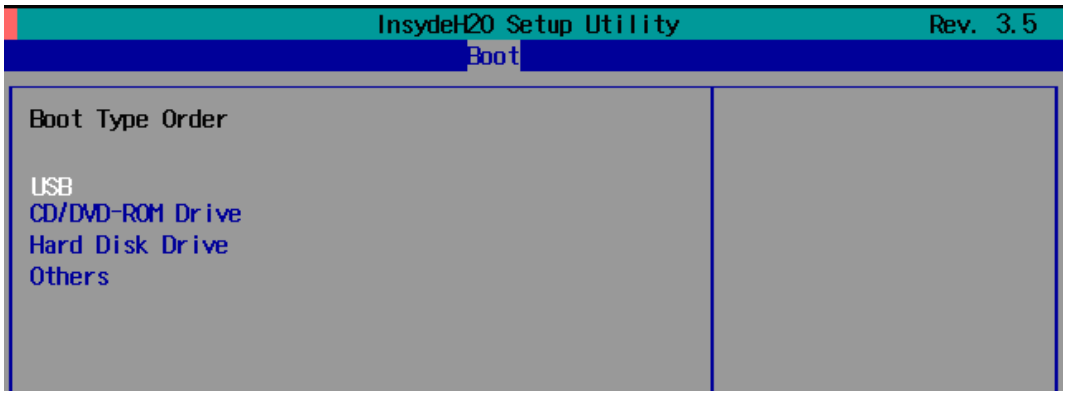
2. Use the right/left arrow keys to navigate to the **Boot** tab, use the up/down arrow keys to navigate to the **Legacy** option, and then press **Enter**.



3. Select **Boot Type Order** to open the dialog that will allow you to set the boot priority for the system drives.



4. Use the arrow keys to highlight **USB** and then press the plus key (**+**) to move it to the first position, or select the other options above it and use the dash key (**–**) to move them down.



5. Press F10 and then press **Enter** to save and exit the BIOS configuration interface. This should initiate the next reboot, during which your system should now boot from the USB drive.

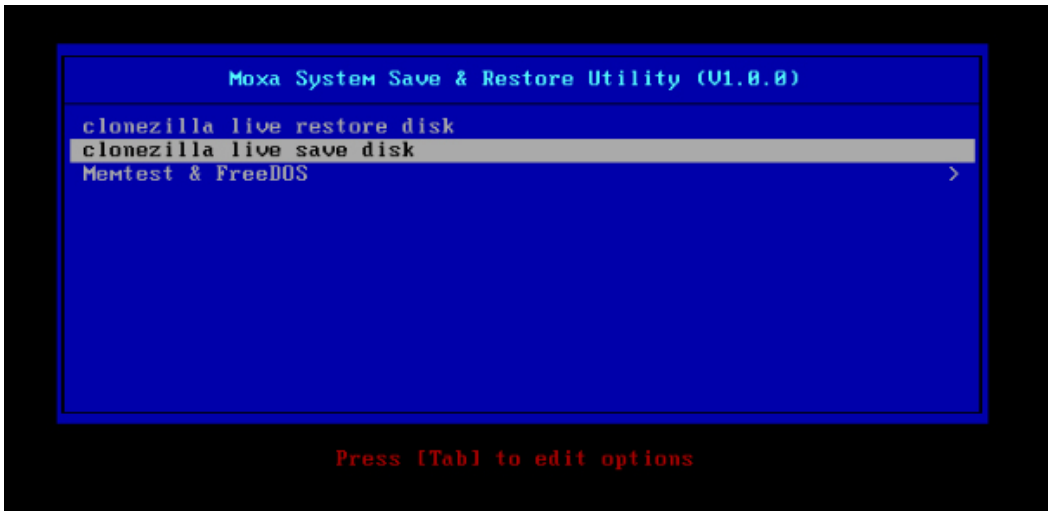
Step 4 (optional): Creating a Custom System Image

The instructions which follow are only to be used if you decided in **Step 1** of this process ***to create a full copy of an already-configured system***. If you have not yet installed any software on your system, then skip this section and instead go to **Step 4: Configuring a Stock OS** Restoration to copy over a clean OS image.

