

Mirror Status Monitor Programming Guide

1. 概要

SMART コマンドおよびベンダ固有のSMART拡張コマンドを使用し、ステータスのモニタリングなどをおこなうことができます。ここでは、その手順を示しています。

2. 対象OS

Windows Server™ 2003、Windows® XP、Windows®2000、WindowsNT®4.0

※Microsoft, Windowsは、米国Microsoft Corporationの米国およびその他の国における商標または登録商標です。

※機種によって対応OSが異なりますのでご注意ください。

3. 注意事項

本内容に従って作成したアプリケーションは、管理者 (Administrator) 権限を持ったユーザーでログインしてから使用してください。

4. 事前準備

ビルドを行う場合にMicrosoft® Windows® ドライバ開発キット (DDK: Device Driver Kit) に含まれるファイル等が必要となります。

5. ステータスの取得

SMART CONTROL APIを用いてSMARTコマンドをミラー・デバイスに送ります。ミラー・デバイスからのSMARTステータスは拡張情報が付加されており、それによりミラー・デバイスのステータスを取得することができます。以下の手順にそってプログラムを記述してください。

5.1. はじめに

プログラミングで使用するWIN32 APIで使用する関数、構造体などに関しては、開発ツール(Visual Studioなど)に付属のSDKドキュメントなどを参考にしてください。

5.2. デバイスのオープン

アクセスするドライブごとにハンドルが必要です。

```
CHAR szString[30];  
HANDLE hSMARTCTRL;  
  
sprintf(szString, "¥¥¥¥.¥¥PhysicalDrive%d", drvnum);  
hSMARTCTRL = CreateFile(szString, GENERIC_READ|GENERIC_WRITE,  
                        FILE_SHARE_READ|FILE_SHARE_WRITE, NULL, OPEN_EXISTING, 0, NULL);
```

*drvnumは対象とするドライブの番号を指定する(0～)

5.3. Identify データの取得

Identifyデータより、モデル名、シリアルナンバーなどを取得することができます。必要がなければ飛ばしてください。

```
DWORD ByteReturned;
SEND_CMD_INPARAMS inpara;
BYTE IdentBuf[sizeof(SEND_CMD_OUTPARAMS) + 512 - 1];
//Identifyデータが入るバッファ

memset(&inpara, 0, sizeof(inpara));
memset(IdentBuf, 0, sizeof(IdentBuf));
inpara.cBufferSize = 512;           // xfer byte
inpara.bDriveNumber = drvnum;      // drive number 0-3
inpara.irDriveRegs.bDriveHeadReg = (drvnum & 1) ? 0xB0 : 0xA0;
inpara.irDriveRegs.bCommandReg = 0xEC;
ByteReturned = 0;

if(DeviceIoControl(hSMARTCTRL, SMART_RCV_DRIVE_DATA, &inpara,
sizeof(inpara)-1, IdentBuf, sizeof(IdentBuf), &ByteReturned, NULL)) {
    if(((SEND_CMD_OUTPARAMS*)IdentBuf)->DriverStatus.bDriverError) {
        エラー;
    }
    成功;
}
エラー;
```

*IdentBuf[sizeof(SEND_CMD_OUTPARAMS) - 1]から512byte分がIdentifyデータです

*drvnumは、0:Primary-Master, 1:Primary-Slave, 2:Secondary-Master, 3:Secondary-Slave

5.4. SMART のイネーブル

ミラー・デバイスのSMART機能を有効にします。電源投入後(またはソフトウェア起動時)は、下記のようにSMART Enable OperationsコマンドでSMART機能を有効にします。

```

DWORD ByteReturned;
SENDCMDINPARAMS inpara;
SENDCMDOUTPARAMS outpara;

memset(&inpara, 0, sizeof(inpara));
memset(&outpara, 0, sizeof(outpara));
inpara.cBufferSize = 0; // xfer byte
inpara.bDriveNumber = drvnum; // drive number 0-3
inpara.irDriveRegs.bFeaturesReg = 0xD8;
inpara.irDriveRegs.bCylLowReg = 0x4F;
inpara.irDriveRegs.bCylHighRe = 0xC2;
inpara.irDriveRegs.bDriveHeadReg = (drvnum & 1) ? 0xB0 : 0xA0;
inpara.irDriveRegs.bCommandReg = 0xB0;
ByteReturned = 0;

if(DeviceIoControl(hSMARTCTRL, SMART_SEND_DRIVE_COMMAND, &inpara,
sizeof(inpara)-1,&outpara, sizeof(outpara)-1, &ByteReturned, NULL)) {
    if(!outpara.DriverStatus.bDriverError)
        成功;
}
エラー;

```

*drvnumは,0:Primary-Master, 1:Primary-Slave, 2:Secondary-Master, 3:Secondary-Slave

