RS1394 CCD SDK

1394 CCD CAMERA Software Development Kit

User's Manual

第1.1版

RATOC Systems, Inc.

ラトックシステム株式会社

目　次

第一章 はじめに

1-1. はじめに

この度は RS1394CCD SDK(Software Development Kits)をお買い求め頂き誠にありがとうございました。本製品をご使用になる前に必ず別紙の RS1394CCD SDK 使用許諾書をお読み頂き、ユーザ登録申込書に必要事項をご記入頂き、捺印の上、弊社宛にご返送下さるようお願い申し上げます。

本 SDK は Microsoft DirectX 等に関する高度な知識を必要とすることなく、Visual C/C++,Visual Basic に関するプログラミング知識さえあれば、1394 CCD カメラからの非圧縮ストリームのキャプチャおよび再生を行うアプリケーションの開発を可能とするソフトウェア開発ライブラリキットです。

CCD カメラドライバに関する情報は、本製品添付の RS 1394CCD カメラドライバユーザーズマニュアルを参照願います。

製品特徴

RS1394CCD SDK は、

- Microsoft Visual C/C++ 6.0 対応 COM ライブラリ（インプロセスサーバー形式 DLL で提供）
- Microsoft Visual BASIC 6.0 対応 ActiveX ライブラリ
- Visual C/C++・Visual BASIC の各種サンプルプログラム

により構成され、Visual C/C++・Visual BASIC から下記のアプリケーションの作成を可能とします。

- 1394 CCD カメラからの非圧縮ストリームの静止画スナップショット
  24 ビットビットマップ形式静止画保存。
- 1394 CCD カメラからの非圧縮ストリームの動画再生およびキャプチャ
  非圧縮および各種圧縮形式の動画保存。
- アプリケーションからの CCD カメラフルコントロール
  IIDC 1394-based Digital Camera Specification Version 1.30 準拠
- 外部トリガーによる静止画スナップショット
  CCD カメラへの外部トリガー信号入力により 1 フレーム画像を取得。外部トリガーの発生イベントをアプリケーションで検出し画像データをビットマップファイルに保存するためには、CCD カメラへのトリガー信号をデジタル入出力ユニットに入力し、デジタル入出力ユニットからアプリケーションが外部トリガーイベントを受け取る必要があります。

推奨動作環境

- CPU PentiumII 500MHz 以上
- 1394 ホストアダプタ（REX-PFW3W もしくは REX-CFW3H）
- メインメモリ 128M 以上搭載
- HDD 回転数 7200rpm / 空き容量 1G 以上
- ビデオメモリ 8MB 以上
- グラフィック表示モード 1280×1024・True Color(32 ビット)・オーバーレイ表示サポート
- OS Win98SE,Win2K(SP2 以上),WinMe,WinXP
- Microsoft Vusual C/C++ 6.0 もしくは Visual BASIC 6.0 SP5 以上
- DirectX 8.0 以上（DirectX SDK 不要）

1-2. 梱包内容の確認

- REX-PFW3W 1394 Host PCI Board と Low Profile 取り付け金具
- REX-CFW3H 1394 Host CardBus Card と AC 電源
- RS1394CCD カメラドライバ CD-ROM
- RS1394CCD カメラドライバ USB キー
- RS1394CCD カメラドライバユーザーズマニュアル
- REX-PFW3W もしくは REX-CFW3H 製品保証書

- RS1394CCD SDK CD-ROM
- SDK ユーザーズマニュアル
- SDK 使用許諾書
- SDK ユーザ登録申込書

REX-PFW3W・REX-CFW3H は製品購入時いずれか選択になります。

1-3. 制約・注意事項

サポート条件

サポート期間は製品をご購入された後、1 年間とさせていただきます。1 年経過後のサポートは別途契約が必要になります。また、本製品ご購入後 1 週間以内にユーザ登録申し込みを行われていない場合、ユーザ登録書をお持ちでない場合、サポートは受け付けられませんのでご了承願います。また、サポート内容によってはお客様の作成されたアプリケーションソースコードのご提供が必須となる場合があります。

尚、ユーザサポートは E-mail もしくは FAX でお願い致します。

アプリケーション運用上の制約

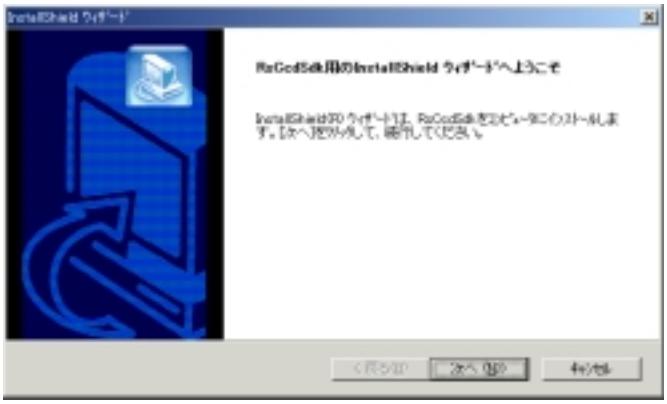
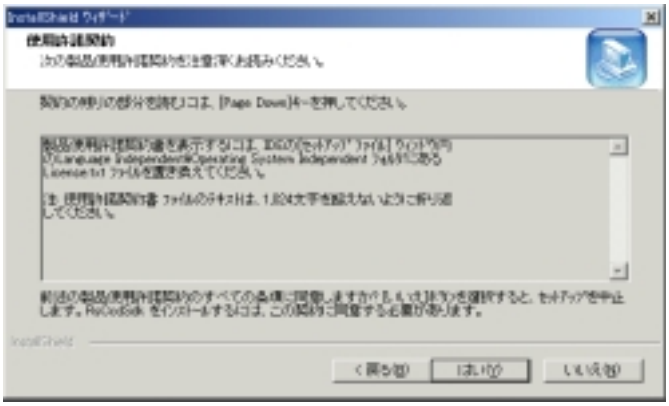
本 RS1394CCD SDK を使用して作成されたアプリケーション（弊社より提供している各種ドライバ、ライブラリ等のオブジェクトを含む）の動作保証はお客様サイドで責任を持って頂くことを前提としています。本製品の運用を理由とする損失、免失利益などの請求につきましては、いかなる責任も負いかねますので予めご了承願います。

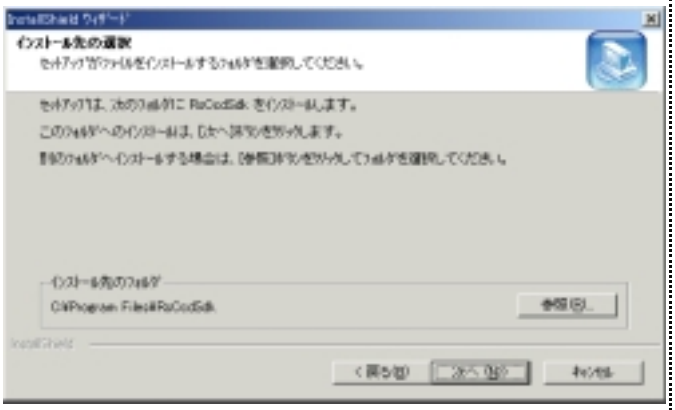
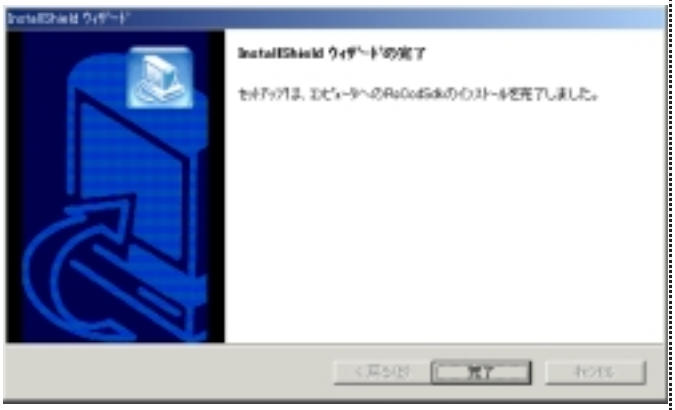
第二章 CCD SDK インストレーション

Visual C/C++, Visual Basic でのアプリケーション作成を行う前に、CCD SDK の COM コンポーネントと ActiveX コンポーネントのセットアップを行って下さい。CCD SDK の COM コンポーネントと ActiveX コンポーネントの配布はライセンスフリーとなっています。

2-1.CCD SDK コンポーネントセットアップ

CCD SDK コンポーネントセットアップを行う前に、別冊の 1394 CCD カメラドライバユーザーズマニュアルを参照し 1394 ホストカードと CCD カメラドライバのセットアップを完了して下さい。

RS1394CCD SDK CD-ROM の「Setup」ﾌｫﾙﾀﾞにある「Setup.exe」を実行します。日本語を選択し、「OK」ﾎﾞﾀﾝをｸﾘｯｸします。	
後は指示に従って内容を確認しまがら、順次「次へ」のﾎﾞﾀﾝをｸﾘｯｸします。	
使用許諾内容を確認し問題なければ、「はい」のﾎﾞﾀﾝをｸﾘｯｸします。 （注意） 右使用許諾内容の内容は、本製品添付のソフトウェア使用許諾書の内容と同じです。	

ｲﾝｽﾄｰﾙ先ﾌｫﾙﾀﾞを確認し、特に問題なければ「次へ」のﾎﾞﾀﾝをｸﾘｯｸします。	
以上で、CCD SDK の COM ｺﾝﾎﾟｰﾈﾝﾄと ActiveX ｺﾝﾎﾟｰﾈﾝﾄのｾｯﾄｱｯﾌﾟは完了です。	

2-2. CCD SDK コンポーネントセットアップの内容

「2-1.CCD SDK コンポーネントセットアップ」で実行した「Setup.exe」は InstallShield Professional Standard Edition Version 6.30 で作成しています。「SETUP.RUL」ファイルが CD-ROM\Setup\InstallShield フォルダに格納されていますので、これを参考にご自身で作成されたアプリケーションのインストーラに CCD SDK コンポーネントのセットアップも組み込むことができます。「SETUP.RUL」で CCD SDK コンポーネントの組み込みを行っている部分は下記の通りです。%Windows ディレクトリ%の System32 フォルダに COM ライブラリ「RSCCD.COM」と ActiveX ライブラリ「VBCCDSDK.DLL」をコピーし、「Regsvr32.exe」ユーティリティでこれらのコンポーネントのレジストリ登録を行えばセットアップ完了です。

System32 フォルダに「RSCCD.COM」と「VBCCDSDK.DLL」をコピーした後、コマンドプロンプトもしくは Windows のスタートメニューの「ファイル名を指定して実行」から登録を行う場合は下記のようになります。

> regsvr32 /s c:winnt\system32\rsccd.dll

> regsvr32 /s c:winnt\system32\vbccdsdl.dll

```
////////////////////////////////////////////////////////
//  OnMoved...ファイルのコピー、削除が終了したときに
//  呼ばれる。
function OnMoved()
   string szProgram, szCmdLine;
begin
   if !MAINTENANCE then    //ﾒﾝﾃﾅﾝｽﾓｰﾄﾞでないときﾚｼﾞｽﾄﾘ登録
        szProgram = "regsvr32.exe /s";
        szCmdLine =  WINDIR ^ "System32\\RsCcd.dll";
        LaunchAppAndWait(szProgram,szCmdLine,WAIT);
        szCmdLine =  WINDIR ^ "System32\\VBCCDSDK.DLL";
        LaunchAppAndWait(szProgram,szCmdLine,WAIT);
   endif;
end;
```

"RsCcd.dll"及び" VBCCDSDK.DLL"がｺﾋﾟｰされた Windows ﾃﾞｨﾚｸﾄﾘの System32 ﾌｫﾙﾀﾞを指定し、"regsvr32.exe /s"を実行する。

第三章 Visual C/C++によるﾌﾟﾛｸﾞﾗﾑ開発

Visual C/C++でアプリケーションを作成する場合は、本製品付属の CCD SDK COM ライブラリを使用します。CCD SDK のインストレーションを行えば CCD SDK COM ライブラリが使用可能な状態となります。以下の CCD SDK COM ライブラリインターフェイスとサンプルプログラムを参考にして、アプリケーションの作成を行います。

3-1. COM インターフェイス仕様

ｲﾝﾀｰﾌｪｲｽ名	説　　明	頁
ccd_About	SDK ﾗｲﾌﾞﾗﾘのﾊﾞｰｼﾞｮﾝ情報を取得。	9
ccd_Initialize	SDK ﾗｲﾌﾞﾗﾘの初期化処理を行う。	21
ccd_Uninitialize	SDK ﾗｲﾌﾞﾗﾘの終了処理を行う。	43
ccd_EnumCamera	現在接続されている CCD ｶﾒﾗを列挙し SDK_CAM_INFO 構造体にｶﾒﾗ情報をｾｯﾄ。	10
ccd_FreeCamera	ccd_EnumCamera 呼び出しにより COM で確保されているﾘｿｰｽを解放。	11
ccd_RegCameraEvent	ｶﾒﾗの取り外しｲﾍﾞﾝﾄ、接続ｲﾍﾞﾝﾄをｱﾌﾟﾘｹｰｼｮﾝに通知されるよう設定。	26
ccd_GetCurrentFormat	指定されたｶﾒﾗの現在設定されているｽﾄﾘｰﾑﾌｫｰﾏｯﾄおよびﾌﾚｰﾑﾚｰﾄを取得。	15
ccd_SetStream	指定されたｶﾒﾗから出力されるｽﾄﾘｰﾑﾌｫｰﾏｯﾄを設定。	36
ccd_SetFrameRate	指定されたｶﾒﾗから出力されるﾌﾚｰﾑﾚｰﾄを設定。	31
ccd_PreviewStart	指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを開始。	23
ccd_PreviewPause	指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰをﾎﾟｰｽﾞ。	22
ccd_PreviewStop	指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを終了。	24
ccd_Snapshot	現在表示されているﾌﾚｰﾑを静止画ﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙに保存。	40
ccd_SetTrigger	CCD ｶﾒﾗの外部ﾄﾘｶﾞｰﾓｰﾄﾞを設定。	38
ccd_SetAudio	動画のﾌﾟﾚﾋﾞｭｰおよび保存時にｵｰﾃﾞｨｵを有効設定。	27
ccd_SetCaptureFile	動画保存するﾌﾙﾊﾟｽﾌｧｲﾙ名とｷｬﾌﾟﾁｬﾌｧｲﾙ保存にﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用するか指定。	28
ccd_SetCaptureTempFile	ｷｬﾌﾟﾁｬ用のﾃﾝﾎﾟﾗﾘﾌｧｲﾙに関する設定。	29
ccd_GetCopyFileProgress	ｷｬﾌﾟﾁｬ停止後のﾃﾝﾎﾟﾗﾘﾌｧｲﾙのｺﾋﾟｰｽﾃｰﾀｽを取得。	14
ccd_CaptureStart	指定されたｶﾒﾗより現在設定されたｽﾄﾘｰﾑ・ﾌﾚｰﾑﾚｰﾄで動画保存を開始。	41
ccd_CaptureStop	動画保存を終了。	42
ccd_SetMovieCompressor	ｷｬﾌﾟﾁｬ保存するﾌｧｲﾙの圧縮形式を設定。	33
ccd_GetPropertyRange	Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、調整ﾚﾝｼﾞ情報を取得。	20
ccd_GetProperty	Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、ｶﾒﾗより現在値を取得。	19
ccd_SetProperty	Zoom/Focus 等に関するﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更。	35
ccd_GetColorRange	UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて調整ﾚﾝｼﾞ情報を取得。	13
ccd_GetColor	UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについてｶﾒﾗより現在値を取得。	12
ccd_SetColor	UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更。	29

■ インターフェイス仕様の書式について

下記の書式にて各インターフェイスの説明を行います。

IRsCcdSdk::ccd_XXXXX

IRsCcdSdk インターフェイス

【インターフェイスの概要動作説明】

構文

```
HRESULT ccd_XXXXX( LONG XXXXX);
```

【関数仕様説明】

パラメータ

【引数説明】

[in] **【入力】**

[out] **【出力】**

戻り値

【戻り値説明】

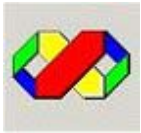
注意

【その他参考情報】

関連

【関連するｲﾝﾀｰﾌｪｲｽ】

IRsCcdSdk::ccd_About



IRsCcdSdk インターフェイス

SDK ﾗｲﾌﾞﾗﾘのﾊﾞｰｼﾞｮﾝ情報を取得。

構文

```
HRESULT ccd_About (
    LPWSTR szVersionInfo
);
```

パラメータ

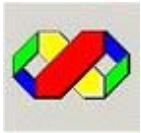
szVersionInfo
[out] SDK ﾗｲﾌﾞﾗﾘのﾊﾞｰｼﾞｮﾝ情報文字列。

戻り値

HRESULT として常に S_OK を返す。

S_OK	成功

IRsCcdSdk::ccd_EnumCamera



IRsCcdSdk インターフェイス

現在接続されている CCD ｶﾒﾗを列挙し、COM の内部情報として列挙したｶﾒﾗのﾘｿｰｽを確保。同時に、第一引数 SDK_CCD_INFO 構造体ｵﾌﾞｼﾞｪｸﾄに列挙したｶﾒﾗの情報を返す。

構文

```
HRESULT ccd_EnumCamera(
    PSDK_CCD_INFO pCameraArray, ULONG *pNumOfCamera
);
```

パラメータ

pCameraArray
[out]　SDK_CCD_INFO 構造体ｵﾌﾞｼﾞｪｸﾄﾎﾟｲﾝﾀ。
pNumOfCamera
[out]　列挙した CCD ｶﾒﾗの数を格納する変数ﾎﾟｲﾝﾀ。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

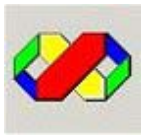
注意

ｱﾌﾟﾘｹｰｼｮﾝ側では要素数 MAX_SUPPORT_CAMERAS の SDK_CAM_INFO 型の配列を確保します。ｶﾒﾗ情報の再構築のために ccd_EnumCamera()呼び出しを行う際、ｱﾌﾟﾘｹｰｼｮﾝを終了する際には必ず ccd_FreeCamera()により COM で確保されたﾘｿｰｽを一旦解放します。

関連

ccd_EnumCamera()参照。

IRsCcdSdk::ccd_FreeCamera



IRsCcdSdk インターフェイス

ccd_EnumCamera() により COM で確保されているﾘｿｰｽを解放。

構文

```
HRESULT ccd_FreeCamera();
```

パラメータ

なし。

戻り値

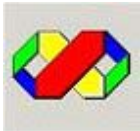
下記 HRESULT を返す。ｴﾗｰ発生時の詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_EnumCamera()参照。

IRsCcdSdk::ccd_GetColor



IRsCcdSdk インターフェイス

UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについてｶﾒﾗより現在値を取得。

構文

```
HRESULT ccd_GetColor(
  LONG CameraIndex, LONG Property, LONG *plValue, LONG fMode
);
```

パラメータ

CameraIndex
[in]　ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Property
[in]　ColorProperty より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。
plValue
[out]　指定したﾌﾟﾛﾊﾟﾃｨの値を格納する場所を示すﾎﾟｲﾝﾀ。fMode に fCameraControlAuto もしくは fCameraControlOnePush が指定されている時は無視されます。
fMode
[in]　指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_GetPropertyRange, ccd_GetProperty, ccd_SetProperty, ccd_GetColorRange, ccd_SetColor, ccd_ShowPropertyPage 参照。

IRsCcdSdk::ccd_GetColorRange

IRsCcdSdk インターフェイス

UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて調整ﾚﾝｼﾞ情報を取得。

```
HRESULT ccd_GetColorRange (
    LONG CameraIndex, LONG Property, PROPERTY_RANGE *pPropValue
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Property
[in] ColorProperty よりﾚﾝｼﾞを取得するﾌﾟﾛﾊﾟﾃｨを指定。
pPropValue
[out] ﾚﾝｼﾞ情報格納先の PROPERTY_RANGE 構造体ｵﾌﾞｼﾞｪｸﾄﾎﾟｲﾝﾀ。PROPERTY_RANGE 構造体参照。

戻り値

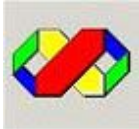
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_GetPropertyRange, ccd_GetProperty, ccd_SetProperty, ccd_GetColor, ccd_SetColor, ccd_ShowPropertyPage 参照。

IRsCcdSdk::ccd_GetCopyFileProgress



IRsCcdSdk インターフェイス

ｷｬﾌﾟﾁｬ停止後のﾃﾝﾎﾟﾗﾘﾌｧｲﾙのｺﾋﾟｰｽﾃｰﾀｽを取得。

```
HRESULT ccd_GetCopyFileProgress (
    LONG CameraIndex, LONG *lpProgress
);
```

パラメータ

CameraIndex

[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。

lpProgress

[out] ｺﾋﾟｰ進捗ｽﾃｰﾀｽが返される変数へのﾎﾟｲﾝﾀ。0-100[%]の値で 100 が完了。

戻り値

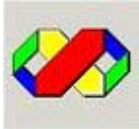
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_StartCapture, ccd_StopCapture, ccd_SetCaptureFile, ccd_SetCaptureTempFile 参照。

IRsCcdSdk::ccd_GetCurrentFormat



IRsCcdSdk インターフェイス

指定されたｶﾒﾗの現在設定されているｽﾄﾘｰﾑﾌｫｰﾏｯﾄおよびﾌﾚｰﾑﾚｰﾄを取得。

構文

```
HRESULT ccd_GetCurrentFormat (
    LONG CameraIndex, LONG *pStreamIndex, LONG *pFpsIndex
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
pStreamIndex
[out] 現在設定されているﾋﾞﾃﾞｵﾓｰﾄﾞのｲﾝﾃﾞｯｸｽ番号。SDK_CCD_INFO 構造体参照。
pFpsIndex
[out] 現在設定されているﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号。SDK_CCD_INFO 構造体参照。

戻り値

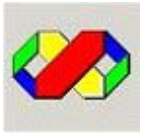
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

ccd_EnumCamera()を呼び出しにより取得したｶﾒﾗ情報構造体
SDK_CCD_INFO SdkCameraArray[MAX_SUPPORT_CAMERAS];
と ccd_GetCurrentFormat()で取得した現在のﾋﾞﾃﾞｵﾓｰﾄﾞ・ﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号により、現在のﾋﾞﾃﾞｵﾓｰﾄﾞ・ﾌﾚｰﾑﾚｰﾄはそれぞれ下記のように定めることができます。
SdkCameraArray[CameraIndex].SupportedStream[**pStreamIndex*].SupportedFormat
SdkCameraArray[CameraIndex].SupportedStream[**pStreamIndex*].SupportedFrameRate[**pFpsIndex*]

IRsCcdSdk::ccd_GetLastError



IRsCcdSdk インターフェイス

最後に発生したｴﾗｰの詳細情報を取得。

構文

```
HRESULT ccd_GetLastError (
    long *ErrNumber, LPWSTR lpwstrRrr
);
```

パラメータ

ErrNumber
[out] ｴﾗｰｺｰﾄﾞの格納先。
lpwstrRrr
[out] ｴﾗｰの内容を記述した文字列の格納先。

戻り値

HRESULT として常に S_OK を返す。

S_OK	成功

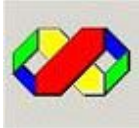
注意

ｴﾗｰの原因が推定できるものを次ﾍﾟｰｼﾞのｴﾗｰｺｰﾄﾞ一覧表に示します。対処方法に基づいて確認を行って下さい。その他、一覧表にないｴﾗｰｺｰﾄﾞについては、個々の環境における致命的なｴﾗｰと思われます。ﾊﾟｿｺﾝを再起動して下さい。