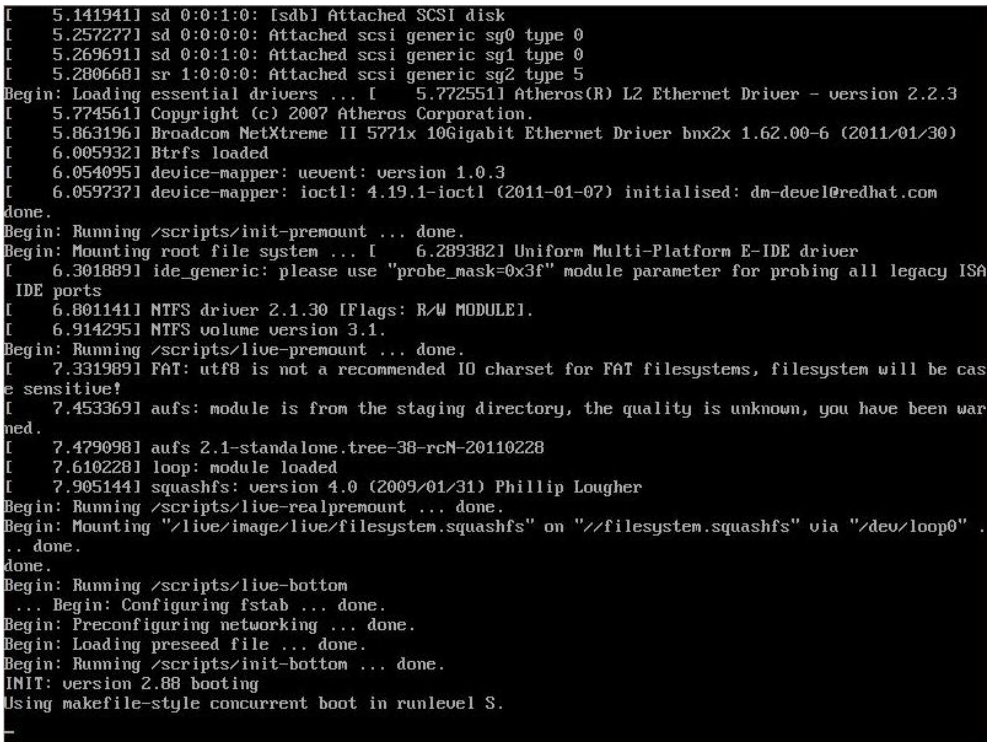
Using this procedure, you will save to the USB drive a copy of the entire system as it is currently configured to be used as a full system recovery image should the system crash. ***All files under*** *F:\home\partimag* ***will be overwritten***.

You should have already changed the BIOS settings to set the USB drive as the first boot priority. If you have not yet reset the boot priority, first return to Step 3: Setting Up the BIOS to Boot Via the USB, just above, and follow the directions there.

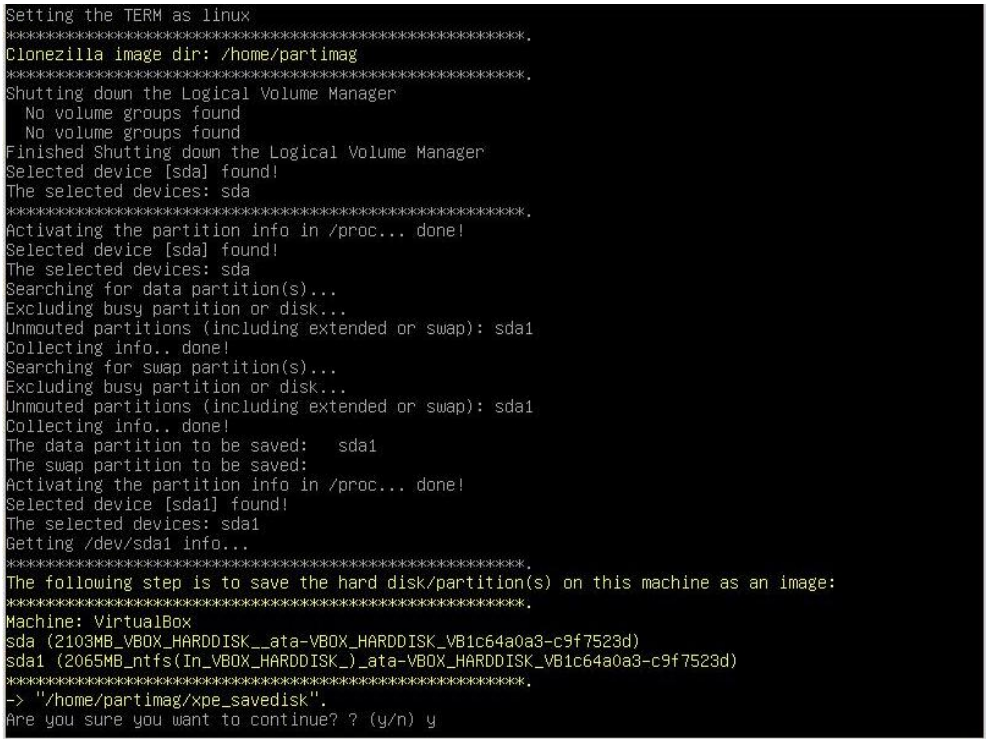
1. Once the system has launched and the DA-682A-DPP has booted the recovery environment from the USB drive, navigate to the entry **Clonezilla Live Save Disk**, and select it by pressing **Enter**. This will take you into the **recovery image creation environment**, allowing you to copy your full system setup to the USB drive.