5.5. ステータスの取得

SMART Return Status Commandを使って、ミラー・デバイスのステータスを取得します。その時点のステータスが取得できます。

Sector Countにセットされるステータスの詳細はインターフェース・スペックを参照ください。(注1)

```

DWORD ByteReturned;
SEND_CMD_INPARAMS inpara;
BYTE SMARTStatus[sizeof(SEND_CMD_OUTPARAMS) + sizeof(IDEREGS) - 1];
void *pTemp;

memset(&inpara, 0, sizeof(inpara));
memset(SMARTStatus, 0, sizeof(SMARTStatus));
inpara.cBufferSize = 0; // xfer byte
inpara.bDriveNumber = drvnum; // drive number 0-3
inpara.irDriveRegs.bFeaturesReg = 0xDA;
inpara.irDriveRegs.bCylLowReg = 0x4F;
inpara.irDriveRegs.bCylHighReg = 0xC2;
inpara.irDriveRegs.bDriveHeadReg = (drvnum & 1) ? 0xB0 : 0xA0;
inpara.irDriveRegs.bCommandReg = 0xB0;
ByteReturned = 0;

if(DeviceIoControl(hSMARTCTRL, SMART_SEND_DRIVE_COMMAND, &inpara,
sizeof(inpara)-1, SMARTStatus, sizeof(SMARTStatus), &ByteReturned, NULL)) {
    if(((SEND_CMD_OUTPARAMS*)SMARTStatus)->DriverStatus.bDriverError) {
        エラー;
    }
    pTemp = SMARTStatus + sizeof(SEND_CMD_OUTPARAMS) - 1;
    switch(((IDEREGS*)pTemp)->bSectorCountReg) {
        case 0: // ノーマルモード
            break;
        case 0x10: // ドライブ1フェイル
            break;
        case 0x11: // ドライブ2フェイル
            break;
        case 0x20: // ドライブ1へのデータ再構築中
            break;
        case 0x21: // ドライブ2へのデータ再構築中
            break;
        case 0x30:
            // 再構築は終了したが、ドライブ1の一部に読めないセクタを検出しました。
            // このセクタを使ったファイルがあった場合には内容が失われる可能性があるので、
            // Windowsに添付のツール(Scandiskなど)にて該当箇所の確認とディスクの復旧を
            // 行ってください。
            break;
        case 0x31:
            // 再構築は終了したが、ドライブ2一部に読めないセクタを検出しました。
            // このセクタを使ったファイルがあった場合には内容が失われる可能性があるので、
            // Windowsに添付のツール(Scandiskなど)にて該当箇所の確認とディスクの復旧を
            // 行ってください。
            break;
    }
}
エラー;

```

*drvnumは、0:Primary-Master, 1:Primary-Slave, 2:Secondary-Master, 3:Secondary-Slave

(注1)

2004 年 8 月以降に発売開始された機種でミラーリング機能搭載モデルの場合、下記の内容がステータス情報として得られます。

0x00 : ノーマル
 0x10 : ドライブ1フェイル
 0x11 : ドライブ2フェイル
 0x20 : ドライブ1へのデータ再構築中
 0x21 : ドライブ2へのデータ再構築中
 0x30 : ドライブ1の再構築完了したが、読めないセクタを検出
 0x31 : ドライブ2の再構築完了したが、読めないセクタを検出

※FC98-NX シリーズに関する情報は以下のホームページをご覧ください。

URL:<http://www.nec.co.jp/fc/>

5.6. データ再構築時のプログレス取得

データ再構築中にそのデータ再構築のプログレスを取得することができます。

プログレスの取得にはSMART Return Status Commandを拡張したコマンドを使用しており、コマンドに対する応答としてSector Count Registerにセットされた値がプログレスとなります。

```

DWORD ByteReturned;
SENCMDINPARAMS inpara;
BYTE SMARTStatus[sizeof(SENCMDOUTPARAMS) + sizeof(IDEREGS) - 1];

memset(&inpara, 0, sizeof(inpara));
memset(SMARTStatus, 0, sizeof(SMARTStatus));
inpara.cBufferSize = 0; // xfer byte
inpara.bDriveNumber = drvnum; // drive number 0-3
inpara.irDriveRegs.bFeaturesReg = RETURN_SMART_STATUS; // DAh
inpara.irDriveRegs.bSectorCountReg = 0x2F;
inpara.irDriveRegs.bSectorNumberReg = 0x00;
inpara.irDriveRegs.bCylLowReg = SMART_CYL_LOW; // 4Fh
inpara.irDriveRegs.bCylHighReg = SMART_CYL_HI; // C2h
inpara.irDriveRegs.bDriveHeadReg = (drvnum & 1) ? 0xB0 : 0xA0;
inpara.irDriveRegs.bCommandReg = SMART_CMD; // B0h
ByteReturned = 0;

if(DeviceIoControl(hSMARTCTRL, SMART_SEND_DRIVE_COMMAND, &inpara,
sizeof(inpara)-1, SMARTStatus, sizeof(SMARTStatus), &ByteReturned, NULL)) {
    if(((SENCMDOUTPARAMS*)SMARTStatus)->DriverStatus.bDriverError) {
        エラー;
    }
    pTemp = SMARTStatus + sizeof(SENCMDOUTPARAMS) - 1;
    // pTempがIDEREGS構造体へのポインタになります。
    // (((IDEREGS*)pTemp)->bSectorCountReg)
    // Sector Count Registerの値がデータ再構築のプログレスとなります。
}
エラー;

```

*drvnumは、0:Primary-Master, 1:Primary-Slave, 2:Secondary-Master, 3:Secondary-Slave