エラーコード一覧表

ｴﾗｰｺｰﾄﾞ	対　処　方　法
1	CCD ｶﾒﾗが接続されていないか、ｶﾒﾗﾄﾞﾗｲﾊﾞが正しくｲﾝｽﾄｰﾙされていません。ﾃﾞﾊﾞｲｽﾏﾈｰｼﾞｬ上にｲﾒｰｼﾞﾝｸﾞﾃﾞﾊﾞｲｽが認識されているか確認して下さい。
2	使用可能なｵｰﾃﾞｨｵﾃﾞﾊﾞｲｽが存在しません。ﾃﾞﾊﾞｲｽﾏﾈｰｼﾞｬ上にｵｰﾃﾞｨｵﾃﾞﾊﾞｲｽが認識されているか確認して下さい。
3	指定したﾄﾞﾗｲﾌﾞに十分な空き容量がありません。
4	動画ﾌｧｲﾙを作成できません。指定したﾌｧｲﾙが他のｱﾌﾟﾘｹｰｼｮﾝによって使用されていないか確認して下さい。
5	指定したﾊﾟｽは存在しません。ccd_SetCaptureFile で指定したﾊﾟｽ名が正しいか再確認して下さい。
6	指定したﾌｧｲﾙ名の拡張子が AVI ではありません。
7	静止画ﾌｧｲﾙを作成できません。指定したﾌｧｲﾙが他のｱﾌﾟﾘｹｰｼｮﾝよって使用されていないか確認して下さい。
8	指定したﾊﾟｽは存在しません。ccd_SnapShot で指定したﾊﾟｽ名が正しいか再確認して下さい。
9	指定したﾌｧｲﾙ名の拡張子が BMP ではありません。
10	無効なｶﾒﾗｲﾝﾃﾞｯｸｽが渡されました。ｲﾝﾃﾞｯｸｽ番号が有効か確認して下さい。

IRsCcdSdk::ccd_GetMemCh



IRsCcdSdk インターフェイス

ｶﾒﾗ内部のﾒﾓﾘに記憶されている設定ﾃﾞｰﾀを読み出しｶﾒﾗに再設定。

構文

```
HRESULT ccd_GetMemCh (
        LONG CameraIndex, LONG MemoryChannel
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
MemoryChannel
[in] ｶﾒﾗ内部のﾒﾓﾘﾁｬﾝﾈﾙ番号を指定。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

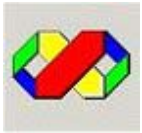
注意

本機能の詳細に関しては CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞを参照願います。

関連

ccd_SetMemCh 参照。

IRsCcdSdk::ccd_GetProperty



IRsCcdSdk インターフェイス

Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、ｶﾒﾗより現在値を取得。

構文

```
HRESULT ccd_GetProperty (
    LONG CameraIndex, LONG Property, LONG *plValue, LONG fMode
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Property
[in] BaseProperty より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。
plValue
[out] 指定したﾌﾟﾛﾊﾟﾃｨの値を格納する場所を示すﾎﾟｲﾝﾀ。
fMode
[in] 指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。Camera Control Flag を使って指定。

戻り値

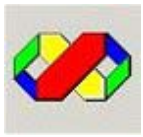
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_GetPropertyRange, ccd_SetProperty, ccd_GetColorRange, ccd_GetColor, ccd_SetColor, ccd_ShowPropertyPage 参照。

IRsCcdSdk::ccd_GetPropertyRange



IRsCcdSdk インターフェイス

Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、調整ﾚﾝｼﾞ情報を取得。

構文

```
HRESULT ccd_GetPropertyRange(
        LONG CameraIndex, LONG Property, PROPERTY_RANGE *pPropValue
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Property
[in] BaseProperty よりﾚﾝｼﾞを取得するﾌﾟﾛﾊﾟﾃｨを指定。
pPropValue
[out] ﾚﾝｼﾞ情報格納先の PROPERTY_RANGE 構造体ｵﾌﾞｼﾞｪｸﾄﾎﾟｲﾝﾀ。PROPERTY_RANGE 構造体参照。

戻り値

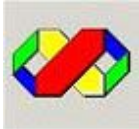
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_GetProperty, ccd_SetProperty, ccd_GetColorRange, ccd_GetColor, ccd_SetColor, ccd_ShowPropertyPage 参照。

IRsCcdSdk::ccd_Initialize



IRsCcdSdk インターフェイス

SDK ﾗｲﾌﾞﾗﾘの初期化処理。他のｲﾝﾀｰﾌｪｲｽﾗｲﾌﾞﾗﾘの呼び出しを行う前に必ず SDK ﾗｲﾌﾞﾗﾘの初期化処理を行います。

構文

```
HRESULT ccd_Initialize ( );
```

パラメータ

なし

戻り値

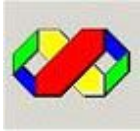
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

ｱﾌﾟﾘｹｰｼｮﾝ終了時、必ず ccd_Uninitialize により SDK ﾗｲﾌﾞﾗﾘの終了処理を行います。

IRsCcdSdk::ccd_PreviewPause



[IRsCcdSdk インターフェイス]

指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰをﾎﾟｰｽﾞ。

構文

```
HRESULT ccd_PreviewPause( LONG CameraIndex );
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。

戻り値

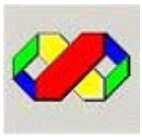
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_PreviewStart, ccd_PreviewStop 参照。

IRsCcdSdk::ccd_PreviewStart



IRsCcdSdk インターフェイス

指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを開始。

構文

```
HRESULT ccd_PreviewStart( LONG CameraIndex, HWND hWnd );
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
hWnd
[in] ﾌﾟﾚﾋﾞｭｰｳｨﾝﾄﾞｳのﾊﾝﾄﾞﾙ。NULL を渡した場合はﾃﾞﾌｫﾙﾄのﾌﾟﾚﾋﾞｭｰｳｨﾝﾄﾞｳが表示されます。

戻り値

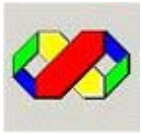
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_PreviewPause, ccd_PreviewStop 参照。

IRsCcdSdk::ccd_PreviewStop



[IRsCcdSdk インターフェイス]

指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを終了。

構文

```
HRESULT ccd_PreviewStop( LONG CameraIndex );
```

パラメータ

CameraIndex
[out] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。

戻り値

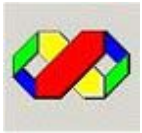
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_PreviewStart, ccd_PreviewPause 参照。

IRsCcdSdk::ccd_ReadRegister



IRsCcdSdk インターフェイス

指定のｶﾒﾗのｵﾌｾｯﾄｱﾄﾞﾚｽから 4 ﾊﾞｲﾄ（Quadlet）ﾘｰﾄﾞ。

構文

HRESULT ccd_ReadRegister (LONG *CameraIndex*, LONG *Offset*, LONG **plValue*);

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Offset
[in] ﾘｰﾄﾞするﾚｼﾞｽﾀのﾍﾞｰｽｱﾄﾞﾚｽからのﾊﾞｲﾄｵﾌｾｯﾄ。
plValue
[out] ﾘｰﾄﾞした値の格納先ｱﾄﾞﾚｽ。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

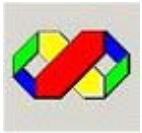
注意

ccd_ReadRegister ﾒｿｯﾄﾞの *Offset* 引数は、CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞに示されている ”Camera Control & Status Register（CSR） ”のｱﾄﾞﾚｽを指定します。例えば、SONY DFW-VL500 の場合 CSR ｱﾄﾞﾚｽは F0F00000h からﾏｯﾋﾟﾝｸﾞされています。
CSR ｱﾄﾞﾚｽは、Configuration ROM の Unit Dependent Directory の command_regs_base とその key_type と Key_value により求めることができます。
詳細は、IIDC Specification および IEEE 1212 ｱﾄﾞﾚｯｼﾝｸﾞ規格参照。

関連

ccd_WriteRegister 参照。

IRsCcdSdk::ccd_RegCameraEvent



IRsCcdSdk インターフェイス

ｶﾒﾗの取り外しｲﾍﾞﾝﾄ、接続ｲﾍﾞﾝﾄをｱﾌﾟﾘｹｰｼｮﾝに通知されるよう設定。

構文

```
HRESULT ccd_RegCameraEvent (HWND hWnd);
```

パラメータ

hWnd
[in]　ｲﾍﾞﾝﾄを受け取るｳｨﾝﾄﾞｳのﾊﾝﾄﾞﾙ。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

ccd_RegCameraEvent()によるｲﾍﾞﾝﾄ登録完了後、ｱﾌﾟﾘｹｰｼｮﾝは新しくｶﾒﾗﾃﾞﾊﾞｲｽが接続された時、もしくは取り外しが発生する毎に下記 WindowProc を通して WM_DEVICECHANGE ｲﾍﾞﾝﾄを受け取ります。wParam には、新しくｶﾒﾗが接続された場合には DBT_DEVICEARRIVAL、取り外された場合には DBT_DEVICEREMOVECOMPLETE がｾｯﾄされます。

```
LRESULT CALLBACK WindowProc(
  HWND hwnd,        // handle to window
  UINT uMsg,        // WM_DEVICECHANGE
  WPARAM wParam,    // device-change event
  LPARAM lParam     // event-specific data
);
```

関連

ccd_EnumCamera, ccd_FreeCamera 参照。

IRsCcdSdk::ccd_SetAudio

IRsCcdSdk インターフェイス

動画のﾌﾟﾚﾋﾞｭｰおよび保存時、同時にｵｰﾃﾞｨｵﾃﾞｰﾀを有効にするか無効にするか設定。

構文

```
HRESULT ccd_SetAudio(LONG CameraIndex, LONG bAudioOn);
```

パラメータ

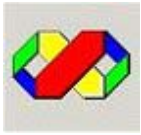
CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
bAudioOn
[in] ｵｰﾃﾞｨｵ入力の ON/OFF 設定。TRUE: ｵｰﾃﾞｨｵｵﾝ FALSE: ｵｰﾃﾞｨｵｵﾌ (ﾃﾞﾌｫﾙﾄ)

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

IRsCcdSdk::ccd_SetCaptureFile



IRsCcdSdk インターフェイス

動画を保存するﾌﾙﾊﾟｽﾌｧｲﾙ名と、ｷｬﾌﾟﾁｬﾌｧｲﾙ保存にﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用するか指定。ﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用しない場合、ccd_StartCapture によりｷｬﾌﾟﾁｬを開始すると第 2 引数で指定したﾌｧｲﾙに直接動画ﾃﾞｰﾀが保存されます。

構文

```
HRESULT ccd_SetCaptureFile (
    LONG CameraIndex, LPCWSTR lpcwstrCaptureFile, LONG bTempFileOn
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
lpcwstrCaptureFile
[in] ｷｬﾌﾟﾁｬ出力名とﾌﾙﾊﾟｽを示すﾜｲﾄﾞ文字列へのﾎﾟｲﾝﾀ（終端 NULL）。
bTempFileOn
[in] ﾃﾝﾎﾟﾗﾘﾌｧｲﾙ設定ﾌﾗｸﾞ。TRUE: ﾃﾝﾎﾟﾗﾘﾌｧｲﾙ使用　FALSE: 直接指定ﾌｧｲﾙに保存（ﾃﾞﾌｫﾙﾄ）

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

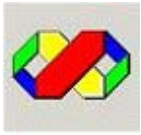
注意

ﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用する設定を行った場合は、基本的に ccd_SetCaptureTempFile によりﾃﾝﾎﾟﾗﾘﾌｧｲﾙの設定を行って下さい。ccd_StopCapture によりｷｬﾌﾟﾁｬを停止すると、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙから第 2 引数で指定したﾌｧｲﾙにｷｬﾌﾟﾁｬﾃﾞｰﾀがｺﾋﾟｰされます。

関連

ccd_StartCapture, ccd_StopCapture, ccd_SetCaptureTempFile, ccd_GetCopyFileProgress 参照。

IRsCcdSdk::ccd_SetCaptureTempFile



[IRsCcdSdk インターフェイス]

ｷｬﾌﾟﾁｬ用のﾃﾝﾎﾟﾗﾘﾌｧｲﾙに関する設定。領域確保ｵﾝにすると、ｷｬﾌﾟﾁｬ実行前に予めﾊｰﾄﾞﾃﾞｨｽｸ上に指定ｻｲｽﾞのﾝﾎﾟﾗﾘﾌｧｲﾙ領域が確保されｷｬﾌﾟﾁｬ実行時の負荷が小さくなります。

構文

```
HRESULT ccd_SetCaptureTempFile (
    LONG CameraIndex, LPCWSTR lpcwstrCaptureFile, LONG bPreAllocOn, long lAllocMBytes
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
lpcwstrCaptureFile
[in] ｷｬﾌﾟﾁｬﾃﾝﾎﾟﾗﾘﾌｧｲﾙ名とﾌﾙﾊﾟｽを示すﾜｲﾄﾞ文字列へのﾎﾟｲﾝﾀ（終端 NULL)。
bPreAllocOn
[in] ﾌｧｲﾙｱﾛｹｰｼｮﾝ設定。TRUE: 領域確保ｵﾝ　FALSE: 領域確保ｵﾌ（ﾃﾞﾌｫﾙﾄ）
lAllocMBytes
[in] ﾊｰﾄﾞﾃﾞｨｽｸ上に領域確保するﾌｧｲﾙのｻｲｽﾞ　(ﾒｶﾞﾊﾞｲﾄ単位)。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。	S_OK	成功
	S_FALSE	失敗

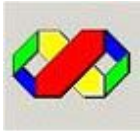
注意

ﾊｰﾄﾞﾃﾞｨｽｸ上に予めｷｬﾌﾟﾁｬ領域を確保することにより、動画ｷｬﾌﾟﾁｬ時のｵｰﾊﾞｰﾍｯﾄﾞを最小にすることができます。第 2 引数のﾌｧｲﾙ名に NULL を指定すると、環境変数 TEMP で定義されたﾌｫﾙﾀﾞにﾃﾞﾌｫﾙﾄのﾌｧｲﾙ名「RSCCDTMP.AVI」でﾃﾝﾎﾟﾗﾘﾌｧｲﾙが作成されます。このﾃﾝﾎﾟﾗﾘﾌｧｲﾙはｺﾋﾟｰ終了時削除されます。

関連

ccd_StartCapture, ccd_StopCapture, ccd_SetCaptureFile, ccd_GetCopyFileProgress 参照。

IRsCcdSdk::ccd_SetColor



[IRsCcdSdk インターフェイス]

UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更する。

構文

```
HRESULT ccd_SetColor (
    LONG CameraIndex, LONG Property, LONG lValue, LONG fMode
);
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Property
[in] ColorProperty より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。
lValue
[in] 新たに設定するﾌﾟﾛﾊﾟﾃｨの値。fMode に fCameraControlAuto もしくは
fCameraControlOnePush が指定されている時は無視されます。
fMode
[in] 指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。

戻り値

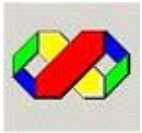
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_GetPropertyRange, ccd_GetProperty, ccd_SetProperty, ccd_GetColorRange, ccd_GetColor, ccd_ShowPropertyPage 参照。

IRsCcdSdk::ccd_SetFrameRate



IRsCcdSdk インターフェイス

指定されたｶﾒﾗから出力されるﾌﾚｰﾑﾚｰﾄを設定。

構文

```
HRESULT ccd_SetFrameRate ( LONG CameraIndex, LONG NewFrameRate );
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
NewFrameRate
[in] 新しく設定するﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号。SDK_CCD_INFO 構造体参照。

戻り値

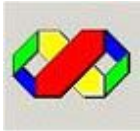
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_EnumCamera, ccd_GetCurrentFormat, ccd_SetStream 参照。

IRsCcdSdk::ccd_SetMemCh



IRsCcdSdk インターフェイス

現在ｶﾒﾗに設定されているﾃﾞｰﾀをｶﾒﾗ内部のﾒﾓﾘに記憶。

構文

```
HRESULT ccd_SetMemCh ( LONG CameraIndex, LONG MemoryChannel );
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
MemoryChannel
[in] ｶﾒﾗ内部のﾒﾓﾘﾁｬﾝﾈﾙ番号を指定。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

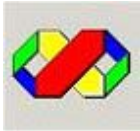
注意

本機能の詳細に関しては CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞを参照願います。

関連

ccd_GetMemCh 参照。

IRsCcdSdk::ccd_SetMovieCompressor



IRsCcdSdk インターフェイス

ｷｬﾌﾟﾁｬ保存するﾌｧｲﾙの圧縮形式を設定。ﾃﾞﾌｫﾙﾄでは無圧縮形式で保存されます。

構文

HRESULT ccd_SetMovieCompressor
(LONG *CameraIndex*, LONG *CompressorIndex*, LPCWSTR *lpcwstrCompName*)

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
CompressorIndex
[in] 圧縮ﾌｫｰﾏｯﾄを示すｲﾝﾃﾞｯｸｽ番号を CompressorFormat 構造体で指定。
lpcwstrCompName
[in] 圧縮ﾌｫｰﾏｯﾄ名が格納されているﾜｲﾄﾞ文字列へのﾎﾟｲﾝﾀ（終端 NULL）。圧縮ﾌｫｰﾏｯﾄｲﾝﾃﾞｯｸｽ番号 *CompressorIndex* に SetByName が指定された場合は、この文字列で圧縮ﾌｫｰﾏｯﾄを指定します。圧縮ﾌｫｰﾏｯﾄ名は DirectShow のﾌｨﾙﾀ名と一致する必要があります。SetByName 以外の圧縮ﾌｫｰﾏｯﾄｲﾝﾃﾞｯｸｽ番号 *CompressorIndex* を指定している場合は、NULL を渡します。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

非圧縮ﾃﾞｰﾀをそのままｷｬﾌﾟﾁｬした場合は、処理が間に合わずこま落ち等の問題が発生します。DV Video Encoder 等の圧縮を行ってｷｬﾌﾟﾁｬすることによりこま落ち問題を改善することができます。

関連

ccd_StartCapture, ccd_StopCapture 参照。

IRsCcdSdk::ccd_SetOverlay

IRsCcdSdk インターフェイス

ｵｰﾊﾞｰﾚｲ機能のｲﾈｰﾌﾞﾙ・ﾃﾞｨｾｰﾌﾞﾙを設定。

構文

HRESULT ccd_SetOverlay (LONG *CameraIndex*, LONG *bOverlayOn*);

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
bOverlayOn
[in] ｵｰﾊﾞｰﾚｲ機能の ON/OFF 設定。TRUE:ｵﾝ FALSE:ｵﾌ

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

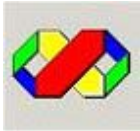
注意

1．ｵｰﾊﾞｰﾚｲ機能のｻﾎﾟｰﾄについて
ｵｰﾊﾞｰﾚｲ機能を有効にして CCD ｶﾒﾗからの動画ﾌﾟﾚﾋﾞｭｰを行うと、高速な動画表示が可能になります。ｵｰﾊﾞｰﾚｲ機能のｻﾎﾟｰﾄは使用しているｸﾞﾗﾌｨｯｸｱﾀﾞﾌﾟﾀに依存します。ｵｰﾊﾞｰﾚｲ機能がｻﾎﾟｰﾄされたｸﾞﾗﾌｨｯｸｱﾀﾞﾌﾟﾀでも､ｱﾀﾞﾌﾟﾀ側の問題によりﾌﾟﾚﾋﾞｭｰ画面の拡大縮小ができない場合があります。
2．外部ﾄﾘｶﾞｰﾓｰﾄﾞについて
ｵｰﾊﾞｰﾚｲﾓｰﾄﾞを有効にした場合は、外部ﾄﾘｶﾞｰﾓｰﾄﾞは使用できません。

関連

ccd_SetTrigger 参照。

IRsCcdSdk::ccd_SetProperty



IRsCcdSdk インターフェイス

Zoom/Focus 等に関するﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更。

構文

HRESULT ccd_SetProperty (LONG *CameraIndex*, LONG *Property*, LONG *lValue*, LONG *fMode*);

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Property
[in] BaseProperty より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。
lValue
[in] 新たに設定するﾌﾟﾛﾊﾟﾃｨの値。fMode に fCameraControlAuto もしくは
fCameraControlOnePush が指定されている時は無視されます。
fMode
[in] 指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。Camera Control Flag を使って指定。

戻り値

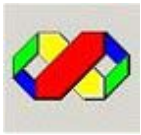
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_GetPropertyRange, ccd_GetProperty, ccd_GetColorRange, ccd_GetColor, ccd_SetColor, ccd_ShowPropertyPage 参照。

IRsCcdSdk::ccd_SetStream



[IRsCcdSdk インターフェイス]

指定されたｶﾒﾗから出力されるｽﾄﾘｰﾑﾌｫｰﾏｯﾄを設定。

構文

```
HRESULT ccd_SetStream ( LONG CameraIndex, LONG StreamIndex );
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
StreamIndex
[in] 新しく設定するｽﾄﾘｰﾑのｲﾝﾃﾞｯｸｽ番号。SDK_CCD_INFO 構造体参照。

戻り値

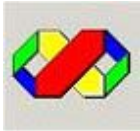
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_EnumCamera, ccd_GetCurrentFormat, ccd_SetFrameRate 参照。

IRsCcdSdk::ccd_WriteRegister



IRsCcdSdk インターフェイス

指定のｶﾒﾗのｵﾌｾｯﾄｱﾄﾞﾚｽに 4 ﾊﾞｲﾄ（Quadlet）ﾗｲﾄ。

構文

```
HRESULT ccd_WriteRegister ( LONG CameraIndex, LONG Offset, LONG lValue );
```

パラメータ

CameraIndex
[in]　ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
Offset
[in]　ﾗｲﾄするﾚｼﾞｽﾀのﾍﾞｰｽｱﾄﾞﾚｽからのﾊﾞｲﾄｵﾌｾｯﾄ。
lValue
[in]　ﾗｲﾄする値。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

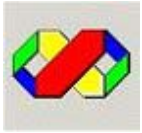
注意

ccd_ReadRegister ﾒｿｯﾄﾞの Offset 引数は、CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞに示されている ”Camera Control & Status Register（CSR） ”のｱﾄﾞﾚｽを指定します。例えば、SONY DFW-VL500 の場合 CSR ｱﾄﾞﾚｽは F0F00000h からﾏｯﾋﾟﾝｸﾞされています。
CSR ｱﾄﾞﾚｽは、Configuration ROM の Unit Dependent Directory の command_regs_base とその key_type と Key_value により求めることができます。
詳細は、IIDC Specification および IEEE 1212 ｱﾄﾞﾚｯｼﾝｸﾞ規格参照。

関連

ccd_ReadRegister 参照。

IRsCcdSdk:: ccd_SetTrigger



IRsCcdSdk インターフェイス

CCD ｶﾒﾗの外部ﾄﾘｶﾞｰﾓｰﾄﾞの ON/OFF を設定。

構文

HRESULT ccd_SetTrigger(LONG *CameraIndex*, LONG *bTriggerOn*);

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
bTriggerOn
[in] 外部ﾄﾘｶﾞｰﾓｰﾄﾞの ON/OFF 設定。TRUE:ｵﾝ FALSE:ｵﾌ

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

1．ﾄﾘｶﾞｰ画像ﾃﾞｰﾀの遅れについて
外部ﾄﾘｶﾞｰ信号が CCD ｶﾒﾗに入力されると、CCD ｶﾒﾗより 1 ﾌﾚｰﾑの画像が送られます。この時、実際に外部信号が入力された時点の画像ﾃﾞｰﾀと、CCD ｶﾒﾗより送られた画像ﾃﾞｰﾀには 100 ﾐﾘ秒前後の遅れがあります。
2.ﾄﾘｶﾞｰｲﾍﾞﾝﾄの取得について
ｱﾌﾟﾘｹｰｼｮﾝがﾄﾘｶﾞｰの発生を検出するためには、外部ﾄﾘｶﾞｰ発生装置からのﾄﾘｶﾞｰ信号は CCD ｶﾒﾗに入力すると同時に、ﾃﾞｼﾞﾀﾙ I/O 入出力ﾕﾆｯﾄを介してﾊﾟｿｺﾝにも入力する必要があります。
例えば、外部ﾄﾘｶﾞｰﾓｰﾄﾞでﾌﾟﾚﾋﾞｭｰを実行した場合、外部ﾄﾘｶﾞｰの入力があると CCD ｶﾒﾗより 1 ﾌﾚｰﾑの画像ﾃﾞｰﾀ送られ、ﾌﾟﾚﾋﾞｭｰ画面に 1 ﾌﾚｰﾑの静止画が表示されます。人がその画像を見てｽﾅｯﾌﾟｼｮｯﾄを実行することはできますが、ｱﾌﾟﾘｹｰｼｮﾝが自動で外部ﾄﾘｶﾞｰを検出してｽﾅｯﾌﾟｼｮｯﾄを実行する場合は、ﾃﾞｼﾞﾀﾙ I/O 入出力ﾕﾆｯﾄが必要になります。
3.ｵｰﾊﾞｰﾚｲﾓｰﾄﾞについて
外部ﾄﾘｶﾞｰﾓｰﾄﾞを有効にした場合は、ｵｰﾊﾞｰﾚｲ描画はできません。ccd_SetOverlay 参照。

IRsCcdSdk::ccd_ShowPropertyPage

IRsCcdSdk インターフェイス

ｶﾒﾗﾄﾞﾗｲﾊﾞのﾌﾟﾛﾊﾟﾃｨﾍﾟｰｼﾞを表示。

構文