2. The DA-682A-DPP will now boot into the image creation environment. Wait for the boot process to finish.



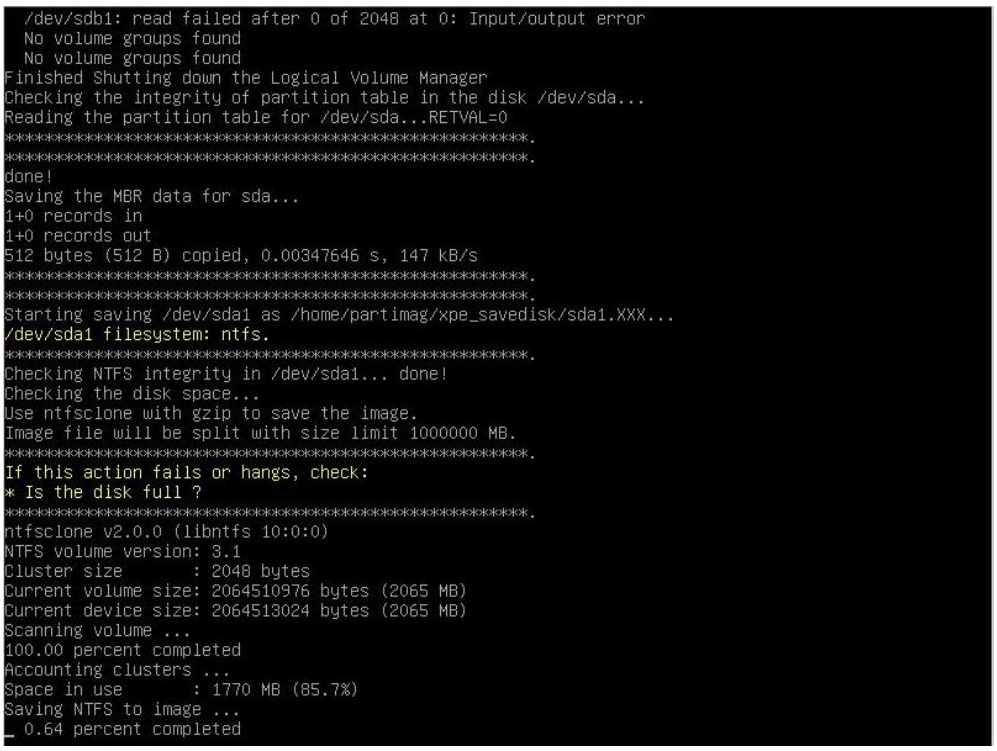
3. Once the image creation environment has completed booting up, you will be given a warning and asked if you wish to continue. Please keep in mind that if you create the recovery image, then ***any residual files currently copied to the*** **`/home/partimag`** ***directory will be deleted***. If there are any files remaining in the USB **partition image** directory and you wish to save them, you must exit the recovery environment and copy these files to another disk. If you wish to continue with the image creation, press **Y** (case insensitive) to continue (screenshot on the next page).



