Copy-Compare Sample Code

Reference Manual

ATP Electronics Inc.

Jaron Cheng

Software Team, the 3rd Engineering Department

Table of Contents

I. Introduction

The major feature of Copy-Compare sample code is to validate data accuracy while accessing storage devices. The basic action modes are writing, reading and comparing data, different purpose of tests can be processed with specific combination of options.

The contents of this sample code are divided into three parts; they are the portion of interface, logic and data processing. The major logic and data processing function have been wrapped as a dynamic linked library named "DLL_CopyCompare.dll". The main program of this sample code only tackles with interface-related functions, software developers can manipulate logic and data processing function through inputs and outputs of the dynamic linked library.

In addition, there are other dynamic linked libraries, such as JudgeOS.dll is for operation system judgment, and GetCapacity.dll is for acquiring logical capacity of devices.

The text will explain all functions in the main program and input/output of every dynamic linked library.

II. Functions in the Main Program

1、Interface

Functions related to interface are included in the class of CCopyCompareDlg：

(1) OnBnClickedAddPattern

To add Patterns，grab text in the pull-down list "IDC_COMBO2" and add it to the list control "m_List".

(2) OnBnClickedClearAll

Clear contents in the list control "m_List".

(3) OnBnClickedCopyCompare

To load device information from the class "CFlashDevice" , initialize deviceand invoke the core function "deviceMain".

(4) OnBnClickedLoadIni

To load ini file.

(5) OnBnClickedPause

To halt process.

(6) OnBnClickedRadio1

To scan removable storage devices "REMOVABLE DISK" , if there is any then check corresponding "Checkbox".

(7) OnBnClickedRadio2

To scan fix storage devices " FIXED DISK" , if there is any then check corresponding "Checkbox".

(8) OnBnClickedRemove

To remove selected pattern in the list control.

(9) OnBnClickedSaveIni

To save ini file.

(10) OnBnClickedSelectFile

To select "Custom pattern" and add it to the list control.

(11) OnCbnSelchangeIdeCommand

To display default read/write speed while opt ATA/SCSI command.

(12) OnInitDialog

Preload necessary information when program is being executed.

(13) OnSelchangeCombo2

To disable "SelectFile" button(IDE_BUTTON8) when custom pattern is selected in the pull-down list "Combo2".

(14) OnTimer

To update the process of Copy-Compare every second.

(15) ScanUFD

To scan "Removable disk" or "Fixed disk" devices, and store necessary information in the class "CFlashDevice".

(16) setCheckBox

To check target devices to execute Copy-Compare。.

(17) showOK

To display OK after process finished.

(18) updateReport

To update information including processing result, percentage, speed, etc.

2、Data Processing

Data processing function is excluded from "CCopyCompareDlg", provides the function to deal with strings.

(1) int searchDriveIndex(char *chDriveName)

Return：Index.

Argument：device label "chDriveName".

Transform labels A://, B://,... etc to index 1, 2, ...etc.

(2) CString searchStringLength(CString ItemName, CString TotalString, int *nStart

Return：Description

Argument：Target name "ItemName", input string "TotalString", search index "nStart"

Search for the description of target name in the input strin： If the input string "TotalString" is 「DriveType=Removable」, target name "ItemName" is 「DriveType」, the default start position of the pointer "*nStart" is 0, then it returns " searchStringLength" is「Removable」and the end position of the pointer.

III. Input and Output of the Dynamic Linked Library

Files： DLL_CopyCore.dll JudgeOS.dll getCapacity.dll

1、Input

Data structure of "myDriveInform"