```
HRESULT ccd_ShowPropertyPage ( LONG CameraIndex );
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。

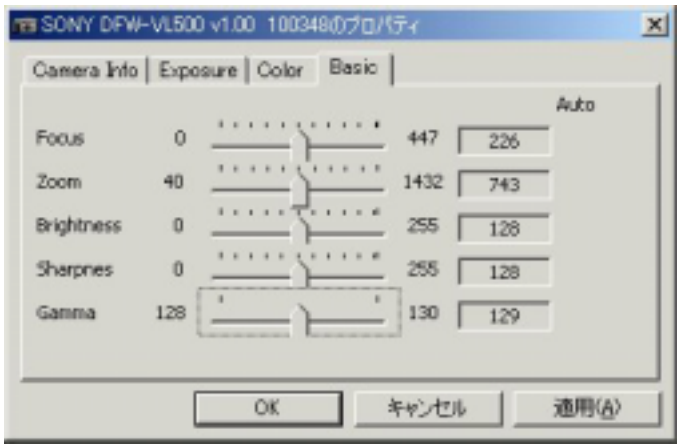
戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

ｶﾒﾗﾄﾞﾗｲﾊﾞのﾌﾟﾛﾊﾟﾃｨﾍﾟｰｼﾞを右に示します。



関連

ccd_GetPropertyRange, ccd_GetProperty, ccd_SetProperty, ccd_GetColorRange, ccd_GetColor, ccd_SetColor 参照。

IRsCcdSdk::ccd_Snapshot

IRsCcdSdk インターフェイス

現在表示されているﾌﾚｰﾑを静止画ﾋﾞｯﾄﾏｯﾌﾟ ﾌｧｲﾙに保存。

構文

HRESULT ccd_Snapshot (LONG *CameraIndex*, LPCWSTR *lpcwstrSnapFile*);

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。
lpcwstrSnapFile
[in] 出力ﾌｧｲﾙ名が格納されているﾜｲﾄﾞ文字列へのﾎﾟｲﾝﾀ（終端 NULL）。

戻り値

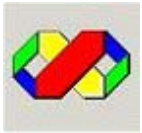
HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

ｽﾅｯﾌﾟｼｮｯﾄしたﾌﾚｰﾑはｶﾗｰ CCD の場合は RGB24、ﾓﾉｸﾛ CCD の場合は RGB8 ﾌｫｰﾏｯﾄで保存されます。

IRsCcdSdk::ccd_CaptureStart



IRsCcdSdk インターフェイス

指定されたｶﾒﾗより現在設定されたｽﾄﾘｰﾑ・ﾌﾚｰﾑﾚｰﾄで動画保存を開始。

構文

HRESULT ccd_CaptureStart(LONG *CameraIndex*, LONG *bPreviewOn* , HWND *hWnd*);

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照
bPreviewOn
[in] ｷｬﾌﾟﾁｬ動作中のﾌﾟﾚﾋﾞｭｰﾓｰﾄﾞの ON/OFF 設定。TRUE:ﾌﾟﾚﾋﾞｭｰする FALSE:ﾌﾟﾚﾋﾞｭｰしない
hWnd
[in] ﾌﾟﾚﾋﾞｭｰｳｨﾝﾄﾞｳのﾊﾝﾄﾞﾙ。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

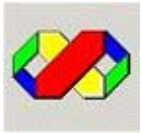
注意

ccd_SetCaptureFilename のﾃﾝﾎﾟﾗﾘﾌｧｲﾙ設定ﾌﾗｸﾞが True の場合は、一旦ﾃﾝﾎﾟﾗﾘﾌｧｲﾙに動画が保存されます。

関連

ccd_CaptureStop, ccd_SetCaptureFilename, ccd_SetCaptureTempFile, ccd_GetCopyFileProgress 参照。

IRsCcdSdk::ccd_CaptureStop



IRsCcdSdk インターフェイス

動画保存を終了。

構文

```
HRESULT ccd_CaptureStop ( LONG CameraIndex );
```

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 SDK_CCD_INFO 構造体参照。

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

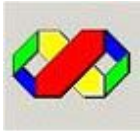
注意

ccd_SetCaptureFilename のﾃﾝﾎﾟﾗﾘﾌｧｲﾙ設定ﾌﾗｸﾞが True の場合は、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙに保存された動画が ccd_SetCaptureFilename で指定したﾌｧｲﾙにｺﾋﾟｰします。ccd_GetCopyFileProgress を呼び出し、何%ｺﾋﾟｰが終了したが進捗情報を取得することができます。

関連

ccd_CaptureStart, ccd_SetCaptureFilename, ccd_SetCaptureTempFile, ccd_GetCopyFileProgress 参照。

IRsCcdSdk::　ccd_Uninitialize



IRsCcdSdk インターフェイス

SDK ﾗｲﾌﾞﾗﾘの終了処理を行う。

構文

```
HRESULT ccd_Uninitialize();
```

パラメータ

なし

戻り値

HRESULT を返す。以下のいずれかの値。ｴﾗｰの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

関連

ccd_Initialize()参照。

3-2. 構造体・フラグ仕様

SDK_CCD_INFO

```
typedef struct _sdk_cam_info{
      LONG                        CameraIndex;
      WCHAR                       wCameraName[128];
      ULONG                       SupportedBitmapFormat;
      SDK_SUPPORTED_STREAM        SupportedStream[10];
      char                        Reserved[6];
} SDK_CCD_INFO, *PSDK_CCD_INFO;
```

CameraIndex	列挙した CCD ｶﾒﾗの 0 から始まるｲﾝﾃﾞｯｸｽ番号。
wCameraName[128]	CCD ｶﾒﾗ名が格納されるﾜｲﾄﾞ文字列ﾊﾞｯﾌｧ。
SupportedBitmapFormat	CCD ｶﾒﾗがｻﾎﾟｰﾄするﾋﾟｸｾﾙﾌｫｰﾏｯﾄのﾋﾞｯﾄﾌﾗｸﾞ。
SupportedStream[10]	CCD ｶﾒﾗがｻﾎﾟｰﾄするﾋﾞﾃﾞｵﾓｰﾄﾞ。各ﾋﾞﾃﾞｵﾓｰﾄﾞの詳細については SDK_SUPPORTED_STREAM 構造体参照。配列の要素番号がｽﾄﾘｰﾑのｲﾝﾃﾞｯｸｽ番号に相当。
Reserved[6]	予約。

```
typedef struct _sdk_supported_stream{
      LONG                        SupportedFormat;
      SIZE                        SupportedPixSize;
      LONG                        SupportedFrameRate[8];
      LONG                        Reserved;
} SDK_SUPPORTED_STREAM, *PSDK_SUPPORTED_STREAM;
```

SupportedFormat	CCD ｶﾒﾗがｻﾎﾟｰﾄしているﾌｫｰﾏｯﾄを示すｲﾝﾃﾞｯｸｽ番号。PixelFormat Enum 構造体の番号と 1 対 1 の対応になります。					
	番号	4	3	2	1	0
	ﾌｫｰﾏｯﾄ	YUV(4:1:1)	YUV(4:2:2)	YUV(4:4:4)	Y(MONO)	RGB32
SupportedPixSize	CCD ｶﾒﾗがｻﾎﾟｰﾄしている上記ﾌｫｰﾏｯﾄ時のﾋﾟｸｾﾙｻｲｽﾞ。					
SupportedFrameRate[8]	CCD ｶﾒﾗがｻﾎﾟｰﾄしている上記ﾌｫｰﾏｯﾄ時の 100 ﾅﾉ秒単位のﾌﾚｰﾑﾚｰﾄ。配列の要素番号がﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号に相当。					
Reserved	予約。					

PROPERTY_RANGE

```
typedef struct _ccd_propertyvalue
{
        long            Min;
        long            Max;
        long            SteppingDelta;
        long            Default;
        long            CapsFlags;
} PROPERTY_RANGE, *PPROPERTY_RANGE;
```

Min	設定可能な最小値。
Max	設定可能な最大値。
SteppingDelta	1 ｽﾃｯﾌﾟ増分値。
Default	ﾃﾞﾌｫﾙﾄ値。
CapsFlags	ﾋﾞｯﾄﾏｯﾌﾟで定義されたｻﾎﾟｰﾄﾓｰﾄﾞ。

関連

ccd_GetPropertyRange, ccd_GetColorRange

BaseProperty

```
typedef enum
{ Focus=0, Zoom, Brightness, Sharpness, Gamma, Autoexp, Shutter, Gain, Iris } BaseProperty;
```

関連

ccd_GetPropertyRange, ccd_GetProperty, ccd_SetProperty

ColorProperty

typedef enum{ UB = 0, VR, Hue, Saturation, } ColorProperty;

関連

ccd_GetColorRange, ccd_GetColor, ccd_SetColor

PixelFormat

typedef enum{ FormatRGB32 = 0, FormatY800, FormatY444, FormatY422, FormatY411,} PixelFormat;

関連

SDK_CCD_INFO 構造体の SDK_SUPPORTED_STREAM 型構造体 ﾒﾝﾊﾞｰ SupportedStream のﾒﾝﾊﾞｰ SupportedFormat は、上記の enum 構造体の番号でｱｻｲﾝすることができます。

CompressorFormat

typedef enum
{ SetByName = 0, NoCompressor, MSVideo, Indeo, Cinepak, DVVideoEncode,} CompressorFormat;

関連

ccd_SetMovieCompressor

Camera Control Flag

```
#define fCameraControlAuto          0X0001L
#define fCameraControlManual        0X0002L
#define fCameraControlOnePush       0X0004L
#define fCameraControlOn            0X0008L
#define fCameraControlOff           0X0010L

#define fFeatureControlAbsolute     0X0000L
#define fFeatureControlRelative     0X0010L

#define MAX_NUMBER_OF_CAMERAS       20
```

注意

ccd_GetPropertyRange, ccd_GetProperty, ccd_SetProperty, ccd_ShowPropertyPage

3-3. Win32 コンソールサンプルプログラム

(3-3-1) プログラム作成の準備

Microsoft Visual C++ 6.0 を起動します。「ﾌｧｲﾙ」ﾒﾆｭｰの新規作成を選択し、新規作成ﾀﾞｲｱﾛｸﾞの「ﾌﾟﾛｼﾞｪｸﾄ」ﾀｸﾞを開きます。「Win32 Console Application」を選択し、ﾌﾟﾛｼﾞｪｸﾄ名を入力後、「OK」ﾎﾞﾀﾝをｸﾘｯｸします。「空のﾌﾟﾛｼﾞｪｸﾄ」を選択し、「終了」をｸﾘｯｸします。
再度、「ﾌｧｲﾙ」ﾒﾆｭｰの新規作成を選択し、新規作成ﾀﾞｲｱﾛｸﾞの「ﾌｧｲﾙ」ﾀｸﾞを開きます。「C++ ｿｰｽﾌｧｲﾙ」を選択し、ﾌｧｲﾙ名入力後「OK」をｸﾘｯｸします。
製品添付の「RS1394CCD SDK CD-ROM」の「ForVC\Include」ﾌｫﾙﾀﾞにある“RsCCD.h”と“RsCCD_i.c”をﾌﾟﾛｼﾞｪｸﾄのﾌｫﾙﾀﾞにｺﾋﾟｰし、ｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑの先頭でｲﾝｸﾙｰﾄﾞします。

(3-3-2) サンプルプログラム

■ Previewｻﾝﾌﾟﾙ：CD-ROM\ForVc\ConApp\Preview\Preview.cpp

CCDｶﾒﾗのﾌﾟﾚﾋﾞｭｰを行うｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑです。接続されているCCDｶﾒﾗをﾌﾟﾚﾋﾞｭｰ状態にし、何かｷｰ入力があるとﾌﾟﾚﾋﾞｭｰを停止しﾌﾟﾛｸﾞﾗﾑを終了します。

```
#include <windows.h>
#include <stdio.h>
#include "RsCCD_i.c"
#include "RsCCD.h"

#define SAFE_RELEASE(pUnk) if (pUnk)\
{\
    pUnk->Release();\
    pUnk = NULL;\
}

IRsCcdSdk *rsk = 0;                                  ｲﾝﾀｰﾌｪｲｽﾎﾟｲﾝﾀ宣言
SDK_CAM_INFO StreamInfo[MAX_NUMBER_OF_CAMERAS];      ｶﾒﾗ情報が格納される構造体

void main(void)
{
    HRESULT hr;
    ULONG NumOfCamera = 0;                           ｶﾒﾗの台数
    int iCam;                                        ｶﾒﾗ番号

    CoInitialize( NULL );                            COM ﾗｲﾌﾞﾗﾘを初期化
    hr = CoCreateInstance( CLSID_RsCcdSdk, NULL,     CCDSdk COM ｵﾌﾞｼﾞｪｸﾄを作成しｲﾝﾀｰﾌｪｰｽ取得
                           CLSCTX_INPROC,
                           IID_IRsCcdSdk, (void**) &rsk );
    rsk->ccd_Initialize();                           CCDSdk COM の初期化
    hr = rsk->ccd_EnumCamera(StreamInfo, &NumOfCamera);  CCD ｶﾒﾗを列挙しｶﾒﾗ情報を取得
    iCam = NumOfCamera -1;                           最後に見つけたｶﾒﾗ番号をｾｯﾄ
    hr = rsk->ccd_PreviewStart( iCam, NULL );        ﾌﾟﾚﾋﾞｭｰ開始
    printf("Press Enter key to quit.");getchar();
    rsk->ccd_PreviewStop( iCam );                    ﾌﾟﾚﾋﾞｭｰ停止
    rsk->ccd_FreeCamera();                           ccd_EnumCamera で確保したｶﾒﾗﾘｿｰｽを解放
    rsk->ccd_Uninitialize();                         CCDSdk COM の終了処理
    SAFE_RELEASE(rsk)                                CCDSdk COM ｵﾌﾞｼﾞｪｸﾄの解放
    CoUninitialize();
}
```

Control ｻﾝﾌﾟﾙ：CD-ROM\ForVc\ConApp\Preview\Control.cpp

CCD ｶﾒﾗのﾌﾟﾚﾋﾞｭｰを行い、ZOOM/FOCUS のｶﾒﾗｺﾝﾄﾛｰﾙと WhiteBalance の OnePush 自動調整を行うｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑです。”T “ or ” W “ ｷｰ入力により Zoom ｺﾝﾄﾛｰﾙ、 ”F ” or “ N “ ｷｰ入力により Focus ｺﾝﾄﾛｰﾙを行います。 ”0 ” ｷｰ入力により WhiteBalance の OnePush 自動調整を行います。

```
#include <windows.h>
#include <stdio.h>
#include <conio.h>

#include "RsCCD_i.c"
#include "RsCCD.h"

IRsCcdSdk *rsk = 0;                                  CCDSdk COM ｲﾝﾀｰﾌｪｲｽﾎﾟｲﾝﾀ
SDK_CAM_INFO StreamInfo[MAX_NUMBER_OF_CAMERAS];      ｶﾒﾗ情報が格納される構造体

#define SAFE_RELEASE(pUnk) if (pUnk)\
{\
    pUnk->Release();\
    pUnk = NULL;\
}

void main(void)
{
    HRESULT hr;                                      関数戻り値
    ULONG   NumOfCamera = 0;                         ｶﾒﾗ台数
    int       iCam;                                  ｶﾒﾗ番号
    LONG    uiPos;                                   ﾌﾟﾛﾊﾟﾃｨの値
    CCD_PROPERTYVALUE CcdProp;                       ﾌﾟﾛﾊﾟﾃｨﾚﾝｼﾞ構造体ｵﾌﾞｼﾞｪｸﾄ
    int      ch;
    char     szDisp[256] ;                           ﾒｯｾｰｼﾞ表示用
    int       fMode = 0;

    CoInitialize( NULL );                            COM ﾗｲﾌﾞﾗﾘを初期化
    hr = CoCreateInstance( CLSID_RsCcdSdk, NULL,     CCDSdk COM ｵﾌﾞｼﾞｪｸﾄを作成しｲﾝﾀｰ
            CLSCTX_INPROC, IID_IRsCcdSdk, (void**) &rsk );   ﾌｪｰｽﾎﾟｲﾝﾀ取得
    rsk->ccd_Initialize();                           CCDSdk COM の初期化
    hr = rsk->ccd_EnumCamera(StreamInfo, &NumOfCamera);   CCD ｶﾒﾗを列挙しｶﾒﾗ情報を取得
    if( NOERROR != hr || NumOfCamera == 0 )
    {
        rsk->ccd_Uninitialize();                     CCDSdk COM の終了処理
        SAFE_RELEASE(rsk)                            CCDSdk COM ｵﾌﾞｼﾞｪｸﾄの解放
        return;
    }

    iCam = NumOfCamera -1;                           最後に見つけたｶﾒﾗ番号をｾｯﾄ
    rsk->ccd_PreviewStart( iCam, NULL );             ﾌﾟﾚﾋﾞｭｰ開始
    do
    {
        printf( "\nT or W : Zoom ｺﾝﾄﾛｰﾙ。F or N : Focus ｺﾝﾄﾛｰﾙ。
                    0 : UB OnePush。'Y'で終了:\n" );
        ch = _getch(); ch = toupper( ch );
        switch(ch)
        {
        case 'T':                                    Zoom-
            rsk->ccd_GetPropertyRange( iCam, Zoom, &CcdProp );   最小、最大値など取得
            rsk->ccd_GetProperty( iCam, Zoom, &uiPos, fMode );   現在の値を取得
            uiPos = uiPos - (CcdProp.Max-CcdProp.Min) / 20;      1/20 ﾚﾝｼﾞで調整
            if(uiPos <= CcdProp.Min)
                uiPos = CcdProp.Min;
            rsk->ccd_SetProperty( iCam, Zoom, uiPos, fMode );    新しい値をｾｯﾄ
```

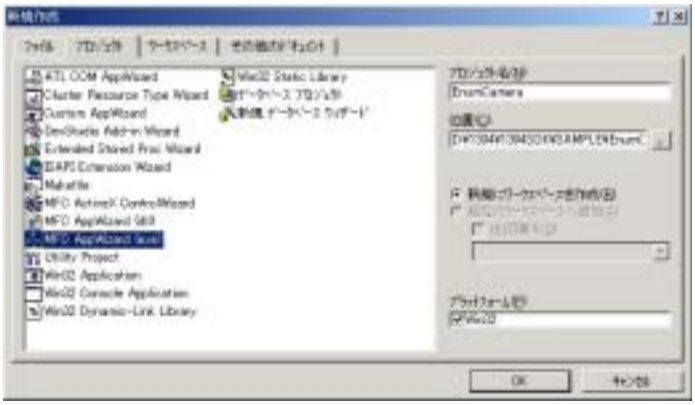
```
            sprintf(szDisp,"Zoom - %d (Min%d - Max%d)\n", uiPos,
                                        CcdProp.Min, CcdProp.Max);
            printf(szDisp);
            break;
      ----- 途中省略 -----
        case 'O':                                                   One push for UB
            fMode = fCameraControlOnePush;                          OnePush ﾓｰﾄﾞｾｯﾄ
            rsk->ccd_GetColor( iCam, UB, &uiPos, fMode );           WhiteBalance OnePush 自動調整
            rsk->ccd_SetColor( iCam, UB, uiPos, fMode );
            rsk->ccd_GetColor( iCam, VR, &uiPos, fMode );           WhiteBalance OnePush 自動調整
            rsk->ccd_SetColor( iCam, UB, uiPos, fMode );
            break;
        }
    } while( ch != 'Y' );

    rsk->ccd_PreviewStop( iCam );                                   ﾌﾟﾚﾋﾞｭｰ停止
    rsk->ccd_FreeCamera();                                          ｶﾒﾗﾘｿｰｽを解放
    rsk->ccd_Uninitialize();                                        CCDSdk COM の終了処理
    SAFE_RELEASE(rsk)                                               CCDSdk COM ｵﾌﾞｼﾞｪｸﾄの解放
    return;
}
```

3-4. MFC サンプルプログラム

(3-4-1) プログラム作成の準備

STEP.1

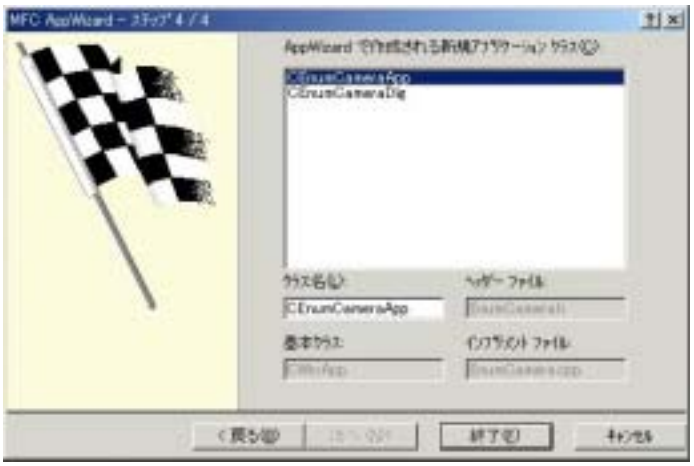
Microsoft Visual C++ 6.0 を起動します。新しいﾌﾟﾛｼﾞｪｸﾄを作成します。「ﾌｧｲﾙ」ﾒﾆｭｰの「新規作成」を選択し、「ﾌﾟﾛｼﾞｪｸﾄ」ﾀｸﾞを開き「MFC AppWizard(exe)」を指定しﾌﾟﾛｼﾞｪｸﾄ名を入力します。「OK」ﾎﾞﾀﾝをｸﾘｯｸして、次に進みます。



作成するｱﾌﾟﾘｹｰｼｮﾝの種類に「ﾀﾞｲｱﾛｸﾞﾍﾞｰｽ」を選択し、後はﾃﾞﾌｫﾙﾄ設定のままで次に進みます。最後右のｳｨｻﾞｰﾄﾞ画面で「終了」をｸﾘｯｸします。

（注記）

上記は EnumCamera ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑの場合を例に取って説明しています。ｱﾌﾟﾘｹｰｼｮﾝの種類等は特に規定しません。



ｱﾌﾟﾘｹｰｼｮﾝから CCD SDK COM のｲﾝﾀｰﾌｪｲｽを使用できるようにするためには、CCD SDK COM の初期化処理が必要です。このﾌﾟﾛｸﾞﾗﾑ例では CEnumCameraApp ｸﾗｽ、もしくは CEnumCameraDlg ｸﾗｽに記述できます。ｱﾌﾟﾘｹｰｼｮﾝの終了時には、CCD SDK COM の終了処理が必要です。

以下は、CEnumCameraApp ｸﾗｽに記述した時の例になります。

EnumCamera ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑの詳細については、CD-ROM\ForVC\MfcApp\EnumCamera を参照して下さい。

STEP.2

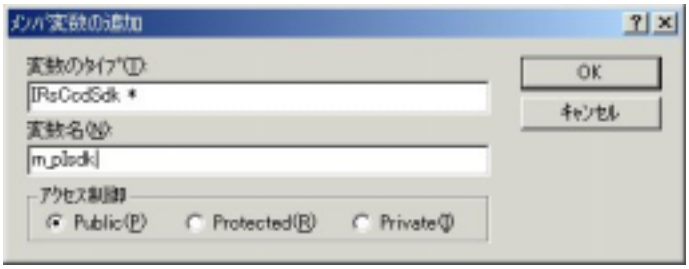
CCD SDK CD-ROM の「ForVC\Include」ﾌｫﾙﾀﾞにある「RSCCD_i.c」と「RSCCD.h」をﾌﾟﾛｼﾞｪｸﾄﾌｫﾙﾀﾞに手動でｺﾋﾟｰします。