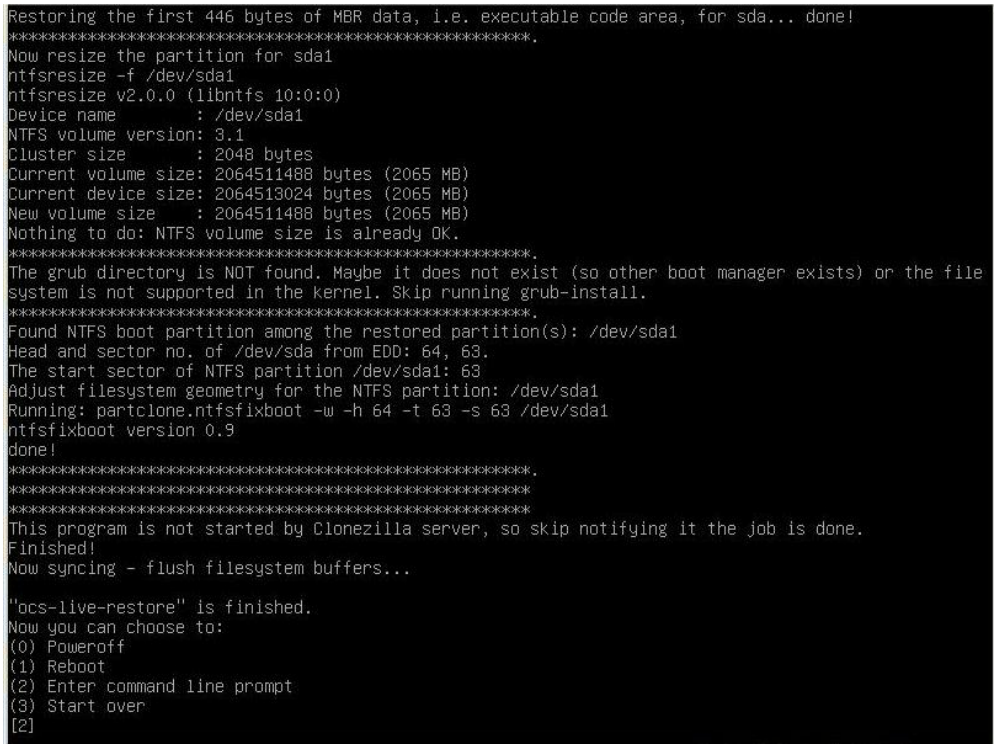
WARNING

The same filename is used for all recovery images, whether for the full system backup or for the clean OS image installation. This means that currently, it is impossible to have more than one system image per USB drive.

4. At this point, the recovery environment will copy of the entire hard drive to your USB drive. This will likely take several minutes, and perhaps as long as half an hour. Do not remove the USB drive during this time; wait patiently for the process to finish. Depending on the speed of your USB drive, this may be a good time to get a cup of coffee, or take a nap.



5. At this point you may choose to power down the computer (press **0**), reboot (press **1**), enter a console terminal (access a console TTY -- press **2**), or re-initiate the entire procedure (press **3**). ***Do not remove the USB drive until you have rebooted or powered down the system.***



6. Once you have powered down the system and removed the USB drive, you have finished configuring the recovery environment. The USB drive should be clearly labeled and stored in a safe place. You may now continue to the next section, where you will return the BIOS to its original state (Step 4) and test the recovery procedure for successful configuration (Step 5).

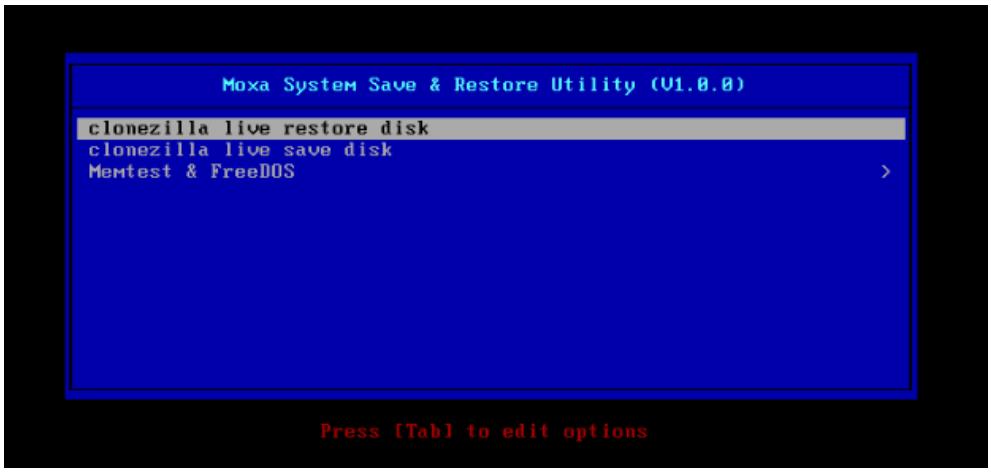
Step 4: Configuring a Stock OS Restoration

From within the desktop environment manually copy the directory containing the base OS from the software DVD over to the USB drive. To do this, copy **#:\<Software DVD>\recovery\os_image** to the partition image directory, **F:\home\partimag**