(1) bool fEnable：True- Enable device，False- Disable device

(2) char *chDriveName：Device name

(3) CString szDriveLabel：Device label

(4) CString szDriveFormat：Device format

(5) CString szCustomFile：Path of custom pattern file

(6) UINT nMode：Process mode

(7) UINT nDriveType：Device type

(8) unsigned __int64 i64StartSector：Start address of processing

(9) unsigned __int64 i64EndSector：End address of processing

(10) unsigned __int64 i64TotalSector：Total sector numbers of processing

(11) unsigned __int64 AccessNumber：Total access sector number in random mode

(12) unsigned __int64 SubStep：Sector number of step size

(13) int nDriveIndex：Device index

(14) int nCopySize：Copy size of each access

(15) int nEditNumber：Edit number of start address of each device

(16) int nLoopNumber：Loop number

(17) int nLoopCount：Loop count

(18) int nPercentage：Percentage of process

(19) int nPatternCount：Pattern count

(20) PatternData myPatternData：Pattern information

① int nPatternMode：Pattern mode，such as fix pattern, random pattern… etc

② int nPattern：Content of the pattern, such as 0xAA, 0xFF… etc

③ CString szCustomFile：file path of custom pattern

(21) unsigned char* pWriteMemory：Pointer to write buffer

(22) unsigned char* pReadMemory：Pointer to read buffer

(23) HANDLE DeviceHandle：Device handle

2、Output

(1) int g_nSectionCount：Pattern count

(2) DriveControl g_DriveControl：Information related to device control

① UINT Mode：Process mode

② RetryCount：Retry count

③ ULONGLONG BufferStart：Buffer start of custom file

④ ULONGLONG BufferEnd：Buffer end of custom file

⑤ Option： Test option in one byte

- Option.BIT.Enable：1- Device enabled，0- Device disabled
- Option.BIT.ErrorStop：1- Halt if error occurs，0- Error continues
- Option.BIT.HideWarning：1- Hide warning message，0- Show warning message
- Option.BIT.ATACommand：1- ATA command，0- SCSI command
- Option.BIT.Beep：1- Beep，0- No beep
- Option.BIT.ReadEachWrite：1- Read each write，0- Read after write
- Option.BIT.Sequential：1- Sequential access，0- Random access
- Option.BIT.FormatAfterward：1- Format after process done，0- No formatting

⑥ Option2： Test option 2 in one bytes

- Option2.BIT.Removable：1- Removable device，0- Fix device
- Option2.BIT.WriteVerify：1- Write verify，0- No write verify
- Option2.BIT.ListControl：1- Use pattern in the list，0- Do not use pattern in the list

⑦ Status：Process status in one byte

- Status.BIT.Processing：1- Processing，0- Halt
- Status.BIT.ErrorOccurred： 1- Error occurred，0- Normal
- Status.BIT.Reading： 1- Reading device，0- Others
- Status.BIT.ForceStop： 1- Force to stop， 0- Normal
- Status.BIT.Looping： 1- Loop transits， 0- Normal
- Status.BIT.Patterning：1- Pattern transits， 0- Normal
- Status.BIT.Formating： 1- Formatting， 0- Normal
- Status.BIT.LogicalWrite： 1- Logical mode， 0- Physical mode

(2) int g_nLoopCount：Loop count

(3) unsigned __int64 g_lStartSector：Start sector number

(4) unsigned __int64 g_lEndSector：End sector number

(5) unsigned __int64 g_lSectorSum：Processed sector numbers in sequential mode

(6) unsigned __int64 g_lAccessCount：Processed sector number in random access mode

(7) unsigned __int64 g_lSubSector：Sector numbers of step size

(8) unsigned __int64 g_lSumWrite：Written sector numbers

(9) unsigned __int64 g_lSumRead：Read sector numbers

(10) int g_nTempCopySize：Buffered copy size in byte

(11) int g_nTempSubStep：Buffered step size in sector

(12) ErrorCode g_ErrorCode：Returned error code

The format of error code is combined with three bytes, the first byte represents process mode, the second one is process status, and the last one error code①.

(15) int g_nRetryCount：Retry count

(16) bool g_pauseStart：True- Halt process，false- Start processing

(17) double g_dbStartTime：Access start time in millisecond

(18) double g_dbEndTime：Access end time in millisecond

(19) double g_dbWriteLength：Total written length in sector

(20) double g_dbWriteTime：Total written time in millisecond

(21) double g_dbReadLength：Total read length in sector

(22) double g_dbReadTime：Total read time in millisecond

(23) int g_nPercentage：Percentage of process ing

(24) int g_nPercentageWrite：Writing percentage

(25) int g_nPercentageRead：Reading percentage

(26) double g_dbProcessLength：Total accessed sector numbers in random access mode

(27) CompareTime g_myTimer：Process time in second

(28) ULONGLONG g_dwErrorLogOffset：Offset in the list control of error log

(29) int g_nWritePhase：Current write phase

(30) int g_nReadPhase：Current read phase

(31) const WindowsVersion JudgeOS：Return operation system version②

① int OSIndex：Index of operation system

② DWORD MajorVersion：Major version

③ DWORD MinorVersion：Sub-version

(32) __int64 getDriveTotalSector：Return logic capacity of device in sector

IV. Annex

A1、Error Code

Code	Description
0x01	Busy
0x02	Error ID of status table
0x03	Failure read status table
0x04	Error write length
0x05	Write length cross border
0x06	Compare data error
0x07	Buffer overflow
0x08	Data pattern error
0x09	Failure open pattern file
0x0A	Error start address
0x0B	Error end address
0X0C	Error read length
0X0D	Process time out
0X0E	Invalid command
0X0F	Failure open handle
0X10	Failure read capacity

A2、Operation System Version

Product Name	Index	Main	Sub
Windows 98	1	4	10
WindowsMe	2	4	90
Windows NT 4.0 Windows 2000	3	4	0
Windows XP	4	5	1
Windows XP Professional x64 Edition Windows Server 2003 Windows Server 2003 R2	5	5	2
Windows Vista Windows 2008	6	6	0
Windows 7 Windows 2008 R2	7	6	1
Windows 8 Windows Sever 2012	8	6	2
Windows 8.1 Preview Windows Sever 2012 R2 Preview	9	6	3

V. Change History

Version	Contents
0.0.1	English version released