EnumCamera.h を開き、「RSCCD_i.c」と「RSCCD.h」をｲﾝｸﾙｰﾄﾞします。

```
#include "resource.h"                    // メイン シンボル

#include "RsCCD_i.c"
#include "RsCCD.h"


// CEnumCameraApp:
class CEnumCameraApp : public CWinApp
```

STEP.3

ClassView の CEnumCameraApp を右ｸﾘｯｸし、CCD SDK COM へのｲﾝﾀｰﾌｪｲｽﾎﾟｲﾝﾀを定義します。



STEP.4

EnumCamera.cpp を開き、右のように CCD SDK COM のﾛｰﾄﾞ処理を記述します。

```
// CEnumCameraApp クラスの構築

CEnumCameraApp::CEnumCameraApp()
{
    // ここに InitInstance 中の重要な初期化処理を記述。

    // COM ﾗｲﾌﾞﾗﾘの初期化
    CoInitialize( NULL );

    // CCD Sdk COM ｵﾌﾞｼﾞｪｸﾄを作成しｲﾝﾀｰﾌｪｰｽﾎﾟｲﾝﾀ取得
    HRESULT hr = CoCreateInstance(
                 CLSID_RsCcdSdk,
                 NULL,
                 CLSCTX_INPROC,
                 IID_IRsCcdSdk,
                 reinterpret_cast<PVOID *>(&m_pIsdk) );
    if( NOERROR != hr ){
        AfxMessageBox("CoCreateInstance Error");
    }
}
```

STEP.5

同じ EnumCamera.cpp のﾀﾞｲｱﾛｸﾞが終了した部分に CCD SDK COM のｱﾝﾛｰﾄﾞ処理を記述します。

```
BOOL CEnumCameraApp::InitInstance()
{
    ・・・・・途中省略・・・・・
    CEnumCameraDlg dlg;
    m_pMainWnd = &dlg;
    int nResponse = dlg.DoModal();
    if (nResponse == IDOK)
    {
        // TODO: ダイアログが <OK> で消された時のコード
        //       を記述してください。
    }
    else if (nResponse == IDCANCEL)
    {
        // TODO: ダイアログが <ｷｬﾝｾﾙ> で消された時の
        //       コードを記述してください。
    }

    // CCD Sdk COM ｵﾌﾞｼﾞｪｸﾄのﾘﾘｰｽ
    if (m_pIsdk)
    {
        m_pIsdk->Release();
        m_pIsdk = NULL;
    }
    // COM ﾗｲﾌﾞﾗﾘの終了処置
    CoUninitialize();

    return FALSE;
}
```

次に、ﾀﾞｲｱﾛｸﾞを表示して「OK」ﾎﾞﾀﾝを押したらCCDｶﾒﾗのﾌﾟﾚﾋﾞｭｰを行い、「ｷｬﾝｾﾙ」ﾎﾞﾀﾝによりﾌﾟﾚﾋﾞｭｰを停止しﾌﾟﾛｸﾞﾗﾑを終了するｱﾌﾟﾘｹｰｼｮﾝの作成手順を説明します。

STEP.6

ｱﾌﾟﾘｹｰｼｮﾝの初期化処理として、CenumCameraDlgｸﾗｽのOnInitDialog()に右のｺｰﾄﾞを追加します。

```
BOOL CEnumCameraDlg::OnInitDialog()
{
    最初からここまではそのまま
    // TODO: 特別な初期化を行う時はこの場所に追加。

    // CCD Sdk COMの初期化処理
    m_pIsdk->ccd_Initialize();
    return TRUE;
}
```

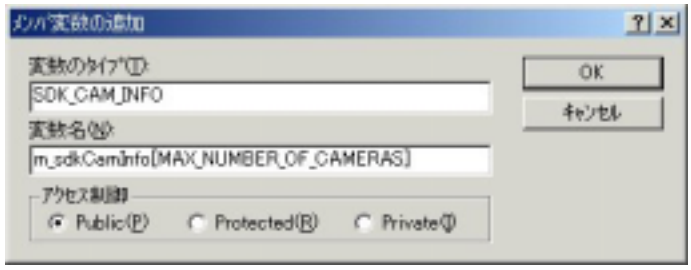
ｱﾌﾟﾘｹｰｼｮﾝの終了処理として、OnCancel()に右のｺｰﾄﾞを追加します。

```
void CEnumCameraDlg::OnCancel()
{
    // TODO: この位置に特別な後処理を追加してください。

    // CCD Sdk COMの終了処理
    m_pIsdk->ccd_Uninitialize();

    CDialog::OnCancel();
}
```

ここまでで、一度ﾋﾞﾙﾄﾞして下さい。次に、「OK」ﾎﾞﾀﾝが押されたらﾌﾟﾚﾋﾞｭｰする処理に取りかかります。

STEP.7

ClassViewのCEnumCameraDlgを右ｸﾘｯｸし、SDK_CAM_INFO型の構造体配列sdkCamInfo[MAX_NUMBER_OF_CAMERAS]を定義します。
同様に、ULONG m_NumOfCameraを定義。



STEP.8

「OK」ﾎﾞﾀﾝが押されたらﾌﾟﾚﾋﾞｭｰする処理を追加します。

CDialog::OnOK();をｺﾒﾝﾄしないとｱﾌﾟﾘｹｰｼｮﾝが終了してしまいます。

```
void CEnumCameraDlg::OnOK()
{
    HRESULT hr = m_pIsdk->ccd_EnumCamera
                        (m_sdkCamInfo, &m_NumOfCamera);
    if( NOERROR != hr  || 0 == m_NumOfCamera ){
       AfxMessageBox("Can't find ccd camera.");
       return;
    }
    hr = m_pIsdk->ccd_PreviewStart
                          (m_NumOfCamera-1, NULL);
    // CDialog::OnOK();
}
```

STEP.9

「ｷｬﾝｾﾙ」ﾎﾞﾀﾝが押されたらﾌﾟﾚﾋﾞｭｰを停止する処理を追加します。

```
void CEnumCameraDlg::OnCancel()
{
    // ﾌﾟﾚﾋﾞｭｰ停止
    m_pIsdk->ccd_PreviewStop( m_NumOfCamera-1 );
    // ccd_EnumCameraで確保したｶﾒﾗﾘｿｰｽを解放
    m_pIsdk->ccd_FreeCamera();

    // CCD Sdk COMの終了処理
    m_pIsdk->ccd_Uninitialize();
    CDialog::OnCancel();
}
```

STEP.10

EnumCameraDlg.h に「RsCCD.h」をｲﾝｸﾙｰﾄﾞします。

```
// EnumCameraDlg.h : ヘッダー ファイル
//

#include  "RsCCD.h"

/////////////////////////////////////////////////////////////
// CEnumCameraDlg ダイアログ

class CEnumCameraDlg : public CDialog
{
```

STEP.11

ﾋﾞﾙﾄﾞして実行すると右のﾀﾞｲｱﾛｸﾞが表示され、「OK」ﾎﾞﾀﾝをｸﾘｯｸすると背後のﾌﾟﾚﾋﾞｭｰｳｨﾝﾄﾞｳが表示されます。

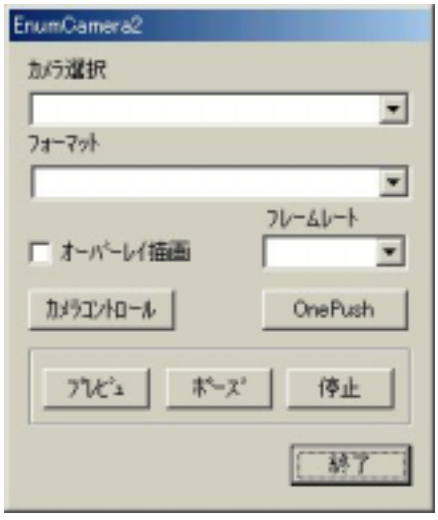
(3-4-2) MFC サンプルプログラム解説

EnumCamera2 ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM\ForVC\MfcApp\EnumCamera2 を参照して下さい。

EnumCamera2 ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑは接続されている CCD ｶﾒﾗをｽｷｬﾝし、ｶﾒﾗ選択ｺﾝﾎﾞﾎﾞｯｸｽに CCD ｶﾒﾗのﾕﾆｰｸな識別名称を、ﾌｫｰﾏｯﾄ及びﾌﾚｰﾑﾚｰﾄｺﾝﾎﾞﾎﾞｯｸｽに選択されているｶﾒﾗがｻﾎﾟｰﾄしているﾌｫｰﾏｯﾄ・ﾌﾚｰﾑﾚｰﾄをｾｯﾄします。

また　ｶﾒﾗの取り外しを行ったり、新たに接続した時に送られてくる、WM_DEVICECHANGE ｶﾒﾗｲﾍﾞﾝﾄに対する応答処理を行います。ﾌﾟﾚﾋﾞｭｰﾎﾞﾀﾝにより、選択されている CCD ｶﾒﾗのﾌﾟﾚﾋﾞｭｰを行います。

以下に、EnumCamera2 ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑの骨格について解説します。ﾌﾟﾛｸﾞﾗﾑの詳細についてはｿｰｽﾞｺｰﾄﾞを参照して下さい。



ﾀﾞｲｱﾛｸﾞ画面の準備

ﾀﾞｲｱﾛｸﾞに各ｺﾝﾄﾛｰﾙを配置します。この時、ｺﾝﾎﾞﾎﾞｯｸｽのﾌﾟﾛﾊﾟﾃｨの「ｿｰﾄ」を行わないようにすることが重要です。

ccd_EnumCamera により SDK_CAM_INFO 型配列の各要素に接続されている全てのｶﾒﾗ情報がｾｯﾄされ、各配列の要素順にｺﾝﾎﾞﾎﾞｯｸｽのﾘｽﾄﾎﾞｯｸｽに AddString() します。GetCurSel()はｺﾝﾎﾞﾎﾞｯｸｽのﾘｽﾄﾎﾞｯｸｽで現在選択されている項目の 0 から始まるｲﾝﾃﾞｯｸｽ番号を返します。この時返されるｲﾝﾃﾞｯｸｽ番号を、ccd_XXX()で指定するｶﾒﾗｲﾝﾃﾞｯｸｽ番号としてそのまま使用できるようにします。その他、ccd_SetStream, ccd_SetFrameRate の引数、ｽﾄﾘｰﾑｲﾝﾃﾞｯｸｽ番号・ﾌﾚｰﾑﾚｰﾄｲﾝﾃﾞｯｸｽ番号についても同様です。

ｱﾌﾟﾘｹｰｼｮﾝの初期化処理

本ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑはﾀﾞｲｱﾛｸﾞﾍﾞｰｽですので、OnInitDialog()で必要となるｲﾆｼｬﾗｲｽﾞを行います。

```
BOOL CEnumCamera2Dlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    m_pApp = (CEnumCamera2App *)AfxGetApp();
    m_pApp->m_pIsdk->ccd_Initialize();

    hr = m_pApp->m_pIsdk->ccd_RegCameraEvent( m_hWnd );
    if( NOERROR != hr ){
        AfxMessageBox("ccd_RegCameraEvent Failed");
        return TRUE;
    }

    m_pApp->m_pIsdk->ccd_EnumCamera
                            (m_sdkCamInfo, &m_NumOfCamera);
    if( 0 == m_NumOfCamera ){
        AfxMessageBox("Can't find CCD camera.");
        return TRUE;
    }

    Refresh_cboCamera(m_CurrentCameraId = 0);
    m_pApp->m_pIsdk->ccd_GetCurrentFormat(m_CurrentCameraId,
                &m_CurrentStreamId, &m_CurrentFrameRateId);

    Refresh_cboStream(m_CurrentCameraId, m_CurrentStreamId);
    Refresh_cboFrameRate(m_CurrentCameraId, m_CurrentStreamId);
    return TRUE;
}
```

CEnumCamera2App *m_pApp
CCD Sdk COM の初期化処理

WM_DEVICECHANGE ｲﾍﾞﾝﾄが通知されるよう登録

ｱﾌﾟﾘ起動時接続されているｶﾒﾗの情報取得
SDK_CAM_INFO m_sdkCamInfo[MAX_NUMBER_OF_CAMERAS] は、class CEnumCamera2Dlg のﾒﾝﾊﾞｰ変数

最初のｶﾒﾗをｶﾚﾝﾄに設定しｶﾒﾗｺﾝﾎﾞ初期化
選択したｶﾒﾗより現在動作しているｽﾄﾘｰﾑ・ﾌﾚｰﾑﾚｰﾄを取得

ｽﾄﾘｰﾑｺﾝﾎﾞ初期化
ﾌﾚｰﾑﾚｰﾄｺﾝﾎﾞ初期化

ｱﾌﾟﾘｹｰｼｮﾝの終了処理

「ｷｬﾝｾﾙ」ﾎﾞﾀﾝが押されたらｱﾌﾟﾘｹｰｼｮﾝを終了するようにするため、OnCancel()で必要となる終了処理を行います。

```
void CEnumCamera2Dlg::OnCancel()
{
    m_pApp->m_pIsdk->ccd_PreviewStop(m_CurrentCameraId);
    m_pApp->m_pIsdk->ccd_FreeCamera();
    m_pApp->m_pIsdk->ccd_Uninitialize();
    CDialog::OnCancel();
}
```

ﾌﾟﾚﾋﾞｭｰ停止
ccd_EnumCamera で確保したｶﾒﾗﾘｿｰｽを解放
CCD Sdk COM の終了処理

■ WM_DEVICECHANGE ｶﾒﾗｲﾍﾞﾝﾄに対する応答処理

WM_DEVICECHANGE ｶﾒﾗｲﾍﾞﾝﾄをｱﾌﾟﾘｹｰｼｮﾝが受け取ることができるようにするためには、ｳｨﾝﾄﾞｳの登録処理 ccd_RegCameraEvent()が必要です。次に、CEnumCamera2Dlg ｸﾗｽに手動で、ﾒｯｾｰｼﾞﾏｯﾌﾟを手動で定義します。定義する位置は、//}}AFX_MSG と DECLARE_MESSAGE_MAP()の間で行います。

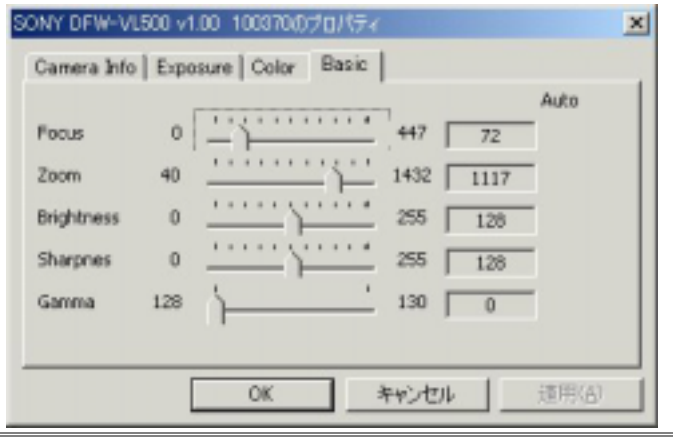
```
class CEnumCamera2Dlg : public Cdialog                    ← EnumCamera2Dlg.h
{
    // 生成されたメッセージ マップ関数
    //{{AFX_MSG(CEnumCamera2Dlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    virtual void OnCancel();
    //}}AFX_MSG

    afx_msg BOOL OnDeviceChange( UINT nEventType, DWORD dwData );  //}}AFX_MSG の後に WM_DEVICECHANGE ｲﾍﾞﾝﾄﾊﾝﾄﾞﾗを手動追加する
    DECLARE_MESSAGE_MAP()
};

BEGIN_MESSAGE_MAP(CEnumCamera2Dlg, CDialog)
        //{{AFX_MSG_MAP(CEnumCamera2Dlg)
        ON_WM_PAINT()                                     ← EnumCamera2Dlg.cpp
        ON_WM_QUERYDRAGICON()
        //}}AFX_MSG_MAP
        ON_WM_DEVICECHANGE()
END_MESSAGE_MAP()
                                                          WM_DEVICECHANGE ｲﾍﾞﾝﾄﾊﾝﾄﾞﾗを手動追加
```

最後にﾒｯｾｰｼﾞﾊﾝﾄﾞﾗ OnDeviceChange()を記述します。nEventType には、DBT_DEVICEREMOVECOMPLETE, DBT_DEVICEARRIVAL 等のｲﾍﾞﾝﾄの内容を示すｺｰﾄﾞがｾｯﾄされます。実際の処理の内容はｱﾌﾟﾘｹｰｼｮﾝにより異なりますが、通常は今まで確保されていたｶﾒﾗﾘｿｰｽと情報を ccd_FreeCamera()で破棄し、新たに ccd_EnumCamera()を呼び出し新たにｶﾒﾗﾘｿｰｽと情報を構築し直します。その情報を元に、各ｺﾝﾎﾞﾎﾞｯｸｽのﾘｽﾄ内容も再構築します。

```
OOL CEnumCamera2Dlg::OnDeviceChange( UINT nEventType, DWORD dwData )
{
    switch (nEventType)
    {
    case DBT_DEVICEREMOVECOMPLETE:                        ｶﾒﾗの取り外し
    case DBT_DEVICEARRIVAL:                               ｶﾒﾗの追加

        m_pApp->m_pIsdk->ccd_FreeCamera();                ccd_EnumCamera で確保したｶﾒﾗﾘｿｰｽを解放
        m_pApp->m_pIsdk->ccd_EnumCamera
                              ( m_sdkCamInfo, &m_NumOfCamera );
        if( 0 == m_NumOfCamera ){
            AfxMessageBox("Can't find CCD camera.");
            return TRUE;
        }

        Refresh_cboCamera(m_CurrentCameraId = 0);         以下、OnInitDialog()と同様の処理
        m_pApp->m_pIsdk->ccd_GetCurrentFormat(m_CurrentCameraId,  0 番目のｶﾒﾗをｶﾚﾝﾄｾﾚｸﾄに設定しｶﾒﾗｺﾝﾎﾞ初期化
                    &m_CurrentStreamId, &m_CurrentFrameRateId);
        Refresh_cboStream(m_CurrentCameraId, m_CurrentStreamId);   ｽﾄﾘｰﾑｺﾝﾎﾞ初期化
        Refresh_cboFrameRate(m_CurrentCameraId,           ﾌﾚｰﾑﾚｰﾄｺﾝﾎﾞ初期化
                                        m_CurrentStreamId);
        break;
    }
    return TRUE;
}
```

ｶﾒﾗﾌﾟﾛﾊﾟﾃｨﾍﾟｰｼﾞを表示

「ｶﾒﾗｺﾝﾄﾛｰﾙ」ﾎﾞﾀﾝが押されたらｽﾞｰﾑ・ﾌｫｰｶｽ等のﾍﾞｰｼｯｸﾌﾟﾛﾊﾟﾃｨ、ﾎﾜｲﾄﾊﾞﾗﾝｽ等のｶﾗｰﾌﾟﾛﾊﾟﾃｨを制御できるｶﾒﾗﾌﾟﾛﾊﾟﾃｨﾍﾟｰｼﾞを表示できるようにします。




```
void CEnumCamera2Dlg::OnBtnCameraCtrl()
{
     hr = m_pApp->m_pIsdk->ccd_ShowPropertyPage( m_CurrentCameraId );
     if( NOERROR != hr ){
          AfxMessageBox("ccd_ShowPropertyPage Failed");
     }
}
```

WhiteBalance OnePush 自動調整機能の作成

「OnePush」ﾎﾞﾀﾝが押されたら、UB/VR ﾎﾜｲﾄﾊﾞﾗﾝｽの OnePush 自動調整を行います。

```
void CEnumCamera2Dlg::OnBtnOnePush()
{
     LONG ColorValue;
     HRESULT  hr;

     hr = m_pApp->m_pIsdk->ccd_GetColo
                              (m_CurrentCameraId, UB, &ColorValue, 0);   現在の WhiteBalance B-gain(Blue)を取得

     hr = m_pApp->m_pIsdk->ccd_SetColor
          (m_CurrentCameraId, UB, ColorValue, fCameraControlOnePush);    B-gain の OnePush 自動調整実行

     hr = m_pApp->m_pIsdk->ccd_GetColor
                              (m_CurrentCameraId, VR, &ColorValue, 0);   現在の WhiteBalance R-gain(Red)を取得

     hr = m_pApp->m_pIsdk->ccd_SetColor
          (m_CurrentCameraId, VR, ColorValue, fCameraControlOnePush);    R-gain の OnePush 自動調整実行
}
```

Preview/Pause/Stop 処理の作成

「Preview/Pause/Stop」ﾎﾞﾀﾝが時の処理は下記の通りです。

```
void CEnumCamera2Dlg::OnBtnPreview()
{
	if( BST_CHECKED == m_chkOverlay.GetCheck() ){
		m_pApp->m_pIsdk->ccd_SetOverlay( m_CurrentCameraId, TRUE );		ｵｰﾊﾞｰﾚｲ設定有効
	}else{
		m_pApp->m_pIsdk->ccd_SetOverlay( m_CurrentCameraId, FALSE );		ｵｰﾊﾞｰﾚｲ設定無効
	}

	HRESULT hr = m_pApp->m_pIsdk->ccd_PreviewStart(m_CurrentCameraId, NULL);
	if( NOERROR != hr ){
		AfxMessageBox("ccd_PreviewStart Failed");
	}
}
```

```
void CEnumCamera2Dlg::OnBtnPause()
{
	HRESULT hr = m_pApp->m_pIsdk->ccd_PreviewPause( m_CurrentCameraId );
}
```

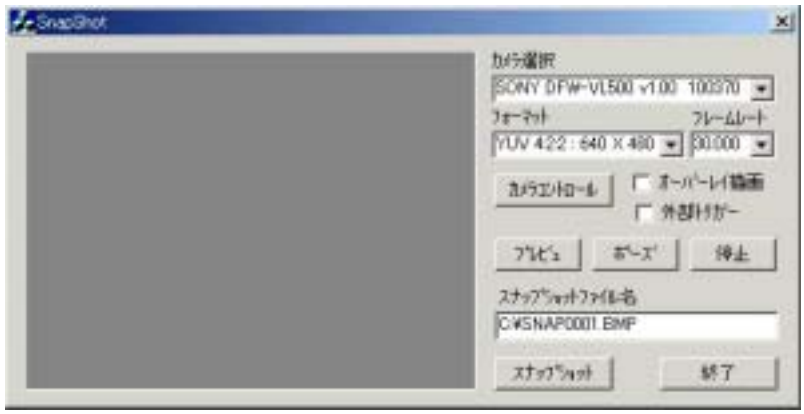
```
void CEnumCamera2Dlg::OnBtnStop()
{
	HRESULT hr = m_pApp->m_pIsdk->ccd_PreviewStop( m_CurrentCameraId );
}
```

SnapShot ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM\ForVC\MfcApp\SnapShot を参照して下さい。

EnumCamera2 ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑとの相違点は静止画のｽﾅｯﾌﾟｼｮｯﾄ機能です。

ﾌﾟﾚﾋﾞｭｰした状態で「ｽﾅｯﾌﾟｼｮｯﾄ」ﾎﾞﾀﾝをｸﾘｯｸすると指定のﾌｧｲﾙ名でﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙが保存され、ﾋﾟｸﾁｬｰﾎﾞｯｸｽにﾋﾞｯﾄﾏｯﾌﾟ画像が表示されます。