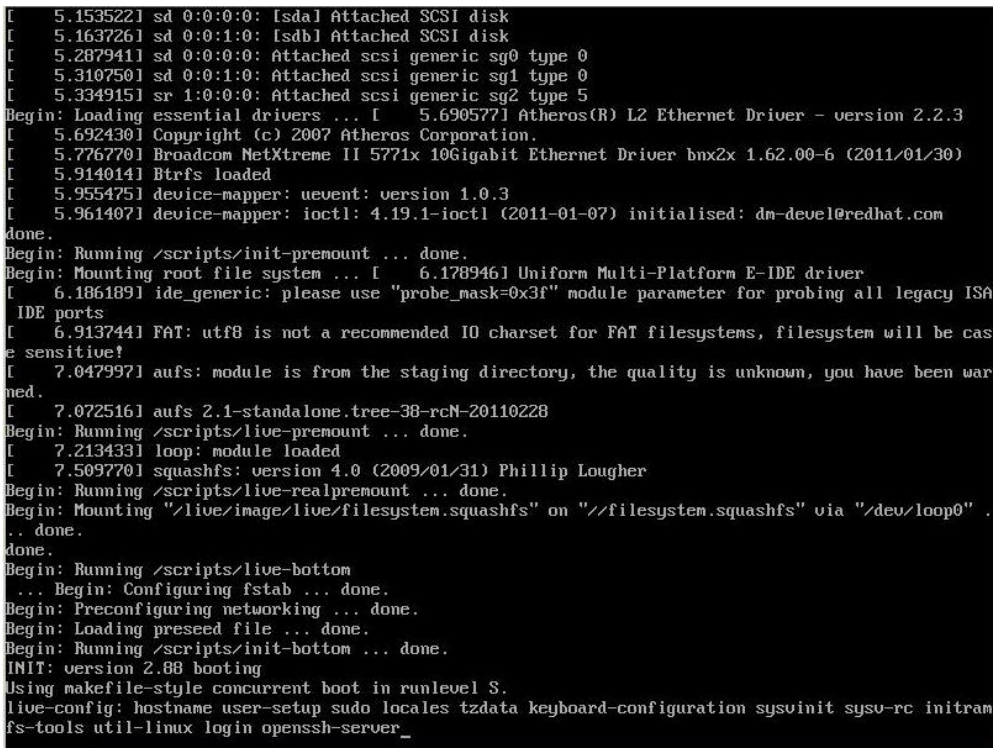
Step 5: Performing a Trial Recovery

Connect the USB drive to any of the DA-682A-DPP's USB ports and then reboot the computer. The system will boot from the USB into the Clonezilla boot loader.

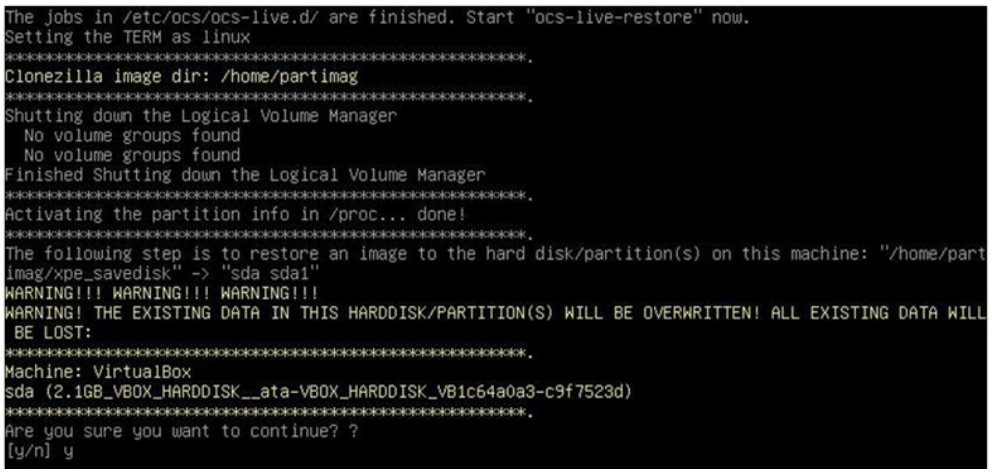
1. Select **Clonezilla Live Restore Disk** to boot into the system restoration environment.



2. Wait for the boot process to finish.

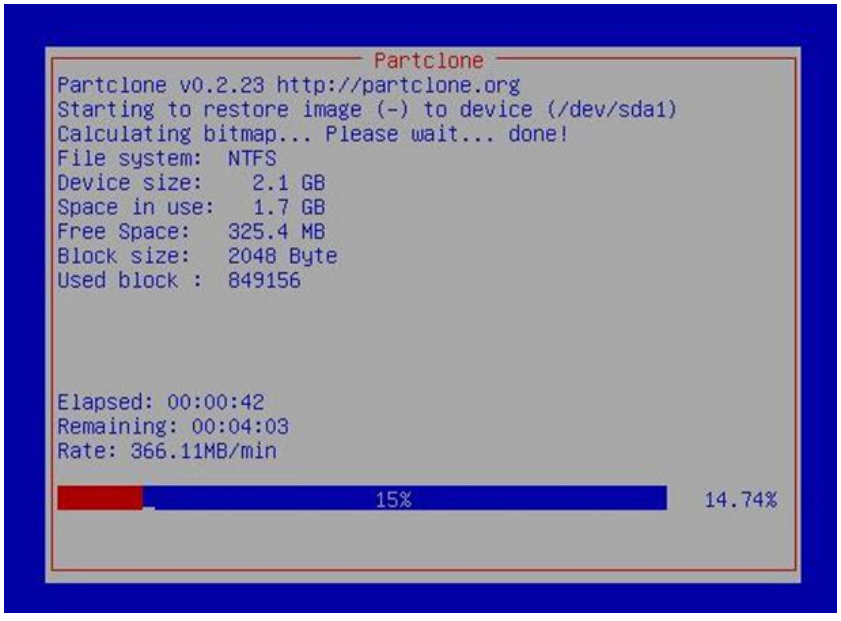


3. At this point, the system will remind you that you are about to overwrite your entire operating system with a new drive image, and ask you if you want to continue. When prompted, enter **Y** (case insensitive) from the keyboard to start the system restoration process. Any other letter or **Ctrl-C** will cancel it and exit Clonezilla.