ｽﾅｯﾌﾟｼｮｯﾄ実行前にﾌｧｲﾙ名を「C:\SNAP001.BMP」のﾌﾙﾊﾟｽ形式で指定します。

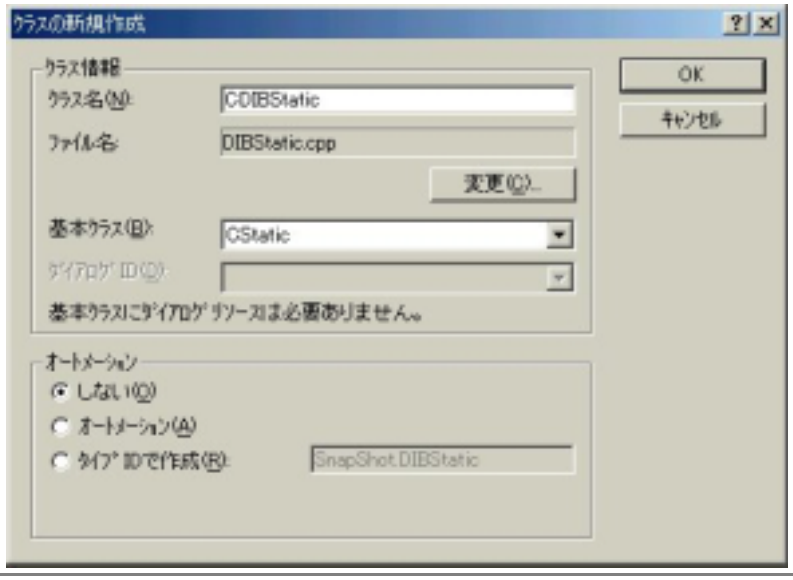


「ﾌﾟﾚﾋﾞｭｰ」ﾎﾞﾀﾝｸﾘｯｸ時の処理

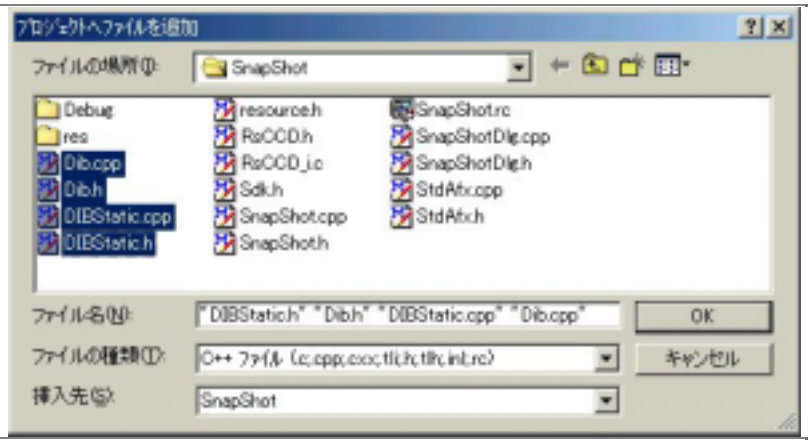
```
void CSnapShotDlg::OnBtnPreview()
{
     if( BST_CHECKED == m_chkOverlay.GetCheck() ){
          m_pApp->m_pIsdk->ccd_SetOverlay( m_CurrentCameraId, TRUE );          ｵｰﾊﾞｰﾚｲ設定
     }else{
          m_pApp->m_pIsdk->ccd_SetOverlay( m_CurrentCameraId, FALSE );
     }

     if (BST_CHECKED == m_chkTrigger.GetCheck()){
          m_pApp->m_pIsdk->ccd_SetTrigger( m_CurrentCameraId, TRUE );          外部ﾄﾘｶﾞｰ設定
     }else{
          m_pApp->m_pIsdk->ccd_SetTrigger( m_CurrentCameraId, FALSE );
     }

     HRESULT hr = m_pApp->m_pIsdk->ccd_PreviewStart(m_CurrentCameraId, NULL);
     if( NOERROR != hr ){
          AfxMessageBox("ccd_PreviewStart Failed");
     }
}
```

ﾋﾟｸﾁｬｰﾎﾞｯｸｽへのﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙ貼り付け方法（参考例として）

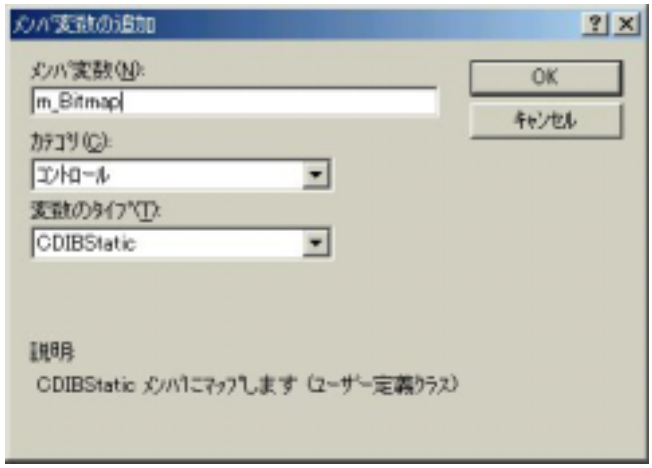
ClassWizard を起動し「ｸﾗｽ情報」ﾀﾞを開きます。「ｸﾗｽの追加」より「新規」を選択します。
基本ｸﾗｽに「CStatic」を選択し、ｸﾗｽ名に「CDIBStatic」を入力し、「OK」ﾎﾞﾀﾝをｸﾘｯｸします。



CD-ROM\ForVC\CDIBStatic ﾌｫﾙﾀﾞより、「DIBStatic.cpp/DIBStatic.h」と「DIB.cpp/DIB.h」の４つのﾌｧｲﾙをﾌﾟﾛｼﾞｪｸﾄにｺﾋﾟｰします。
「ﾌﾟﾛｼﾞｪｸﾄ」ﾒﾆｭｰより、「ﾌﾟﾛｼﾞｪｸﾄへ追加」の「ﾌｧｲﾙ」を開きます。右図のようにｺﾋﾟｰしたﾌｧｲﾙを全て追加します。



ClassView より「CSnapShotDlg」の右ｸﾘｯｸより、ﾒﾝﾊﾞｰ変数の追加を開きます。ｶﾃｺﾞﾘに「ｺﾝﾄﾛｰﾙ」を選択し、変数のﾀｲﾌﾟに前で作成したﾕｰｻﾞ定義ｸﾗｽ「CDIBStatic」を選択します。ﾒﾝﾊﾞ変数「m_Bitmap」を入力します。



上記の処理を手動で行う場合は、右の「SnapShotDlg.h」のｸﾗｽに
CDIBStatic　m_Bitmap;
を追加し、

「SnapShotDlg.cpp」の
DoDataExchange に、
DDX_Control(pDX, IDC_PicSnapShop, m_Bitmap);
を追加します。

```
class CSnapShotDlg : public CDialog
{
// 構築
public:

// ダイアログ データ
	//{{AFX_DATA(CSnapShotDlg)
	enum { IDD = IDD_SNAPSHOT_DIALOG };
	CDIBStatic	m_Bitmap;
	//}}AFX_DATA
};

void CSnapShotDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CSnapShotDlg)
	DDX_Control(pDX, IDC_PicSnapShop, m_Bitmap);
	//}}AFX_DATA_MAP
}
```

「ｽﾅｯﾌﾟｼｮｯﾄ」ﾎﾞﾀﾝｸﾘｯｸ時の処理

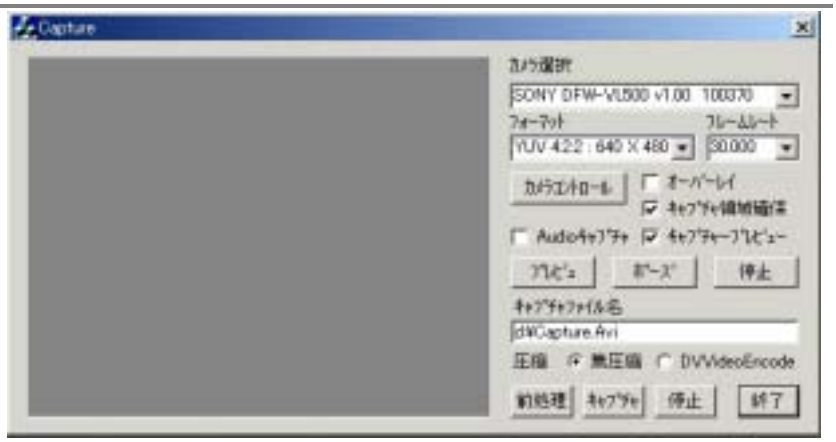
SnapShotDlg.h に DIBStatic.h をｲﾝｸﾙｰﾄﾞし、「ｽﾅｯﾌﾟｼｮｯﾄ」ﾎﾞﾀﾝがｸﾘｯｸされた時の処理を作成します。

```
#include "AFXPRIV.H"                                                    ワイドﾞ文字変換ﾏｸﾛに必要

void CSnapShotDlg::OnBtnSanpShot()
{                                                                       ワイドﾞ文字変換ﾏｸﾛに必要
     USES_CONVERSION;
     WCHAR wcharFileName[ MAX_PATH ];

     UpdateData(true);

     LPTSTR lpszFileName = m_cstrFileName.GetBuffer( MAX_PATH );        ﾌｧｲﾙ名をﾜｲﾄﾞ文字列に変
     wcscpy(wcharFileName, T2W(lpszFileName));                          換。

     HRESULT hr = m_pApp->m_plsdk->ccd_Snapshot(m_CurrentCameraId, wcharFileName );  ｽﾅｯﾌﾟｼｮｯﾄの実行。
     if( NOERROR != hr ){
          AfxMessageBox("ccd_Snapshot Failed");
          m_cstrFileName.ReleaseBuffer();
          return;
     }

     m_Bitmap.LoadDib(lpszFileName);                                    保存したﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙを
     m_cstrFileName.ReleaseBuffer();                                    ﾋﾟｸﾁｬｰﾎﾞｯｸｽにﾛｰﾄﾞ。
}
```

外部ﾄﾘｶﾞｰﾓｰﾄﾞ設定処理

```
void CSnapShotDlg::OnBtnPreview()
{

     if( BST_CHECKED == m_chkOverlay.GetCheck() ){                      ｵｰﾊﾞｰﾚｲ設定
          m_pApp->m_plsdk->ccd_SetOverlay( m_CurrentCameraId, TRUE );
     }else{
          m_pApp->m_plsdk->ccd_SetOverlay( m_CurrentCameraId, FALSE );
     }

     if (BST_CHECKED == m_chkTrigger.GetCheck()){                       外部ﾄﾘｶﾞｰ設定
          m_pApp->m_plsdk->ccd_SetTrigger( m_CurrentCameraId, TRUE );
     }else{
          m_pApp->m_plsdk->ccd_SetTrigger( m_CurrentCameraId, FALSE );
     }

     HRESULT hr = m_pApp->m_plsdk->ccd_PreviewStart(m_CurrentCameraId, NULL);  ﾌﾟﾚﾋﾞｭｰ開始
     if( NOERROR != hr ){
          AfxMessageBox("ccd_PreviewStart Failed");
     }

}
```

Capture ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM\ForVC\MfcApp\Capture を参照して下さい。



EnumCamera2 ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑとの相違点は動画のｷｬﾌﾟﾁｬ機能です。

「ﾌﾟﾚﾋﾞｭｰ」を実行するとﾀﾞｲｱﾛｸﾞ内のﾋﾟｸﾁｬｰﾎﾞｯｸｽに動画が表示されます。「ｷｬﾌﾟﾁｬ」ﾎﾞﾀﾝをｸﾘｯｸする前に保存する動画ﾌｧｲﾙ名を入力し「前処理」ﾎﾞﾀﾝをｸﾘｯｸします。前処理を実行すると「ｷｬﾌﾟﾁｬ領域確保」のﾁｪｯｸﾎﾞｯｸｽにﾁｪｯｸを入れている場合は、予めﾊｰﾄﾞﾃﾞｨｽｸに 200M ﾊﾞｲﾄのﾌｧｲﾙ領域が確保されます。ｷｬﾌﾟﾁｬ実行時、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙに一時的に動画が保存されます。ｷｬﾌﾟﾁｬを停止すると、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙからｷｬﾌﾟﾁｬﾌｧｲﾙ名で指定したﾌｧｲﾙに動画ﾃﾞｰﾀがｺﾋﾟｰされます。

ﾋﾟｸﾁｬｰﾎﾞｯｸｽ内にﾌﾟﾚﾋﾞｭｰする

```
void CCaptureDlg::OnBtnPreview()
{
    if( BST_CHECKED == m_chkOverlay.GetCheck() ){
        m_pApp->m_pIsdk->ccd_SetOverlay( m_CurrentCameraId, TRUE );     ｵｰﾊﾞｰﾚｲ設定有効
    }else{
        m_pApp->m_pIsdk->ccd_SetOverlay( m_CurrentCameraId, FALSE );    ｵｰﾊﾞｰﾚｲ設定無効
    }

    HRESULT hr = m_pApp->m_pIsdk->ccd_PreviewStart                      ﾋﾟｸﾁｬｰﾎﾞｯｸｽのｳｨﾝﾄﾞｳﾊﾝﾄﾞﾙを指定
                            (m_CurrentCameraId, m_picPreview.m_hWnd);
    if( NOERROR != hr ){
        AfxMessageBox("ccd_PreviewStart Failed");
    }
}
```

前処理実行

```
#include "AFXPRIV.H"                                                    ﾜｲﾄﾞ文字列ﾏｸﾛのﾍｯﾀﾞｰ

void CCaptureDlg:: OnBtnPreCapture()
{                                                                       ﾜｲﾄﾞ文字列ﾏｸﾛ定義
    USES_CONVERSION;

    WCHAR wcharCaptureFile[ MAX_PATH ];
    WCHAR wcharTempFile[ MAX_PATH ];                                    仮に 200MBytes の予めｱﾛｹｰﾄ
    LONG  lMBytes = 200;                                                ｷｬﾌﾟﾁｬｰを実行する前に、HDD 上に
    HRESULT hr;                                                         動画保存用ﾌｧｲﾙをｱﾛｹｰｼｮﾝ

    UpdateData(true);

    LPTSTR lpszFileName = m_editFileName.GetBuffer( MAX_PATH );         ｴﾃﾞｨｯﾄﾎﾞｯｸｽ入力ｷｬﾌﾟﾁｬﾌｧｲﾙ名
    wcscpy(wcharCaptureFile, T2W(lpszFileName));                        ﾌｧｲﾙ名をﾜｲﾄﾞ文字列に変換
    m_editFileName.ReleaseBuffer();

    wcscpy(wcharTempFile, T2W(_T("c:\\temp.avi")));                     ﾃﾝﾎﾟﾗﾘｷｬﾌﾟﾁｬﾌｧｲﾙ名設定
```

```
    hr = m_pApp->m_pIsdk->ccd_SetCaptureFile( m_CurrentCameraId,
    wcharCaptureFile, true );

    if( BST_CHECKED == m_chkAllocate.GetCheck() ){
        AfxMessageBox("OOMB Allocate...Hit any key to capture");
        hr = m_pApp->m_pIsdk->ccd_SetCaptureTempFile
                ( m_CurrentCameraId, wcharTempFile, true, lMBytes );
    } else {
        hr = m_pApp->m_pIsdk->ccd_SetCaptureTempFile
                ( m_CurrentCameraId, wcharTempFile, false, 0 );
    }

    CButton *radNoComp = (CButton *) GetDlgItem(IDC_RadNoComp);
    if (radNoComp->GetCheck() == 1)
        hr = m_pApp->m_pIsdk->ccd_SetMovieCompressor
                        (m_CurrentCameraId, 0, NULL);
    else
        hr = m_pApp->m_pIsdk->ccd_SetMovieCompressor
                        (m_CurrentCameraId, DVVideoEncode, NULL);

    if( BST_CHECKED == m_chkAudio.GetCheck() ){
        hr = m_pApp->m_pIsdk->ccd_SetAudio(m_CurrentCameraId, true);
    } else {
        hr = m_pApp->m_pIsdk->ccd_SetAudio(m_CurrentCameraId, false);
}
```

ｷｬﾌﾟﾁｬﾃﾝﾎﾟﾗﾘﾌｧｲﾙ設定

圧縮ﾓｰﾄﾞの設定

Audio ｷｬﾌﾟﾁｬの設定

動画ｷｬﾌﾟﾁｬ開始

```
void CCaptureDlg::OnBtnCapture()
{
    HRESULT hr;

    if( BST_CHECKED == m_chkPreview.GetCheck() ){
        hr = m_pApp->m_pIsdk->ccd_CaptureStart
                (m_CurrentCameraId, TRUE, m_picPreview.m_hWnd);
    }else{
        hr = m_pApp->m_pIsdk->ccd_CaptureStart
                (m_CurrentCameraId, FALSE, m_picPreview.m_hWnd);
    }
}
```

ｷｬﾌﾟﾁｬｰ開始

ｷｬﾌﾟﾁｬを停止

ｷｬﾌﾟﾁｬを停止するとﾃﾝﾎﾟﾗﾘに保存した動画ﾃﾞｰﾀのｺﾋﾟｰが開始します。
右のｺﾋﾟｰﾌﾟﾛｸﾞﾚｽ表示ﾀﾞｲｱﾛｸﾞを表示します。

```
void CCaptureDlg::OnBtnCapStop()
{
    m_pApp->m_pIsdk->ccd_CaptureStart( m_CurrentCameraId );

    CCopyDlg dlgCopyProgress;
    dlgCopyProgress.DoModal();
}
```

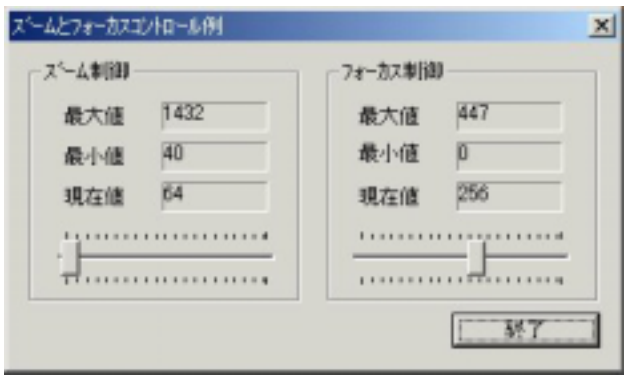
Functions ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM\ForVC\MfcApp\Functions を参照して下さい。

Function はその他の CCD SDK COM のｲﾝﾀｰﾌｪｲｽを説明するｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑになります。ﾒｲﾝﾀﾞｲｱﾛｸﾞは右の画面です。

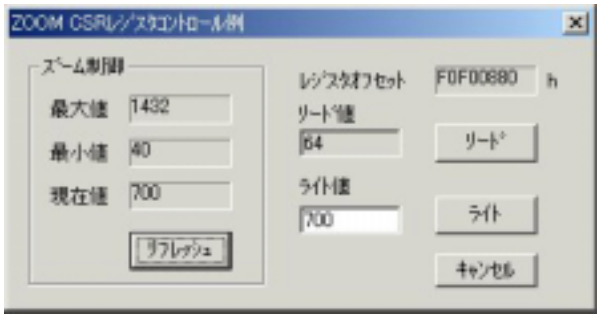
CCD ｶﾒﾗがﾒﾓﾘ記憶機能を持つ場合は、「CH ｾｰﾌﾞ」ﾎﾞﾀﾝをｸﾘｯｸすることにより、ｶﾒﾗの Memory_Channel”1”に現在値が保存され、「CH ﾛｰﾄﾞ」ﾎﾞﾀﾝをｸﾘｯｸすることにより、Memory_Channel”1”に記憶された値をｶﾒﾗに設定します。

ﾒｲﾝﾀﾞｲｱﾛｸﾞでﾌﾟﾚﾋﾞｭｰした状態で「Zoom」ﾎﾞﾀﾝをｸﾘｯｸします。ﾀﾞｲｱﾛｸﾞの初期化処理で、指定のｶﾒﾗよりｽﾞｰﾑとﾌｫｰｶｽのﾌﾟﾛﾊﾟﾃｨを読み出し、各最大・最小調整範囲を表示します。同時に、読み出した値に従ってｽﾗｲﾀﾞｰﾊﾞｰの設定を行います。

ｽﾗｲﾀﾞｰﾊﾞｰをｽﾗｲﾄﾞさせることで、ｽﾞｰﾑとﾌｫｰｶｽの調整を行います。



ﾒｲﾝﾀﾞｲｱﾛｸﾞでﾌﾟﾚﾋﾞｭｰした状態で「CSR」ﾎﾞﾀﾝをｸﾘｯｸします。ｽﾞｰﾑ機能を例にして、CSR ﾚｼﾞｽﾀに直接ﾘｰﾄﾞﾗｲﾄして制御する方法を説明します。



ｽﾞｰﾑﾌﾟﾛﾊﾟﾃｨの取得

```
BOOL CZoomDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    HRESULT hr;
    PROPERTY_RANGEPropValue;
    LONG ZoomValue,FocusValue;
    LONG ScrollRange;

    m_theApp = (CFunctionsApp *)AfxGetApp();                    CWinApp ｵﾌﾞｼﾞｪｸﾄへのﾎﾟｲﾝﾀ取得
    CFunctionsDlg *MainDlg = (CFunctionsDlg *)m_theApp->m_pMainWnd;  ﾒｲﾝﾀﾞｲｱﾛｸﾞのｳｨﾝﾄﾞｳﾊﾝﾄﾞﾙ取得
    m_CameraId = MainDlg->m_CurrentCameraId;                    ﾒｲﾝﾀﾞｲｱﾛｸﾞで択されているｶﾒﾗ取得

    hr = m_theApp->m_pIsdk->ccd_GetPropertyRange                Zoom ﾌﾟﾛﾊﾟﾃｨの調整ﾚﾝｼﾞ取得
                              (m_CameraId, Zoom, &PropValue);
    SetDlgItemInt(IDC_TxtZoomMaxVal, PropValue.Max, true);      最大・最小値表示
    SetDlgItemInt(IDC_TxtZoomMinVal, PropValue.Min, true);

    m_sliderZoom.SetRange(PropValue.Min, PropValue.Max);        Zoom ｽﾗｲﾀﾞｰﾊﾞｰの調整範囲設定
```

```
    ScrollRange = PropValue.Max - PropValue.Min;
    if( ScrollRange < 20 ){
        m_sliderZoom.SetTicFreq( 1 );                          目盛は(最大値-最小値)が 20 以下なら
    }else{                                                     1 刻み
        m_sliderZoom.SetTicFreq( ScrollRange / 20 );
    }

    hr = m_theApp->m_pIsdk->ccd_GetProperty                    ｶﾒﾗより Zoom の現在値を取得
                            (m_CameraId, Zoom, &ZoomValue, 0);
    SetDlgItemInt(IDC_TxtZoomCurVal, ZoomValue, true);
    m_sliderZoom.SetPos(ZoomValue);                            Zoom ｽﾗｲﾀﾞｰﾊﾞｰの現在値を設定

    // Focus についても同様
}
```

■ CSR ﾚｼﾞｽﾀのﾘｰﾄﾞﾗｲﾄｻﾝﾌﾟﾙ

```
void CCSRDlg::OnBtnRead()
{
    HRESULT hr;
    LONG  csrOffset = 0xF0F00880;                              Zoom ｺﾝﾄﾛｰﾙは CSR ｵﾌｾｯﾄ 880h

    hr = m_theApp->m_pIsdk->ccd_ReadRegister                   CSR ｵﾌｾｯﾄ 880h にﾏｯﾋﾟﾝｸﾞされている
                            (m_CameraId, csrOffset, &m_lReadVal);  ZOOM ﾚｼﾞｽﾀを Quadlet ﾘｰﾄﾞ

    m_lReadVal = m_lReadVal & 0xFFF;                           Bit20-31 の現在値 Value を取得
    UpdateData( FALSE );                                       （IIDC Digital Camera Spec 参照）
}
```

```
void CCSRDlg::OnBtnWrite()
{
    HRESULT hr;
    LONG  lReadVal,lWriteVal;
    LONG  csrOffset = 0xF0F00880;                              Zoom ｺﾝﾄﾛｰﾙは CSR ｵﾌｾｯﾄ 880h

    UpdateData( TRUE );

    hr = m_theApp->m_pIsdk->ccd_ReadRegister                   CSR ｵﾌｾｯﾄ 880h にﾏｯﾋﾟﾝｸﾞされている
                            (m_CameraId, csrOffset, &lReadVal);  ZOOM ﾚｼﾞｽﾀを Quadlet ﾘｰﾄﾞ

    lWriteVal = (lReadVal & 0xFFFFF000) | (m_lWriteVal & 0xFFF);   Bit 0-19 はﾘｰﾄﾞした値を、Bit20-31 に
                                                               入力値をｾｯﾄ
    hr = m_theApp->m_pIsdk->ccd_WriteRegister                  CSR ｵﾌｾｯﾄ 880h にﾏｯﾋﾟﾝｸﾞされている
                            (m_CameraId, csrOffset, lWriteVal);  ZOOM ﾚｼﾞｽﾀに Quadlet ﾗｲﾄ
}
```