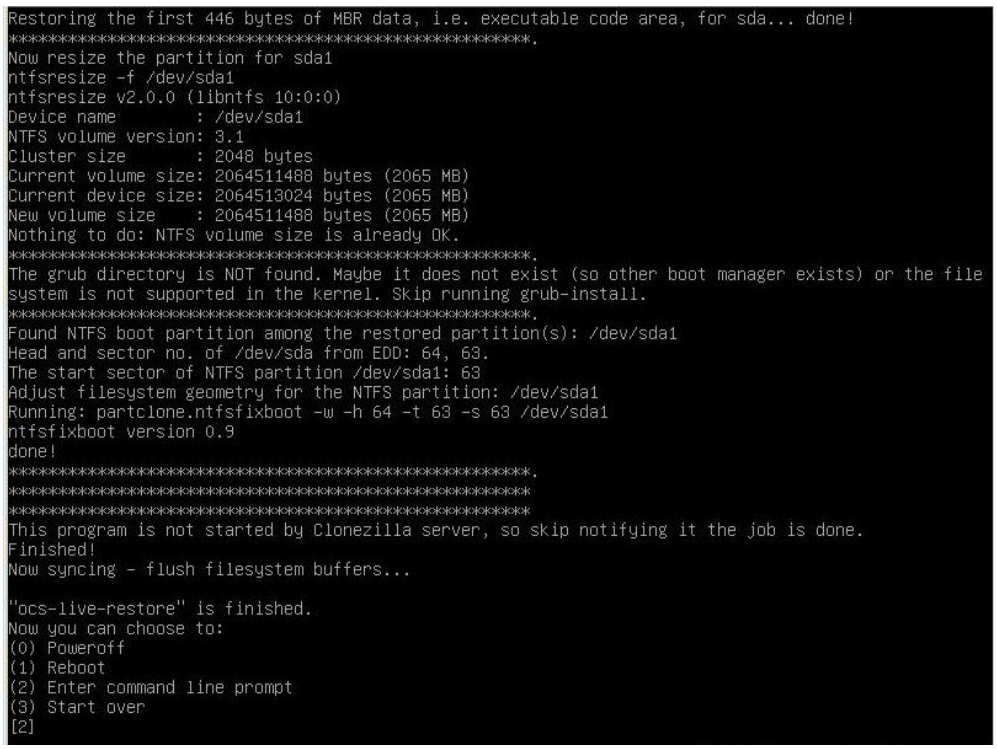


4. The system will give you another warning that you are about to overwrite your hard drive, and erase all data on the partition listed (**sda1**, in the example below). If you wish to continue, enter **Y** (case insensitive).

5. Now, Clonezilla will copy the system image you have configured on to your primary system drive. Your original system (and any stored data or configurations that were made after the recovery disk was created) will be entirely wiped clean. Wait for the process to finish; depending on the system, this should take about 10 minutes.



6. At this point, complete the restoration by selecting (**0**) **Poweroff**. This will shut down the computer; however, if the **Power Switch** remains inserted in the front panel of the computer and is left in the **ON** position, then the system will immediately initiate a soft reboot. To avoid this, users may use the switch to cut power to the computer immediately following the shutdown, or may simply remove the power switch from the front panel and then use the console to shut down the computer by pressing **0**.



7. After the computer has powered down, remove the USB drive and store it in a safe place.

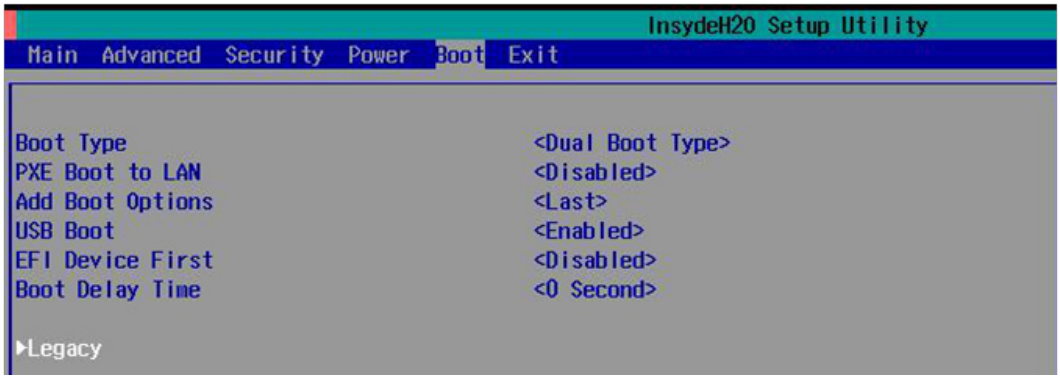
Step 6: Resetting the BIOS to its Original State