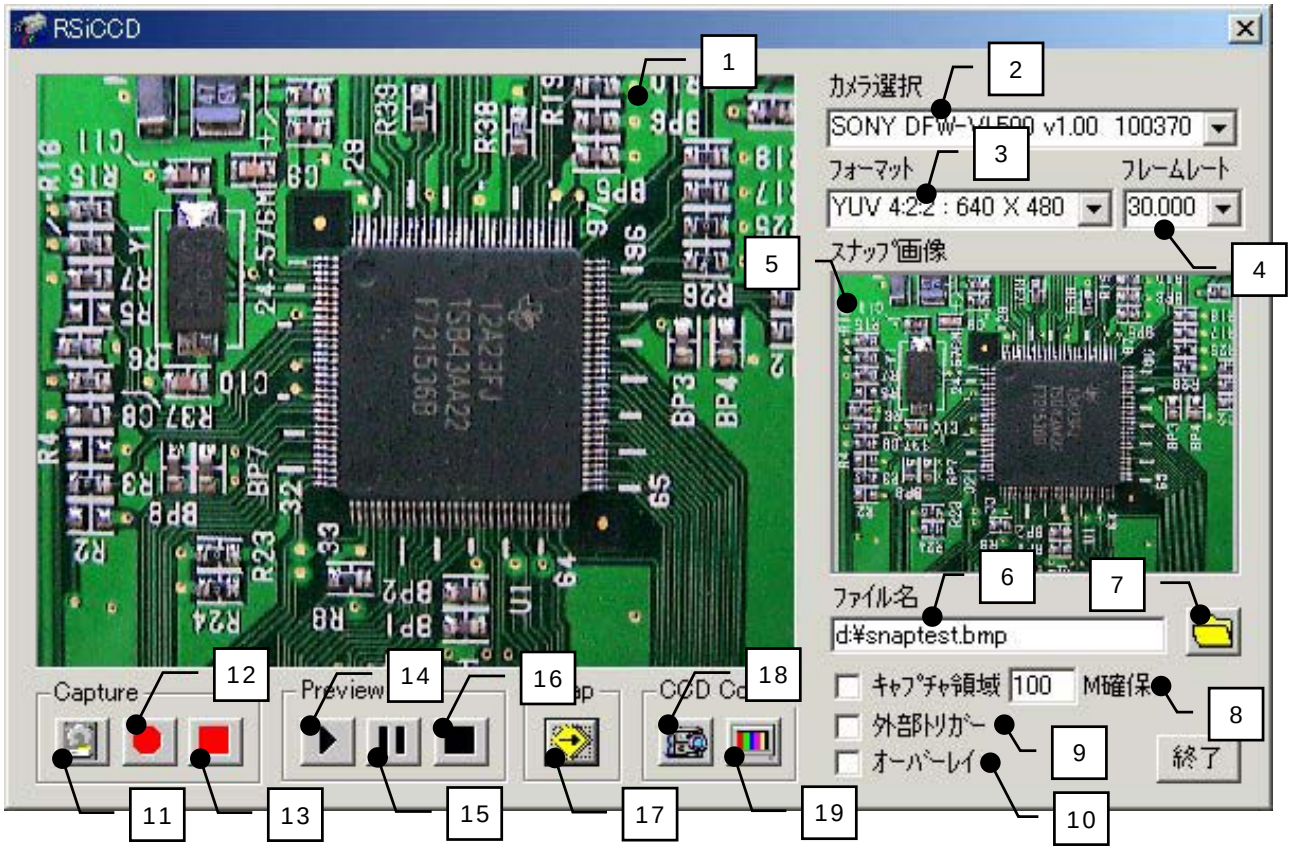
■ ﾒﾓﾘﾁｬﾝﾈﾙのｾｰﾌﾞとﾛｰﾄﾞ

```
void CFunctionsDlg::OnBtnChLoad()
{
    HRESULT hr = m_theApp->m_pIsdk->ccd_GetMemCh               ﾒﾓﾘｰﾁｬﾝﾈﾙ 1 に記憶されている設定をﾛｰ
                                        (m_CurrentCameraId, 0);  ﾄﾞ
}
```

```
void CFunctionsDlg::OnBtnChSave()
{
    HRESULT hr = m_theApp->m_pIsdk->ccd_SetMemCh               ﾒﾓﾘｰﾁｬﾝﾈﾙ 1 に現在の設定を記録
                                        (m_CurrentCameraId, 0);
}
```

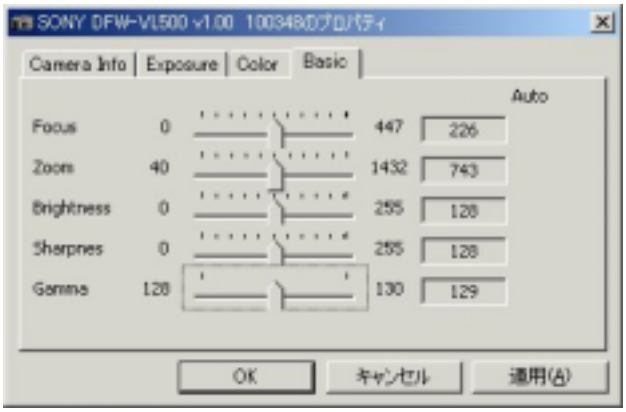
RSiCCD ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM\ForVC\MfcApp\RSiCCD を参照して下さい。

本ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑはこれまで説明したｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑの主な機能をまとめたものです。

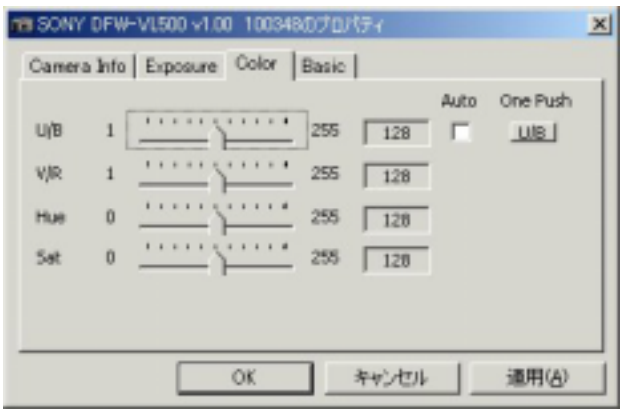


1	プレビュー実行によりこのスクリーン部分に動画が表示されます。
2	カメラリストからターゲットカメラを選択します。
3	選択されたカメラがサポートするフォーマットリストからフォーマットを選択します。
4	選択されたカメラがサポートするフレームレートリストからフレームレートを選択します。
5	スナップショット実行によりこのスクリーンに保存されたビットマップが表示されます。
6	キャプチャもしくはスナップショットするファイル名を入力します。スナップショットを行う場合のファイル名拡張子は「.BMP」、キャプチャを行う場合のファイル名拡張子は「.AVI」にします。
7	キャプチャもしくはスナップショットのファイル選択ダイアログが表示されます。
8	指定サイズのキャプチャファイルを予めアロケーションし、キャプチャ実行時の負荷を軽減します。キャプチャファイルがアロケーションサイズを越えた場合は、自動的にサイズ拡張されます。
9	CCD カメラが外部トリガーモードをサポートしていれば、トリガーモードで動作。外部トリガーの入力があると、プレビュースクリーンに１フレーム表示されます。
10	オーバーレイ表示機能をイネーブル。プレビューが高速になります。
11	キャプチャを行う前に必要となる設定を行います。
12	選択されたカメラからの動画キャプチャを開始します。
13	キャプチャを停止します。

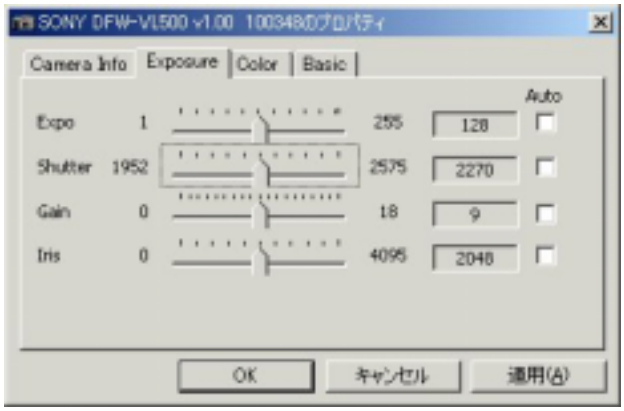
14	選択されたカメラからの動画プレビューを開始します。
15	プレビューを一時停止します。
16	プレビューを停止します。
17	プレビューしている動画 1 フレームをビットマップファイルに保存します。保存されたビットマップファイルは 5 に表示されます。
18	CCD カメラのプロパティ設定ダイアログ表示。
19	選択されたカメラについてホワイトバランスのオンショット自動調整を行います。

前ページ“17”のカメラ調整ボタンにより、右の CCD カメラのプロパティ設定ダイアログが表示されます。Basic タグを開きます。 （注意） 各機能の詳細およびサポート有無は CCD カメラ仕様を参照してください。	Basic 

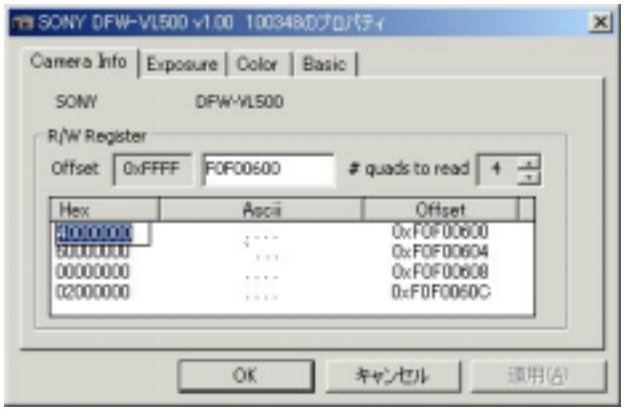
Focus	内蔵ズームレンズのピントを設定。
Zoom	内蔵ズームレンズの撮影倍率を調整。
Brightness	画像の黒レベルを調整。
Sharpness	画像の輪郭のシャープさを調整。
Gamma	ガンマ特性を調整。

Color タグを開きます。 （注意） 各機能の詳細およびサポート有無は CCD カメラ仕様を参照してください。	Color 

U/B	White Balance の B-gain 調整。
V/R	White Balance の R-gain 調整。
Hue	色相を調整。
Sat	Saturation:色の飽和度を調整。

Exposure タグを開きます。 （注意） 各機能の詳細およびサポート有無は CCD カメラ仕様を参照してください。	Exposure 

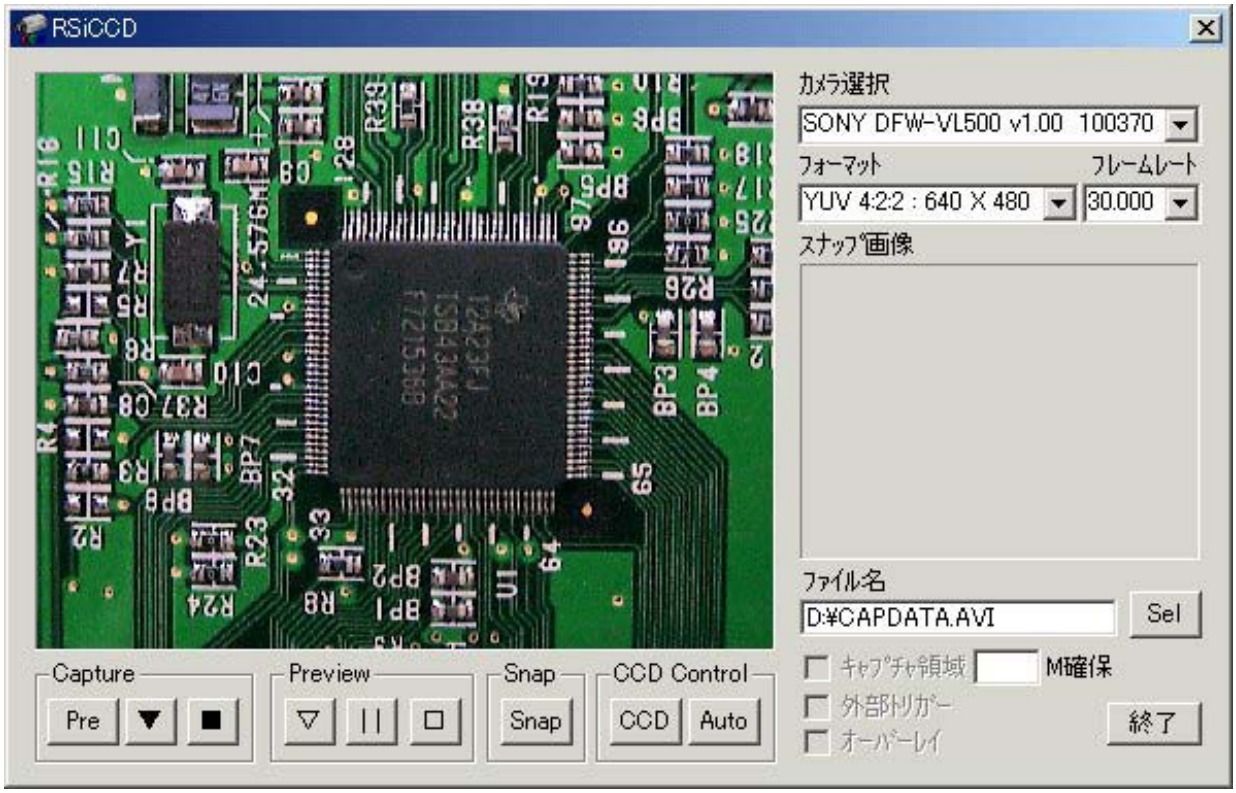
Expo	Exposure:自動露出（Auto Exposure）の露出補正を調整。
Shutter	電子シャッターの露光時間を調整。
Gain	映像信号アンプのゲインを調整。
Iris	内蔵レンズの絞りを調整。

Camera Info タグを開きます。CCD カメラの Status and Control Register（CSR）レジスタに対し Read/Write を行います。 （注意） CSR レジスタの詳細については CCD カメラの仕様書および IIDC 1394-based Digital Camera 仕様書を参照してください。	Camera Info 

3-5. VC サンプルプログラム

C 言語のｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ RSiCCD が CD-ROM\ForVC\W32App\RSiCCD に格納されています。

本ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑは「Win32 Application」のﾌﾟﾛｼﾞｪｸﾄ仕様で作成したものです。C ｿｰｽﾌｧｲﾙのﾌｧｲﾙ拡張子は「.cpp」で作成して下さい。詳細については、添付のｿｰｽｺｰﾄﾞを参照願います。



RsCCDSDK 追加関数マニュアル

追加された関数は CCD の画像をメモリ上で扱うためのものです。
新しいインターフェースを使用する際には下記のように宣言してインターフェースを作成する必要があります。

```
// CCD_COM コンポーネントの宣言
        IRsCcdSdk_1A *rsk;
// Camera ｺﾝﾄﾛｰﾙ COM ｲﾝﾀｰﾌｪｰｽ作成
        hr = CoCreateInstance(
                CLSID_RsCcdSdk,
                NULL,
                CLSCTX_INPROC,
                IID_IRsCcdSdk_1A,
                (void**) &rsk );
```

実際の関数の使用方法は CD 内のサンプル
GetBMPData(¥SDK¥ForVC¥MfcApp¥GetBMPData)をご参照ください。

Ver1.0 で作成したプログラムはそのままご使用になれます。

IRsCcdSdk_1A::
ccd_ReadyFrameData

YUV データをメモリ上で受け取るためにフィルタを接続する。

構文

```
HRESULT ccd_GetFrameInfo(
        [in]LONG CameraIndex
);
```

パラメータ

CameraIndex
[in]　ターゲットのカメラをインデックス番号で指定。 SDK_CCD_INFO 構造体参照。

戻り値

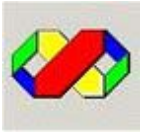
HRESULT を返す。以下のいずれかの値。エラーの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

注意

受け取り可能なフォーマットは YUV411， YUV422， YUV444。

IRsCcdSdk_1A::
ccd_GetFrameInfo



現在接続されている CCD **カメラのストリームデータに関する情報を取得。**

構文

```
HRESULT ccd_GetFrameInfo(
    LONG CameraIndex, LONG *pPixelFormat, LONG *pDataWidth,
    LONG *pDataHeight
);
```

パラメータ

CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**
pPixelFormat
[out]　**データの種類を示す数へのポインタ。**
pDataWidth
[out]　YUV **データの横幅へのポインタ。**
pDataHeight
[out]　YUV **データの縦幅へのポインタ。**

戻り値

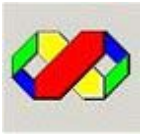
HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

注意

受け取り可能なフォーマットは YUV411， YUV422， YUV444。

IRsCcdSdk_1A::
ccd_GetFrameData



現在接続されている CCD **カメラから** YUV **データを取得する。**

構文

```
HRESULT ccd_GetFrameData(
  LONG CameraIndex, LONG* lpData, LONG* lpDataSize
);
```

パラメータ

CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**
lpData
[in, out]　**画像データへのポインタ。**
lpDataSize
[in, out]　**データサイズへのポインタ。**

戻り値

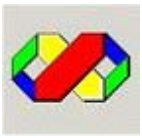
HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

注意

lpDataが0のとき、この関数はlpDataSizeへ必要なデータサイズを返す。
受け取り可能なフォーマットは YUV411，　YUV422，　YUV444。

IRsCcdSdk_1A::
ccd_SetFrameDataMsg



CCD カメラからストリームへフレームが送られた時に送られるメッセージを指定

構文

```
HRESULT ccd_GetFrameInfo(
    [in]LONG CameraIndex, [in]HWND hWnd, [in]LONG UserMsg
);
```

パラメータ

CameraIndex
[in]　ターゲットのカメラをインデックス番号で指定。 SDK_CCD_INFO 構造体参照。
hWnd
[in]　メッセージを受け取るウィンドウのハンドル
UserMsg
[in]　送信するメッセージ

戻り値

HRESULT を返す。以下のいずれかの値。エラーの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

IRsCcdSdk_1A::
ccd_ReleaseFrameData

YUV **受け取り用のフィルタを切り離す。**

構文

```
HRESULT ccd_ReleaseFrameData(
  LONG CameraIndex
);
```

パラメータ

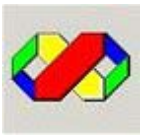
CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**

戻り値

HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

IRsCcdSdk_1A::
ccd_ReadyBmpData



ＲＧＢデータをメモリ上で受け取るためにフィルタを接続する。

構文

```
HRESULT ccd_GetFrameInfo(
        [in]LONG CameraIndex
);
```

パラメータ

CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**

戻り値

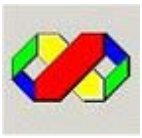
HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

注意

受け取り可能なフォーマットはＲＧＢ２４bit。

IRsCcdSdk_1A::
ccd_GetBmpInfo



ストリームの BMP **に関する情報を取得。**

構文

```
HRESULT ccd_GetFrameInfo(
      LONG CameraIndex, LONG *pDataWidth, LONG *pDataHeight
);
```

パラメータ

CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**
pDataWidth
[out]　YUV **データの横幅へのポインタ。**
pDataHeight
[out]　YUV **データの縦幅へのポインタ。**

戻り値

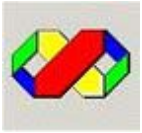
HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

注意

受け取り可能なフォーマットはＲＧＢ２４bit

IRsCcdSdk::
ccd_GetBmpData



現在接続されている CCD **カメラから** RGB24bit **データを取得する。**

構文

```
HRESULT ccd_GetFrameData(
  LONG CameraIndex, LONG* lpData, LONG* lpDataSize
);
```

パラメータ

CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**
lpData
[in, out]　**画像データへのポインタ。**
lpDataSize
[in, out]　**データサイズへのポインタ。**

戻り値

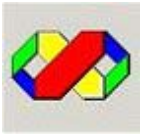
HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

注意

lpData **が** 0 **のとき、この関数は** lpDataSize **へデータサイズを返す。**
データは RGB24bit。

IRsCcdSdk_1A::
ccd_SetBmpDataMsg



CCD カメラからストリームへフレームが送られた時に送られるWindowsメッセージを指定

構文

```
HRESULT ccd_GetFrameInfo(
    [in]LONG CameraIndex, [in]HWND hWnd, [in]LONG UserMsg
);
```

パラメータ

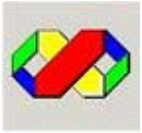
CameraIndex
[in]　ターゲットのカメラをインデックス番号で指定。 SDK_CCD_INFO 構造体参照。
hWnd
[in]　メッセージを受け取るウィンドウのハンドル
UserMsg
[in]　送信するメッセージ

戻り値

HRESULT を返す。以下のいずれかの値。エラーの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

IRsCcdSdk_1A::
ccd_ReleaseBmpData



BMP 受け取り用のフィルタを切り離す。

構文

```
HRESULT  ccd_ReleaseFrameData(
  LONG  CameraIndex
);
```

パラメータ

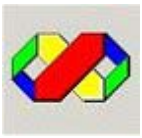
CameraIndex
[in]　ターゲットのカメラをインデックス番号で指定。 SDK_CCD_INFO 構造体参照。

戻り値

HRESULT を返す。以下のいずれかの値。エラーの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

IRsCcdSdk::
ccd_ReadyOneShot



ＣＣＤカメラへ Continuous_Shot **コマンドを送る。**(Video data **の停止**)

構文

```
HRESULT  ccd_GetFrameData(
  LONG  CameraIndex
);
```

パラメータ

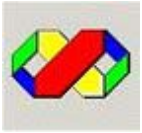
CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**

戻り値

HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

IRsCcdSdk_1A::
ccd_OneShot



CCD カメラへ One_Shot コマンドを送る。

構文

```
HRESULT  ccd_ReleaseFrameData(
  LONG  CameraIndex
);
```

パラメータ

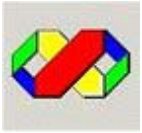
CameraIndex
[in]　ターゲットのカメラをインデックス番号で指定。 SDK_CCD_INFO 構造体参照。

戻り値

HRESULT を返す。以下のいずれかの値。エラーの詳細は ccd_GetLastError で取得。

S_OK	成功
S_FALSE	失敗

IRsCcdSdk::
ccd_ReleaseOneShot



CCD **カメラへ** Continuous_Shot **コマンドを送る。**(Video data **の開始**)

構文

```
HRESULT  ccd_GetFrameData(
  LONG  CameraIndex
);
```

パラメータ

CameraIndex
[in]　**ターゲットのカメラをインデックス番号で指定。** SDK_CCD_INFO **構造体参照。**

戻り値

HRESULT **を返す。以下のいずれかの値。エラーの詳細は** ccd_GetLastError **で取得。**

S_OK	**成功**
S_FALSE	**失敗**

第四章 Visual BASIC によるﾌﾟﾛｸﾞﾗﾑ開発

Visual BASIC でアプリケーションを作成する場合は、本製品付属の CCD SDK ActiveX ライブラリを使用します。CCD SDK のインストレーションを行えば CCD SDK ActiveX ライブラリが使用可能な状態となります。以下の CCD SDK ActiveX ライブラリインターフェイスとサンプルプログラムを参考にして、アプリケーションの作成を行います。

4-1. ActiveX インターフェイス仕様

ｲﾝﾀｰﾌｪｲｽ名	説　　明	頁
ccdAbout	CCD SDK ActiveX のﾊﾞｰｼﾞｮﾝ情報を取得。	72
ccdInitialize	CCD SDK ActiveX の初期化処理を行う。	88
ccdUninitialize	CCD SDK ActiveX の終了処理を行う。	109
ccdEnumCamera	ActiveX に対し現在接続されている CCD ｶﾒﾗを列挙するよう指示。	73
ccdFreeCamera	ccdEnumCamera 呼び出しにより CCD SDK ActiveX 内で確保したﾘｿｰｽを解放。	74
ccdGetCameraInfo	指定されたｶﾒﾗのﾕﾆｰｸな名前とｻﾎﾟｰﾄするｽﾄﾘｰﾑの個数を取得。	75
ccdGetConnectedCameraInfo	ActiveX から現在接続されている全てのｶﾒﾗ情報を取得。	76
ccdGetCurrentFormat	指定されたｶﾒﾗの現在設定されているｽﾄﾘｰﾑﾌｫｰﾏｯﾄおよびﾌﾚｰﾑﾚｰﾄを取得。	80
ccdGetSupportedFrameRate	指定ｶﾒﾗがｻﾎﾟｰﾄするｽﾄﾘｰﾑ・ﾌﾚｰﾑﾚｰﾄｲﾝﾃﾞｯｸｽ番号のﾌﾚｰﾑﾚｰﾄを取得。	83
ccdGetSupportedStream	指定ｶﾒﾗがｻﾎﾟｰﾄするｽﾄﾘｰﾑｲﾝﾃﾞｯｸｽ番号のﾌｫｰﾏｯﾄとﾋﾟｸｾﾙｻｲｽﾞを取得。	84
ccdSetStream	指定されたｶﾒﾗから出力されるｽﾄﾘｰﾑﾌｫｰﾏｯﾄを設定。	102
ccdSetFrameRate	指定されたｶﾒﾗから出力されるﾌﾚｰﾑﾚｰﾄを設定。	97
ccdPreviewStart	指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを開始。	90
ccdPreviewPause	指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰをﾎﾟｰｽﾞ。	89
ccdPreviewStop	指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを終了。	91
ccdSnapshot	現在表示されているﾌﾚｰﾑを静止画ﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙに保存。	106
ccdSetTrigger	CCD ｶﾒﾗの外部ﾄﾘｶﾞｰﾓｰﾄﾞを設定。	104
ccdSetAudio	動画のﾌﾟﾚﾋﾞｭｰおよび保存時にｵｰﾃﾞｨｵを有効設定。	93
ccdSetCaptureFilename	動画を保存するﾌﾙﾊﾟｽﾌｧｲﾙ名とｷｬﾌﾟﾁｬﾌｧｲﾙ保存にﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用するか指定。	94
ccdSetCaptureTempFile	ｷｬﾌﾟﾁｬ用のﾃﾝﾎﾟﾗﾘﾌｧｲﾙに関する設定。	95
ccdGetCopyFileProgress	ｷｬﾌﾟﾁｬ停止後のﾃﾝﾎﾟﾗﾘﾌｧｲﾙのｺﾋﾟｰｽﾃｰﾀｽを取得。	79
ccdSetMovieCompressor	ｷｬﾌﾟﾁｬ保存するの圧縮形式を設定。	99
ccdCaptureStart	指定されたｶﾒﾗより現在設定されたｽﾄﾘｰﾑ・ﾌﾚｰﾑﾚｰﾄで動画保存を開始。	107
ccdCaptureStop	動画保存を終了。	108
ccdGetPropertyRange	Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、調整ﾚﾝｼﾞ情報を取得。	87
ccdGetProperty	Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、ｶﾒﾗより現在値を取得。	86
ccdSetProperty	Zoom/Focus 等に関するﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更。	101

ccdGetColorRange	UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて調整ﾚﾝｼﾞ情報を取得。	78
ccdGetColor	UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについてｶﾒﾗより現在値を取得。	77
ccdSetColor	UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更する。	96
ccdShowPropertyPage	ｶﾒﾗﾄﾞﾗｲﾊﾞのﾌﾟﾛﾊﾟﾃｨﾍﾟｰｼﾞを表示。	105
ccdGetMemCh	ｶﾒﾗ内部のﾒﾓﾘに記憶されている設定ﾃﾞｰﾀを読み出しｶﾒﾗに再設定。	85
ccdSetMemCh	現在ｶﾒﾗに設定されているﾃﾞｰﾀをｶﾒﾗ内部のﾒﾓﾘに記憶。	98
ccdSetOverlay	ｵｰﾊﾞｰﾚｲ機能のｲﾈｰﾌﾞﾙ・ﾃﾞｨｾｰﾌﾞﾙを設定。	100
ccdReadRegister	指定のｶﾒﾗのｵﾌｾｯﾄｱﾄﾞﾚｽから 4 ﾊﾞｲﾄ（Quadlet）ﾘｰﾄﾞ。	92
ccdWriteRegister	指定のｶﾒﾗのｵﾌｾｯﾄｱﾄﾞﾚｽに 4 ﾊﾞｲﾄ（Quadlet）ﾗｲﾄ。	103
ccdGetLastError	最後に発生したｴﾗｰの詳細内容を取得。	81

■ ActiveX インターフェイスの書式について

ccdSdk.ccdXxxx　　【インターフェイス名】

ActiveX インターフェイス

【インターフェイスの概要動作説明】

構文

ccd_Xxxxx(Val As XXX) As Long　　【関数仕様説明】

パラメータ

【引数説明】
[in]　入力
[out]　出力

戻り値

【戻り値説明】

注意

【その他参考情報】

関連

【関連するｲﾝﾀｰﾌｧｲｽ】

ccdSdk.ccdAbout

[ActiveX インターフェイス]

CCD SDK ActiveX のﾊﾞｰｼﾞｮﾝ情報を取得。

構文

ccdAbout*(bstrAbout As String) As Long*

パラメータ

bstrAbout
[out]　SDK ﾊﾞｰｼﾞｮﾝ情報の格納先。”RS1394 CCD SDK ActiveX Version X.X”が返されます。

戻り値

以下のいずれかの値。

True	成功
False	失敗

ccdSdk.ccdEnumCamera

ActiveX インターフェイス

CCD SDK ActiveX に対し現在接続されている CCD ｶﾒﾗを列挙するよう指示。ActiveX 内に現在接続されているｶﾒﾗ情報と個々のﾘｿｰｽが確保されます。

構文

ccdEnumCamera*(lpNumOfCamera As Long) As Long*

パラメータ

pNumOfCamera
[out]　現在接続されている CCD ｶﾒﾗ数の格納先変数。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ｱﾌﾟﾘｹｰｼｮﾝ側では要素数 MAX_SUPPORT_CAMERAS の tCameraInfo 構造体配列を確保します。ｶﾒﾗ情報の再構築のために、再度 ccdEnumCamera()呼び出しを行う際、ｱﾌﾟﾘｹｰｼｮﾝを終了する際には必ず ccdFreeCamera()により ActiveX 内で確保されたﾘｿｰｽを解放します。

関連

ccdGetCameraInfo, ccdGetConnectedCameraInfo, ccdGetCurrentFormat, ccdGetSupportedFrameRate, ccdGetSupportedStream, ccdSetStream, ccdSetFrameRate, ccdFreeCamera 参照。

ccdSdk.ccdFreeCamera

[ActiveX インターフェイス]

ccdEnumCamera により CCD SDK ActiveX 内で確保している全てのﾘｿｰｽを解放。

構文

ccdFreeCamera

パラメータ

なし。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdEnumCamera, ccdGetCameraInfo, ccdGetConnectedCameraInfo, ccdGetCurrentFormat, ccdGetSupportedFrameRate, ccdGetSupportedStream, ccdSetStream, ccdSetFrameRate 参照。

ccdSdk.ccdGetCameraInfo

ActiveX インターフェイス

指定されたｶﾒﾗのﾕﾆｰｸな名前とそのｶﾒﾗがｻﾎﾟｰﾄするｽﾄﾘｰﾑの個数を取得。

構文

ccdGetCameraInfo*(lCameraIndex As Long, bstrCameraName As String,*
lpNumOfSupportedStream As Long) As Long

パラメータ

lCameraIndex
[in]　ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。tCameraInfo 構造体参照。
bstrCameraName
[out]　ﾕﾆｰｸなｶﾒﾗの名前の格納先。
lpNumOfSupportedStream
[out]　ｶﾒﾗがｻﾎﾟｰﾄするｽﾄﾘｰﾑ個数。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdEnumCamera, ccdGetConnectedCameraInfo, ccdGetCurrentFormat, ccdGetSupportedFrameRate, ccdGetSupportedStream, ccdSetStream, ccdSetFrameRate, ccdFreeCamera 参照。

ccdGetConnectedCameraInfo

ActiveX インターフェイス

CCD SDK ActiveX から現在接続されている全てのｶﾒﾗ情報を取得し、RSCCDSDK.BAS でｸﾞﾛｰﾊﾞﾙ定義された tCameraInfo 構造体変数にｾｯﾄ。

構文

```
Public Function ccdGetConnectedCameraInfo() As Boolean
```

パラメータ

なし

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

本サブルーチンは ” RSCCDSDK.BAS ” で定義されています。

関連

ccdEnumCamera, ccdGetCameraInfo, ccdGetCurrentFormat, ccdGetSupportedFrameRate, ccdGetSupportedStream, ccdSetStream, ccdSetFrameRate, ccdFreeCamera 参照。

ccdSdk.ccdGetColor

ActiveX インターフェイス

UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについてｶﾒﾗより現在値を取得。

構文

ccdGetColor *(lCameraIndex As Long, lProperty As Long, lpValue As Long, fMode As Long) As Long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
Property
[in] COLOR_PROPERTY より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。
lpValue
[out] 指定したﾌﾟﾛﾊﾟﾃｨの値を格納する変数。
fMode
[in] 指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。Camera Control Flag を使って指定。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdGetPropertyRange, ccdGetProperty, ccdSetProperty, ccdGetColorRange, ccdSetColor 参照。

ccdSdk.ccdGetColorRange

[ActiveX インターフェイス]

UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて調整ﾚﾝｼﾞ情報を取得。

構文

ccdGetColorRange *(lCameraIndex As Long, lProperty As Long, lpRangeMin As Long, lpRangeMax As Long, lpRangeSteppingDelta As Long, lpRangeDefault As Long, lpRangeCapsFlags As Long) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lProperty
[in] COLOR_PROPERTY よりﾚﾝｼﾞを取得するﾌﾟﾛﾊﾟﾃｨを指定。
lpRangeMin
[out] 調整範囲の最小値。tCCDRangeValue 構造体参照。
lpRangeMax
[out] 調整範囲の最大値。tCCDRangeValue 構造体参照。
lpRangeSteppingDelta
[out] 1 ｽﾃｯﾌﾟ増分値。tCCDRangeValue 構造体参照。
lpRangeDefault
[out]ﾃﾞﾌｫﾙﾄ値。tCCDRangeValue 構造体参照。
lpRangeCapsFlags
[out] ﾋﾞｯﾄﾏｯﾌﾟで定義されたｻﾎﾟｰﾄﾓｰﾄﾞ。tCCDRangeValue 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdGetPropertyRange, ccdGetProperty, ccdSetProperty, ccdGetColor, ccdSetColor 参照。

ccdSdk.ccdGetCopyFileProgress

[ActiveX インターフェイス]

ｷｬﾌﾟﾁｬ停止後のﾃﾝﾎﾟﾗﾘﾌｧｲﾙのｺﾋﾟｰｽﾃｰﾀｽを取得。

構文

ccdGetCopyFileProgress*(lCameraIndex As Long, lpProgress As Long) As Long*

パラメータ

lCameraIndex
[in]　ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lpProgress
[out] ｺﾋﾟｰ進捗ｽﾃｰﾀｽが返される変数。0-100[%]の値で 100 が完了。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdStartCapture, ccdStopCapture, ccdSetCaptureFile, ccdSetCaptureTempFile 参照。

ccdSdk.ccdGetCurrentFormat

ActiveX インターフェイス

指定されたｶﾒﾗの現在設定されているｽﾄﾘｰﾑﾌｫｰﾏｯﾄおよびﾌﾚｰﾑﾚｰﾄを取得。

構文

ccdGetCurrentFormat
(lCameraIndex As Long, lpCurrentStream As Long, lpFrameRate As Long) As Long

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lpCurrentStream
[out] 現在設定されているﾋﾞﾃﾞｵﾓｰﾄﾞのｲﾝﾃﾞｯｸｽ番号。tCameraInfo 構造体参照。
lpFrameRate
[out] 現在設定されているﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号。tCameraInfo 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdEnumCamera を呼び出しにより取得したｶﾒﾗ情報構造体（RSCCDSDK.BAS 参照）
Public gCameraInfo(MAX_NUMBER_OF_CAMERAS) As tCameraInfo
と ccdGetCurrentFormat()で取得した現在のﾋﾞﾃﾞｵﾓｰﾄﾞ・ﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号により、現在のﾋﾞﾃﾞｵﾓｰﾄﾞ・ﾌﾚｰﾑﾚｰﾄはそれぞれ下記のように定めることができます。
gCameraInfo[*lCameraIndex*].SupportedStream[*lpStreamIndex*].SupportedFormat
gCameraInfo[*lCameraIndex*].SupportedStream[*lpStreamIndex*].SupportedFrameRate[*lpFpsIndex*]

関連

ccdEnumCamera, ccdGetCameraInfo, ccdGetConnectedCameraInfo, ccdGetSupportedFrameRate, ccdGetSupportedStream, ccdSetStream, ccdSetFrameRate, ccdFreeCamera 参照。

ccdSdk.ccdGetLastError

ActiveX インターフェイス

最後に発生したｴﾗｰの詳細内容を取得。

構文

ccdGetLastError *(lpErrorCode As Long) As String*

パラメータ

lpErrorCode
[out] ｴﾗｰｺｰﾄﾞの格納先。

戻り値

String 型のｴﾗｰ情報を返す。

注意

ｴﾗｰの原因が推定できるものを次ﾍﾟｰｼﾞのｴﾗｰｺｰﾄﾞ一覧表に示します。対処方法に基づいて確認を行って下さい。その他、一覧表にないｴﾗｰｺｰﾄﾞについては、個々の環境における致命的なｴﾗｰと思われます。ﾊﾟｿｺﾝを再起動して下さい。

エラーコード一覧表

ｴﾗｰｺｰﾄﾞ	対　処　方　法
1	CCD ｶﾒﾗが接続されていないか、ｶﾒﾗﾄﾞﾗｲﾊﾞが正しくｲﾝｽﾄｰﾙされていません。ﾃﾞﾊﾞｲｽﾏﾈｰｼﾞｬ上にｲﾒｰｼﾞﾝｸﾞﾃﾞﾊﾞｲｽが認識されているか確認して下さい。
2	使用可能なｵｰﾃﾞｨｵﾃﾞﾊﾞｲｽが存在しません。ﾃﾞﾊﾞｲｽﾏﾈｰｼﾞｬ上にｵｰﾃﾞｨｵﾃﾞﾊﾞｲｽが認識されているか確認して下さい。
3	指定したﾄﾞﾗｲﾌﾞに十分な空き容量がありません。
4	動画ﾌｧｲﾙを作成できません。指定したﾌｧｲﾙが他のｱﾌﾟﾘｹｰｼｮﾝによって使用されていないか確認して下さい。
5	指定したﾊﾟｽは存在しません。ccdSetCaptureFile で指定したﾊﾟｽ名が正しいか再確認して下さい。
6	指定したﾌｧｲﾙ名の拡張子が AVI ではありません。
7	静止画ﾌｧｲﾙを作成できません。指定したﾌｧｲﾙが他のｱﾌﾟﾘｹｰｼｮﾝよって使用されていないか確認して下さい。
8	指定したﾊﾟｽは存在しません。ccdSnapShot で指定したﾊﾟｽ名が正しいか再確認して下さい。
9	指定したﾌｧｲﾙ名の拡張子が BMP ではありません。
10	無効なｶﾒﾗｲﾝﾃﾞｯｸｽが渡されました。ｲﾝﾃﾞｯｸｽ番号が有効か確認して下さい。
501	CcdEnumCamera の呼び出しが行われていません。
502	指定されたｶﾒﾗｲﾝﾃﾞｯｸｽ、ｽﾄﾘｰﾑｲﾝﾃﾞｯｸｽもしくはﾌﾚｰﾑﾚｰﾄｲﾝﾃﾞｯｸｽは無効です。
503	指定されたｶﾒﾗｲﾝﾃﾞｯｸｽは無効です。もしくはｶﾒﾗが接続されていません。
504	指定されたﾌﾟﾛﾊﾟﾃｨｲﾝﾃﾞｯｸｽが無効です。

ccdSdk.ccdGetSupportedFrameRate

ActiveX インターフェイス

指定されたｶﾒﾗがｻﾎﾟｰﾄしているｽﾄﾘｰﾑｲﾝﾃﾞｯｸｽ番号・ﾌﾚｰﾑﾚｰﾄｲﾝﾃﾞｯｸｽ番号のﾌﾚｰﾑﾚｰﾄを取得。

構文

ccdGetSupportedFrameRate*(lCameraIndex As Long, lStreamIndex As Long, lFrameRateIndex As long,lpFrameRate As Long) As Long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lStreamIndex
[in] ｽﾄﾘｰﾑｲﾝﾃﾞｯｸｽ番号を指定。
lFrameRateIndex
[in] ﾌﾚｰﾑﾚｰﾄｲﾝﾃﾞｯｸｽ番号を指定。
lpFrameRate
[out] ﾌﾚｰﾑﾚｰﾄの格納先変数。100 ﾅﾉ秒単位のﾌﾚｰﾑﾚｰﾄ。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

上記で取得した lpFrameRate[単位：100 ﾅﾉ秒/ﾌﾚｰﾑ]を[単位：ﾌﾚｰﾑ/秒]に変換するためには下記のように行います。

```
Dim lpFrameRate As Long, dFps As Double
dFps = 10000000# / lpFrameRate
cboFrameRate.AddItem Format$(dFps, "#.00")
```

関連

ccdEnumCamera, ccdGetCameraInfo, ccdGetConnectedCameraInfo, ccdGetCurrentFormat, ccdGetSupportedStream, ccdSetStream, ccdSetFrameRate, ccdFreeCamera 参照。

ccdSdk.ccdGetSupportedStream

ActiveX インターフェイス

指定されたｶﾒﾗがｻﾎﾟｰﾄしているｽﾄﾘｰﾑｲﾝﾃﾞｯｸｽ番号のﾌｫｰﾏｯﾄの種類とﾋﾟｸｾﾙｻｲｽﾞを取得。

構文

ccdGetSupportedStream(*lCameraIndex As Long, lStreamIndex As Long,*
lpFormat As long, lpPicSizeX As Long, lpPicSizeY As Long) As Long

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lStreamIndex
[in] ｽﾄﾘｰﾑｲﾝﾃﾞｯｸｽ番号を指定。
lpFormat
[out] ﾌﾚｰﾑﾚｰﾄｲﾝﾃﾞｯｸｽ番号を指定。
lpPicSizeX
[out] ﾋﾟｸｾﾙｻｲｽﾞ幅の格納先。
lpPicSizeY
[out] ﾋﾟｸｾﾙｻｲｽﾞ高さの格納先。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdEnumCamera, ccdGetCameraInfo, ccdGetConnectedCameraInfo, ccdGetCurrentFormat, ccdGetSupportedFrameRate, ccdSetStream, ccdSetFrameRate, ccdFreeCamera 参照。

ccdSdk.ccdGetMemCh

ActiveX インターフェイス

ｶﾒﾗ内部のﾒﾓﾘに記憶されている設定ﾃﾞｰﾀを読み出しｶﾒﾗに再設定。

構文

ccdGetMemCh(*lCameraIndex As Long, lMemoryChannel As Long*) *As long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lMemoryChannel
[in] ｶﾒﾗ内部のﾒﾓﾘﾁｬﾝﾈﾙ番号を指定。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

本機能の詳細に関しては CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞを参照願います。

関連

ccdSetMemCh

ccdSdk.ccdGetProperty

ActiveX インターフェイス

Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、ｶﾒﾗより現在値を取得。

構文

ccdGetProperty

(lCameraIndex As Long, lProperty As Long, lpValue As Long, fMode As Long) As Long

パラメータ

CameraIndex

[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。

Property

[in] BASE_PROPERTY より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。

lpValue

[out] 指定したﾌﾟﾛﾊﾟﾃｨの値を格納する変数。

fMode

[in] 指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。Camera Control Flag を使って指定。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdGetPropertyRange, ccdSetProperty, ccdGetColorRange, ccdGetColor, ccdSetColor 参照。

ccdSdk.ccdGetPropertyRange

ActiveX インターフェイス

Zoom/Focus 等ﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨに関し、調整ﾚﾝｼﾞ情報を取得。

構文

ccdGetPropertyRange(*lCameraIndex As Long, lProperty As Long, lpRangeMin As Long, lpRangeMax As Long, lpRangeSteppingDelta As Long, lpRangeDefault As Long, lpRangeCapsFlags As Long*) *As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lProperty
[in] BASE_PROPERTY よりﾚﾝｼﾞを取得するﾌﾟﾛﾊﾟﾃｨを指定。
lpRangeMin
[out] 調整範囲の最小値。tCCDRangeValue 構造体参照。
lpRangeMax
[out] 調整範囲の最大値。tCCDRangeValue 構造体参照。
lpRangeSteppingDelta
[out] 1 ｽﾃｯﾌﾟ増分値。tCCDRangeValue 構造体参照。
lpRangeDefault
[out]ﾃﾞﾌｫﾙﾄ値。tCCDRangeValue 構造体参照。
lpRangeCapsFlags
[out] ﾋﾞｯﾄﾏｯﾌﾟで定義されたｻﾎﾟｰﾄﾓｰﾄﾞ。tCCDRangeValue 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdGetProperty, ccdSetProperty, ccdGetColorRange, ccdGetColor, ccdSetColor 参照。

ccdSdk.ccdInitialize

ActiveX インターフェイス

CCD SDK ActiveX の初期化処理。他のｲﾝﾀｰﾌｪｲｽﾗｲﾌﾞﾗﾘの呼び出しを行う前に必ず SDK ﾗｲﾌﾞﾗﾘの初期化処理を行います。

構文

ccdInitialize

パラメータ

なし

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ｱﾌﾟﾘｹｰｼｮﾝ終了時、必ず ccdUninitialize により CCD SDK ActiveX の終了処理を行います。

ccdSdk.ccdPreviewPause

ActiveX インターフェイス

指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰをﾎﾟｰｽﾞ。

構文

ccdPreviewPause*(lCameraIndex As Long) As Long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdPreviewStart, ccdPreviewStop 参照。

ccdSdk.ccdPreviewStart

ActiveX インターフェイス

指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを開始。

構文

ccdPreviewStart*(lCameraIndex As Long, hWnd As wireHWND) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
hWnd
[in] ﾌﾟﾚﾋﾞｭｰｳｨﾝﾄﾞｳのﾊﾝﾄﾞﾙ。0 を渡した場合はﾃﾞﾌｫﾙﾄのﾌﾟﾚﾋﾞｭｰｳｨﾝﾄﾞｳが表示されます。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdPreviewStop, ccdPreviewPause 参照。

ccdSdk.ccdPreviewStop

ActiveX インターフェイス

指定されたｶﾒﾗのﾌﾟﾚﾋﾞｭｰを終了。

構文

ccdPreviewStop*(lCameraIndex As Long) As Long*

パラメータ

CameraIndex
[out] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdPreviewStart, ccdPreviewPause 参照。

ccdSdk.ccdReadRegister

ActiveX インターフェイス

指定のｶﾒﾗのｵﾌｾｯﾄｱﾄﾞﾚｽから 4 ﾊﾞｲﾄ（Quadlet）ﾘｰﾄﾞ。

構文

ccdReadRegister*(lCameraIndex As long, lOffset As Long, lpValue As Long) As Long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
Offset
[in] ﾘｰﾄﾞするﾚｼﾞｽﾀのﾍﾞｰｽｱﾄﾞﾚｽからのﾊﾞｲﾄｵﾌｾｯﾄ。
lValue
[out] ﾘｰﾄﾞした値の格納先ｱﾄﾞﾚｽ。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdReadRegister ﾒｿｯﾄﾞの *Offset* 引数は、CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞに示されている ”Camera Control & Status Register（CSR） ”のｱﾄﾞﾚｽを指定します。例えば、SONY DFW-VL500 の場合 CSR ｱﾄﾞﾚｽは F0F00000h からﾏｯﾋﾟﾝｸﾞされています。
CSR ｱﾄﾞﾚｽは、Configuration ROM の Unit Dependent Directory の command_regs_base とその key_type と Key_value により求めることができます。
詳細は、IIDC Specification および IEEE 1212 ｱﾄﾞﾚｯｼﾝｸﾞ規格参照。

関連

ccdWriteRegister 参照。

ccdSdk.ccdSetAudio

ActiveX インターフェイス

動画のﾌﾟﾚﾋﾞｭｰおよび保存時、同時にｵｰﾃﾞｨｵﾃﾞｰﾀを有効にするか無効にするか設定。

構文

ccdSetAudio*(lCameraIndex As Long, bAudioOn As Boolean) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
bAudioOn
[in] ｵｰﾃﾞｨｵ入力の ON/OFF 設定。TRUE: ｵｰﾃﾞｨｵ ｵﾝ FALSE: ｵｰﾃﾞｨｵ ｵﾌ (ﾃﾞﾌｫﾙﾄ)