Now you will need to return the boot priority to its original configuration so that the system will boot from the original disk. This is done for two reasons; the first is security, so that the machine may not be rebooted from unauthorized USB drives

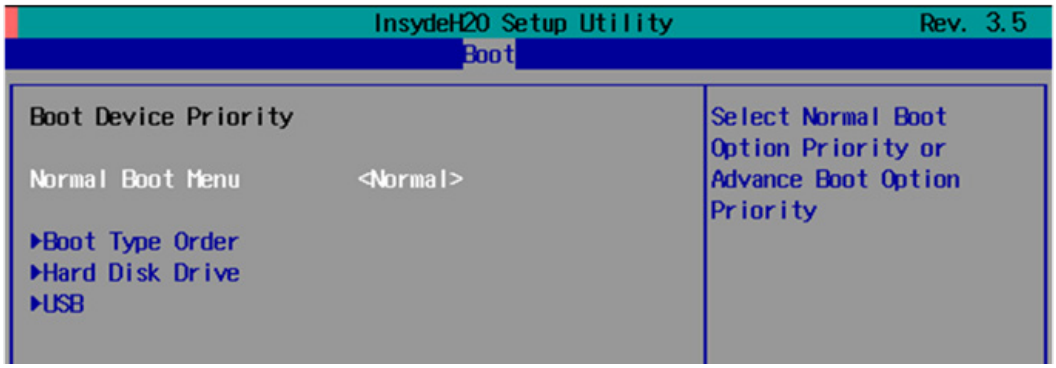
The second reason, however, is functional: currently, if the DA-682A-DPP is set to boot from the USB drive, then ***the DA-682A-DPP will hang any time a USB data drive (i.e.: non-bootable image) is inserted in the machine at boot time***. The DA-682A-DPP does not currently have the capacity to distinguish between simple USB data drives and boot-capable OS drives.

1. Reboot the system, and, during the POST process, press F2 until you hear a long beep. You should then enter the BIOS setup menu. Select SCU to enter the BIOS setup menu.

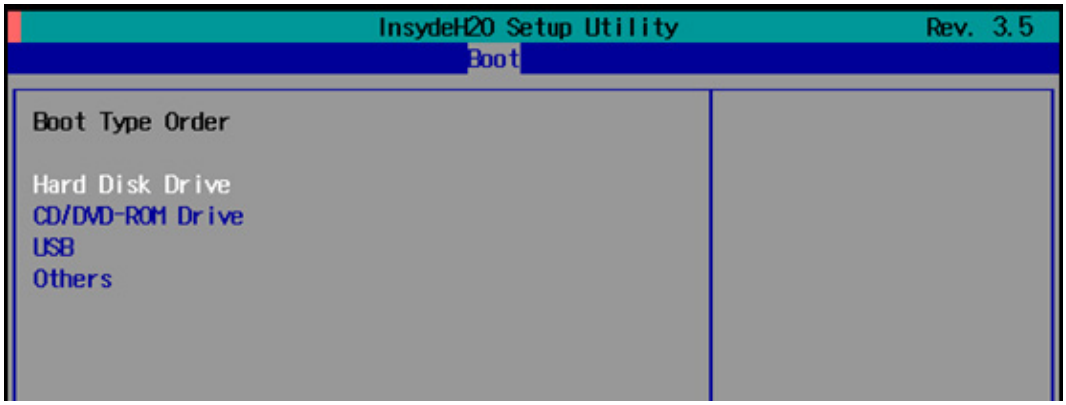
2. Use the left/right arrow keys to navigate to the **Boot** tab, and then press **Enter**.
3. Use the up/down arrows to highlight **Legacy** in the boot tab's menu, and press **Enter**.



4. Use the up/down arrow keys to navigate to the **Boot Type Order** link, and then press **Enter**.



5. Use the up/down arrows to highlight **Hard Disk Drive** and then use the plus/minus signs (**+ -**) to move it to the first boot priority position



6. Press F10 and then press **Enter** to save and exit the BIOS configuration interface. This should initiate the next reboot, and your system should now boot from the USB drive.

A

Sample Code for DA-682A-DPP Customization

This section provides you with a hard copy of the sample code included with the DA-682A-DPP. These short programs are intended to be used either as standalone scripts, or to be included in scripts created to build custom features for end users.

The following topics are covered in this appendix:

- **LED Control Code**
- **Watchdog Control Code**
- **Relay Control Code**

LED Control Code

The complete LED control code is listed here. You can use the code as is or modify the same to suit your requirement.

```
/*  Copyright (C) MOXA Inc. All rights reserved.
   This software is distributed under the terms of the
   MOXA License.  See the file COPYING-MOXA for details.
*/

#include "stdafx.h"
#include <windows.h>
#include "..\\Include\\mxdev.h"

#define LED_PORT_NUMBER 8

int _tmain(intargc, _TCHAR* argv[])
{
   HANDLE hLedDev;
intport_no;
int data;
intnLED = 0;
intnRet = 0;
int port = 0;
int mode = 0;
int n = 0;

   TCHAR sin;
   TCHAR smode;

   _tprintf(_T("LED Test Program\r\n"));
   _tprintf(_T("\t (0) Exit Program\r\n"));
   _tprintf(_T("\t (1) Display LED\r\n"));
   _tprintf(_T("\t (2) Set LED value\r\n"));

   sin = _gettchar();
   n = _tstoi(&sin);

   do
   {
      switch (n)
      {
         // if char == '1', display the LED output status
         case 1:
            // Open device
hLedDev = mxgpio_open();

            for (inti = 0; i< LED_PORT_NUMBER; i++)
            {
               // Get digital input
port_no = i;
nLED = mxgpio_get_data( hLedDev, port_no);
               _tprintf(_T("LED[%d] = %d\r\n"), port_no, nLED);
            }
```

```
            // Close device
mxgpio_close(hLedDev);
            break;

         // if char == '2', Set the LED output
         case 2:
            // Get Port Number
            _gettchar();
            _tprintf(_T("Input the Port Number (0 ~ %d) = \r\n"), LED_PORT_NUMBER-1);
smode = _gettchar();
port_no = _tstoi(&smode);

            // Get Value
            _gettchar();
            _tprintf(_T("Input the value (0 or 1) = "));
smode = _gettchar();
            data = _tstoi(&smode);

            // Open device
hLedDev = mxgpio_open();

               // Set LED
nRet = mxgpio_set_data( hLedDev, port_no, data);

            if ( nRet == -1 )
            {
               _tprintf(_T("Set led signal fail!\r\n"));
            }
            else
            {
               _tprintf(_T("Set led signal success!\r\n"));
            }

            // Close device
mxgpio_close(hLedDev);
            break;
      }
       _gettchar();
      sin = _gettchar();
      n = _tstoi(&sin);
   } while (n != 0);

   return 0;
}
```

Watchdog Control Code

```
/*  Copyright (C) MOXA Inc. All rights reserved.
    This software is distributed under the terms of the
    MOXA License.  See the file COPYING-MOXA for details.
*/

#include "stdafx.h"
#include <windows.h>
#include "..\\Include\\mxdev.h"

int _tmain(intargc, _TCHAR* argv[])
{
    PVOID fd; // Handle to device, obtain from mxwdg_open
    ULONG time;

    // starts watchdog timer
    time = 10;
fd = mxwdg_open(time);

    while ( TRUE )
    {
        _tprintf( _T("Press \"ENTER\" in 10 seconds\n, 'q' to exit"));
        TCHAR ch = _gettchar();
        if ( ch == (TCHAR)'q' )
        {
            break;
        }

        // refresh watchdog timer
mxwdg_refresh(fd);
    }

    // stops watchdog timer
mxwdg_close(fd);
    return 0;
}
```

Relay Control Code

```
#include "stdafx.h"
#include <Windows.h>
#include "..\\Include\\mxdev.h"

#define RELAY_OFFSET 0

int _tmain(int argc, _TCHAR* argv[])
{
     HANDLE hRelayDev;
    int port_no;
    int data;
    int nRelay = 0;
```

```
    int nRet = 0;
    int mode = 0;
    int n = 0;

    TCHAR sin;
    TCHAR smode;

    _tprintf(_T("Relay Test Program\r\n"));
    _tprintf(_T("\t (0) Exit Program\r\n"));
    _tprintf(_T("\t (1) Set Relay value\r\n"));

    sin = _gettchar();
    n = _tstoi(&sin);

    do
    {
        switch (n)
        {
            // if char == '1', Set the Relay output
            case 1:
                // Get Value
                _gettchar();
                _tprintf(_T("Input the value (0:Normal or 1:Inverse) = "));
                smode = _gettchar();
                data = _tstoi(&smode);

                // Open device
                hRelayDev = mxrelay_open();

                  // Set Relay
                nRet = mxrelay_set_data( hRelayDev, RELAY_OFFSET, data);

                if ( nRet == -1 )
                {
                    _tprintf(_T("Set Relay signal fail!\r\n"));
                }
                else
                {
                    _tprintf(_T("Set Relay signal success!\r\n"));
                }

                // Close device
                mxrelay_close(hRelayDev);
                break;
        }
          _gettchar();
        sin = _gettchar();
        n = _tstoi(&sin);
    } while (n != 0);

    return 0;
}
```