戻り値

以下のいずれかの値。

True	成功
False	失敗

ccdSdk.ccdSetCaptureFilename

[ActiveX インターフェイス]

動画を保存するﾌﾙﾊﾟｽﾌｧｲﾙ名と、ｷｬﾌﾟﾁｬﾌｧｲﾙ保存にﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用するか指定。ﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用しない場合、ccdStartCapture によりｷｬﾌﾟﾁｬを開始すると第 2 引数で指定したﾌｧｲﾙに直接動画ﾃﾞｰﾀが保存されます。

構文

ccdSetCaptureFilename(*lCameraIndex As Long, bstrFilename As String, bTempFileOn As Boolean) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
bstrFilename
[in] ｷｬﾌﾟﾁｬ出力名とﾌﾙﾊﾟｽを示す文字列。
bTempFileOn
[in] ﾃﾝﾎﾟﾗﾘﾌｧｲﾙ設定ﾌﾗｸﾞ。TRUE: ﾃﾝﾎﾟﾗﾘﾌｧｲﾙ使用　FALSE: 直接指定ﾌｧｲﾙに保存（ﾃﾞﾌｫﾙﾄ）

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用する設定を行った場合は、基本的に ccdSetCaptureTempFile によりﾃﾝﾎﾟﾗﾘﾌｧｲﾙの設定を行って下さい。ccdStopCapture によりｷｬﾌﾟﾁｬを停止すると、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙから第 2 引数で指定したﾌｧｲﾙにｷｬﾌﾟﾁｬﾃﾞｰﾀがｺﾋﾟｰされます。

関連

ccdCaptureStart, ccdCaptureStop, ccdSetCaptureTempFile, ccdGetCopyFileProgress 参照。

ccdSdk.ccdSetCaptureTempFile

ActiveX インターフェイス

ｷｬﾌﾟﾁｬ用のﾃﾝﾎﾟﾗﾘﾌｧｲﾙに関する設定。領域確保ｵﾝにすると、ｷｬﾌﾟﾁｬ実行前に予めﾊｰﾄﾞﾃﾞｨｽｸ上に指定ｻｲｽﾞのﾝﾎﾟﾗﾘﾌｧｲﾙ領域が確保されｷｬﾌﾟﾁｬ実行時の負荷が小さくなります。

構文

ccdSetCaptureTempFile(*lCameraIndex As Long, bstrFilename As String, bPreAllocOn As Boolean, lAllocMBytes As Long*) As Long

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
bstrFilename
[in] ｷｬﾌﾟﾁｬﾃﾝﾎﾟﾗﾘﾌｧｲﾙ名とﾌﾙﾊﾟｽを示す文字列。
bPreAllocOn
[in] ﾌｧｲﾙｱﾛｹｰｼｮﾝ設定。TRUE: 領域確保ｵﾝ　FALSE: 領域確保ｵﾌ（ﾃﾞﾌｫﾙﾄ）
lAllocMBytes
[in] ﾊｰﾄﾞﾃﾞｨｽｸ上に領域確保するﾌｧｲﾙのｻｲｽﾞ(メガバイト単位)。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ﾊｰﾄﾞﾃﾞｨｽｸ上に予めｷｬﾌﾟﾁｬ領域を確保することにより、動画ｷｬﾌﾟﾁｬ時のｵｰﾊﾞｰﾍｯﾄﾞを最小にすることができます。第 2 引数のﾌｧｲﾙ名に ” ” を指定すると、環境変数 TEMP で定義されたﾌｫﾙﾀﾞにﾃﾞﾌｫﾙﾄのﾌｧｲﾙ名「RSCCDTMP.AVI」でﾃﾝﾎﾟﾗﾘﾌｧｲﾙが作成されます。このﾃﾝﾎﾟﾗﾘﾌｧｲﾙはｺﾋﾟｰ終了時削除されます。

関連

ccdCaptureStart, ccdCaptureStop, ccdSetCaptureFile, ccdGetCopyFileProgress 参照。

ccdSdk.ccdSetColor

ActiveX インターフェイス

UB/VR/Hue/Saturation に関するｶﾗｰﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更する。

構文

ccdSetColor *(lCameraIndex As Long, lProperty As Long, lValue As Long, fMode As Long) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lProperty
[in] COLOR_PROPERTY より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。
lValue
[in] 新たに設定するﾌﾟﾛﾊﾟﾃｨの値。fMode に fCameraControlAuto もしくは fCameraControlOnePush が指定されている時は無視されます。
fMode
[in] 指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。Camera Control Flag を使って指定。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdGetPropertyRange, ccdGetProperty, ccdSetProperty, ccdGetColorRange, ccdGetColor 参照。

ccdSdk.ccdSetFrameRate

ActiveX インターフェイス

指定されたｶﾒﾗから出力されるﾌﾚｰﾑﾚｰﾄを設定。

構文

ccdSetFrameRate*(lCameraIndex As Long, lFrameRate As Long) As Long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lFrameRate
[in] 新しく設定するﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号。tCameraInfo 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdEnumCamera, ccdGetCameraInfo, ccdGetConnectedCameraInfo, ccdGetCurrentFormat, ccdGetSupportedFrameRate, ccdGetSupportedStream, ccdSetStream, ccdFreeCamera 参照。

ccdSdk.ccdSetMemCh

ActiveX インターフェイス

現在ｶﾒﾗに設定されているﾃﾞｰﾀをｶﾒﾗ内部のﾒﾓﾘに記憶。

構文

ccdSetMemCh(*lCameraIndex As Long, lMemoryChannel As Long*) *As long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lMemoryChannel
[in] ｶﾒﾗ内部のﾒﾓﾘﾁｬﾝﾈﾙ番号を指定。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

本機能の詳細に関しては CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞを参照願います。

関連

ccdGetMemCh 参照。

ccdSdk.ccdMovieCompressor

ActiveX インターフェイス

ｷｬﾌﾟﾁｬ保存するﾌｧｲﾙの圧縮形式を設定。ﾃﾞﾌｫﾙﾄでは無圧縮形式で保存されます。

構文

ccdSetMovieCompressor
(lCameraIndex As Long, lCompIndex As Long, bstrCompName As String) As Long

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lCompIndex
[in] 圧縮ﾌｫｰﾏｯﾄを示すｲﾝﾃﾞｯｸｽ番号を COMPRESSOR_FORMAT 構造体で指定。
bstrCompName
[in] 圧縮ﾌｫｰﾏｯﾄ名が格納されている String 文字列。圧縮ﾌｫｰﾏｯﾄｲﾝﾃﾞｯｸｽ番号 *lCompIndex* に SetByName が指定された場合は、この文字列で圧縮ﾌｫｰﾏｯﾄを指定します。圧縮ﾌｫｰﾏｯﾄ名は DirectShow のﾌｨﾙﾀ名と一致する必要があります。SetByName 以外の圧縮ﾌｫｰﾏｯﾄｲﾝﾃﾞｯｸｽ番号 *lCompIndex* を指定している場合は、0 を渡します。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

非圧縮ﾃﾞｰﾀをそのままｷｬﾌﾟﾁｬした場合は、処理が間に合わずこま落ち等の問題が発生します。DV Video Encoder 等の圧縮を行ってｷｬﾌﾟﾁｬすることによりこま落ち問題を改善することができます。

関連

ccdCaptureStart, ccdCaptureStop 参照。

ccdSdk.ccdSetOverlay

ActiveX インターフェイス

ｵｰﾊﾞｰﾚｲ機能のｲﾈｰﾌﾞﾙ・ﾃﾞｨｾｰﾌﾞﾙを設定。

構文

ccdSetOverlay*(lCameraIndex As Long, bOverlayOn As Boolean) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
bOverlayOn
[in] ｵｰﾊﾞｰﾚｲ機能の ON/OFF 設定。TRUE:ｵﾝ FALSE:ｵﾌ

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

1．ｵｰﾊﾞｰﾚｲ機能のｻﾎﾟｰﾄについて
ｵｰﾊﾞｰﾚｲ機能を有効にして CCD ｶﾒﾗからの動画ﾌﾟﾚﾋﾞｭｰを行うと、高速な動画表示が可能になります。ｵｰﾊﾞｰﾚｲ機能のｻﾎﾟｰﾄは使用しているｸﾞﾗﾌｨｯｸｱﾀﾞﾌﾟﾀに依存します。ｵｰﾊﾞｰﾚｲ機能がｻﾎﾟｰﾄされたｸﾞﾗﾌｨｯｸｱﾀﾞﾌﾟﾀでも､ｱﾀﾞﾌﾟﾀ側の問題によりﾌﾟﾚﾋﾞｭｰ画面の拡大縮小ができない場合があります。
2．外部ﾄﾘｶﾞｰﾓｰﾄﾞについて
ｵｰﾊﾞｰﾚｲﾓｰﾄﾞを有効にした場合は、外部ﾄﾘｶﾞｰﾓｰﾄﾞは使用できません。

関連

ccdSetTrigger 参照。

ccdSdk.ccdSetProperty

ActiveX インターフェイス

Zoom/Focus 等に関するﾍﾞｰｽﾌﾟﾛﾊﾟﾃｨについて、ｶﾒﾗの設定値を変更。

構文

ccdSetProperty
(lCameraIndex As Long, lProperty As Long, lValue As Long, fMode As Long) As Long

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
lProperty
[in] BASE_PROPERTY より現在の値を取得したいﾌﾟﾛﾊﾟﾃｨを指定。
lValue
[in] 新たに設定するﾌﾟﾛﾊﾟﾃｨの値。fMode に fCameraControlAuto もしくは
fCameraControlOnePush が指定されている時は無視されます。
fMode
[in] 指定したﾌﾟﾛﾊﾟﾃｨの現在のﾓｰﾄﾞを指定。Camera Control Flag を使って指定。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdGetPropertyRange, ccdGetProperty, ccdGetColorRange, ccdGetColor, ccdSetColor 参照。

ccdSdk.ccdSetStream

[ActiveX インターフェイス]

指定されたｶﾒﾗから出力されるｽﾄﾘｰﾑﾌｫｰﾏｯﾄを設定。

構文

ccdSetStream*(lCameraIndex As Long, lStreamIndex As Long) As Long*

パラメータ

CameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
StreamIndex
[in] 新しく設定するｽﾄﾘｰﾑのｲﾝﾃﾞｯｸｽ番号。tCameraInfo 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

関連

ccdEnumCamera, ccdGetCameraInfo, ccdGetConnectedCameraInfo, ccdGetCurrentFormat, ccdGetSupportedFrameRate, ccdGetSupportedStream, ccdSetFrameRate, ccdFreeCamera 参照。

ccdSdk.ccdWriteRegister

ActiveX インターフェイス

指定のｶﾒﾗのｵﾌｾｯﾄｱﾄﾞﾚｽに 4 ﾊﾞｲﾄ（Quadlet）ﾗｲﾄ。

構文

ccdWriteRegister*(lCameraIndex As long, lOffset As Long, lValue As Long) As Long*

パラメータ

CameraIndex
[in]　ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
Offset
[in]　ﾗｲﾄするﾚｼﾞｽﾀのﾍﾞｰｽｱﾄﾞﾚｽからのﾊﾞｲﾄｵﾌｾｯﾄ。
lValue
[in]　ﾗｲﾄする値。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdWriteRegister ﾒｿｯﾄﾞの *Offset* 引数は、CCD ｶﾒﾗのﾕｰｻﾞｰｽﾞｶﾞｲﾄﾞに示されている ”Camera Control & Status Register（CSR） ”のｱﾄﾞﾚｽを指定します。例えば、SONY DFW-VL500 の場合 CSR ｱﾄﾞﾚｽは F0F00000h からﾏｯﾋﾟﾝｸﾞされています。
CSR ｱﾄﾞﾚｽは、Configuration ROM の Unit Dependent Directory の command_regs_base とその key_type と Key_value により求めることができます。
詳細は、IIDC Specification および IEEE 1212 ｱﾄﾞﾚｯｼﾝｸﾞ規格参照。

関連

ccdReadRegister 参照。

ccdSdk.ccdSetTrigger

ActiveX インターフェイス

CCD ｶﾒﾗの外部ﾄﾘｶﾞｰﾓｰﾄﾞの ON/OFF を設定。

構文

ccdSetTrigger*(lCameraIndex As Long, bTriggerOn As Boolean) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
bTriggerOn
[in] 外部ﾄﾘｶﾞｰﾓｰﾄﾞの ON/OFF 設定。TRUE:ｵﾝ FALSE:ｵﾌ

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

1．ﾄﾘｶﾞｰ画像ﾃﾞｰﾀの遅れについて
外部ﾄﾘｶﾞｰﾓｰﾄﾞのｻﾎﾟｰﾄ有無は CCD ｶﾒﾗに依存します（詳細は CCD ｶﾒﾗのﾏﾆｭｱﾙを参照）。外部ﾄﾘｶﾞｰ信号が CCD ｶﾒﾗに入力されると、CCD ｶﾒﾗより 1 ﾌﾚｰﾑの画像が送られます。この時、実際に外部信号が入力された時点の画像ﾃﾞｰﾀと、CCD ｶﾒﾗより送られた画像ﾃﾞｰﾀには 100 ﾐﾘ秒前後の遅れがあります。
2.ﾄﾘｶﾞｰｲﾍﾞﾝﾄの取得について
ｱﾌﾟﾘｹｰｼｮﾝがﾄﾘｶﾞｰの発生を検出するためには、外部ﾄﾘｶﾞｰ発生装置からのﾄﾘｶﾞｰ信号は CCD ｶﾒﾗに入力すると同時に、ﾃﾞｼﾞﾀﾙ I/O 入出力ﾕﾆｯﾄを介してﾊﾟｿｺﾝにも入力する必要があります。
例えば、外部ﾄﾘｶﾞｰﾓｰﾄﾞでﾌﾟﾚﾋﾞｭｰを実行した場合、外部ﾄﾘｶﾞｰの入力があると CCD ｶﾒﾗより 1 ﾌﾚｰﾑの画像ﾃﾞｰﾀ送られ、ﾌﾟﾚﾋﾞｭｰ画面に 1 ﾌﾚｰﾑの静止画が表示されます。人がその画像を見てｽﾅｯﾌﾟｼｮｯﾄを実行することはできますが、ｱﾌﾟﾘｹｰｼｮﾝが自動で外部ﾄﾘｶﾞｰを検出してｽﾅｯﾌﾟｼｮｯﾄを実行する場合は、ﾃﾞｼﾞﾀﾙ I/O 入出力ﾕﾆｯﾄが必要になります。
3.ｵｰﾊﾞｰﾚｲﾓｰﾄﾞについて
外部ﾄﾘｶﾞｰﾓｰﾄﾞを有効にした場合は、ｵｰﾊﾞｰﾚｲ描画はできません。ccdSetOverlay 参照。

ccdSdk.ccdShowPropertyPage

ActiveX インターフェイス

ｶﾒﾗﾄﾞﾗｲﾊﾞのﾌﾟﾛﾊﾟﾃｨﾍﾟｰｼﾞを表示。

構文

ccdShowPropertyPage*(lCameraIndex As Long)*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。

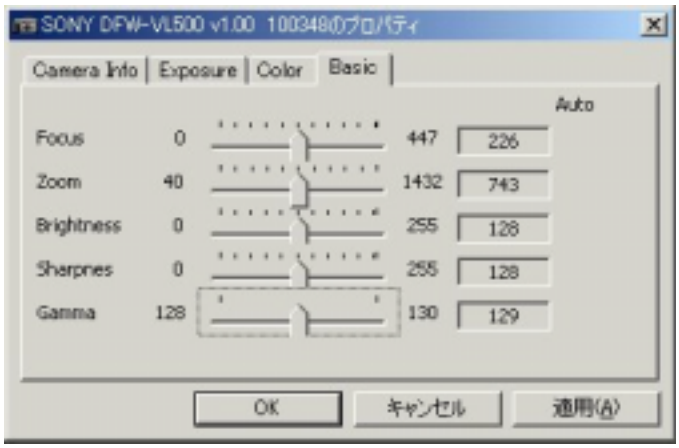
戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ｶﾒﾗﾄﾞﾗｲﾊﾞのﾌﾟﾛﾊﾟﾃｨﾍﾟｰｼﾞを右に示します。



関連

ccdGetPropertyRange, ccdGetProperty, ccdGetColorRange, ccdGetColor, ccdSetColor 参照。

ccdSdk.ccdSnapshot

ActiveX インターフェイス

現在表示されているﾌﾚｰﾑを静止画ﾋﾞｯﾄﾏｯﾌﾟ ﾌｧｲﾙに保存。

構文

ccdSnapshot*(lCameraIndex As Long, bstrFilename As String) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。
bstrFilename
[in] 出力ﾌｧｲﾙ名が格納されている文字列

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ｽﾅｯﾌﾟ ｼｮｯﾄしたﾌﾚｰﾑはｶﾗｰ CCD の場合は RGB24、ﾓﾉｸﾛ CCD の場合は RGB8 ﾌｫｰﾏｯﾄで保存されます。

ccdSdk.ccdCaptureStart

ActiveX インターフェイス

指定されたｶﾒﾗより現在設定されたｽﾄﾘｰﾑ・ﾌﾚｰﾑﾚｰﾄで動画保存を開始。

構文

ccdCaptureStart*(lCameraIndex As Long) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdSetCaptureFilename のﾃﾝﾎﾟﾗﾘﾌｧｲﾙ設定ﾌﾗｸﾞが True の場合は、一旦ﾃﾝﾎﾟﾗﾘﾌｧｲﾙに動画が保存されます。

関連

ccdSetCaptureFilename, ccdSetCaptureTempFile, ccdCaptureStop 参照。

ccdSdk.ccdCaptureStop

ActiveX インターフェイス

動画保存を終了。

構文

ccdCaptureStop*(lCameraIndex As Long) As Long*

パラメータ

lCameraIndex
[in] ﾀｰｹﾞｯﾄのｶﾒﾗをｲﾝﾃﾞｯｸｽ番号で指定。 tCameraInfo 構造体参照。

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdSetCaptureFilename のﾃﾝﾎﾟﾗﾘﾌｧｲﾙ設定ﾌﾗｸﾞが True の場合は、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙに保存された動画が ccdSetCaptureFilename で指定したﾌｧｲﾙにｺﾋﾟｰします。ccdGetCopyFileProgress を呼び出し、何%ｺﾋﾟｰが終了したが進捗情報を取得することができます。

関連

ccdSetCaptureFilename, ccdSetCaptureTempFile, ccdCaptureStart, ccdGetCopyFileProgress 参照。

ccdSdk.ccdUninitialize

[ActiveX インターフェイス]

CCD SDK ActiveX の終了処理を行う。

構文

ccdUninitialize

パラメータ

なし

戻り値

以下のいずれかの値。

True	成功
False	失敗

注意

ccdInitialize 参照。

4-2. 構造体・Flag 定義仕様

tCameraInfo,tSupportedStream,tPixel

```
Public Const MAX_NUMBER_OF_CAMERAS = 20
Public Const MAX_NUMBER_OF_STREAM = 10
Public Const MAX_NUMBER_OF_FRAMERATE = 8
```

```
Public Type tCameraInfo
CameraIndex As Long
szCameraName As String
SupportedStream(MAX_NUMBER_OF_STREAM) As tSupportedStream
End Type
```

CameraIndex	列挙した CCD ｶﾒﾗの 0 から始まるｲﾝﾃﾞｯｸｽ番号。
szCameraName	CCD ｶﾒﾗ名が格納される文字列ﾊﾞｯﾌｧ。
SupportedStream	CCD ｶﾒﾗがｻﾎﾟｰﾄするﾋﾞﾃﾞｵﾓｰﾄﾞ。各ﾋﾞﾃﾞｵﾓｰﾄﾞの詳細については **tSupportedStream** 構造体参照。配列の要素番号がｽﾄﾘｰﾑのｲﾝﾃﾞｯｸｽ番号に相当。

```
Public Type tSupportedStream
    SupportedFormat As Long
    SupportedPixSize As tPixel
    SupportedFrameRate(MAX_NUMBER_OF_FRAMERATE) As Long
End Type
```

SupportedFormat	CCD ｶﾒﾗがｻﾎﾟｰﾄしているﾌｫｰﾏｯﾄを示すｲﾝﾃﾞｯｸｽ番号。ｲﾝﾃﾞｯｸｽ番号は PIXELFORMAT Enum 構造体の番号になります。					
	番号	4	3	2	1	0
	ﾌｫｰﾏｯﾄ	YUV(4:1:1)	YUV(4:2:2)	YUV(4:4:4)	Y(MONO)	RGB32
SupportedPixSize	CCD ｶﾒﾗがｻﾎﾟｰﾄしている上記ﾌｫｰﾏｯﾄ時のﾋﾟｸｾﾙｻｲｽﾞ。					
SupportedStream	CCD ｶﾒﾗがｻﾎﾟｰﾄしている上記ﾌｫｰﾏｯﾄ時の 100 ﾅﾉ秒単位のﾌﾚｰﾑﾚｰﾄ。配列の要素番号がﾌﾚｰﾑﾚｰﾄのｲﾝﾃﾞｯｸｽ番号に相当。					

```
Public Type tPixel
        cx As Long
        cy As Long
End Type
```

cx	幅方向のﾋﾟｸｾﾙｻｲｽﾞ
cy	高さ方向のﾋﾟｸｾﾙｻｲｽﾞ

tCCDRangeValue

```
Public Type tCCDRangeValue
    Min As Long
    Max As Long
    SteppingDelta As Long
    Default As Long
    CapsFlags As Long
End Type
```

Min	設定可能な最小値。
Max	設定可能な最大値。
SteppingDelta	1 ｽﾃｯﾌﾟ増分値。
Default	ﾃﾞﾌｫﾙﾄ値。
CapsFlags	ﾋﾞｯﾄﾏｯﾌﾟで定義されたｻﾎﾟｰﾄﾓｰﾄﾞ。

関連

ccdGetPropertyRange, ccdGetProperty, ccdSetProperty, ccdGetColorRange, ccdGetColor, ccdSetColor 参照

BASE_PROPERTY

```
Public Enum BASE_PROPERTY
    Focus = 0
    Zoom
    Brightness
    Sharpness
    Gamma
    Autoexp
    Shutter
    Gain
    Iris
End Enum
```

関連

ccdGetPropertyRange, ccdGetProperty, ccdSetProperty, ccdGetColorRange, ccdGetColor, ccdSetColor 参照

COLOR_PROPERTY

```
Public Enum COLOR_PROPERTY
    UB = 0
    VR
    Hue
    Saturation
End Enum
```

関連

ccdGetColorRange, ccdGetColor, ccdSetColor 参照

PIXEL_FORMAT

```
Public Enum PIXEL_FORMAT
    FormatRGB24 = 0
    FormatY800
    FormatY444
    FormatY422
    FormatY411
End Enum
```

関連

tCameraInfo 構造体の tSupportedStream 型構造体ﾒﾝﾊﾞｰ SupportedStream のﾒﾝﾊﾞｰ SupportedFormat は、上記の Enum 構造体の番号でｱｻｲﾝすることができます。

COMPRESSOR_FORMAT

```
Public Enum COMPRESSOR_FORMAT
    SetByName = 0
    NoCompressor
    MSVideo
    Indeo
    Cinepak
    DVVideoEncode
End Enum
```

関連

ccdSetMovieCompressor 参照

Camera Control Flag

```
Public Const MAX_NUMBER_OF_CAMERAS = 20
Public Const MAX_NUMBER_OF_STREAM = 10
Public Const MAX_NUMBER_OF_FRAMERATE = 8

Public Const fCameraControlAuto = 1
Public Const fCameraControlManual = 2
Public Const fCameraControlOnePush = 4
Public Const fCameraControlOn = 8
Public Const fCameraControlOff = 16

Public Const fFeatureControlAbsolute = 0
Public Const fFeatureControlRelative  = 16

Public Const strCcdAttach = "CCD_ATTACH"
Public Const strCcdRemove = "CCD_REMOVE"
```

関連

ccdGetProperty, ccdSetProperty,ccdGetColor, ccdSetColor 参照

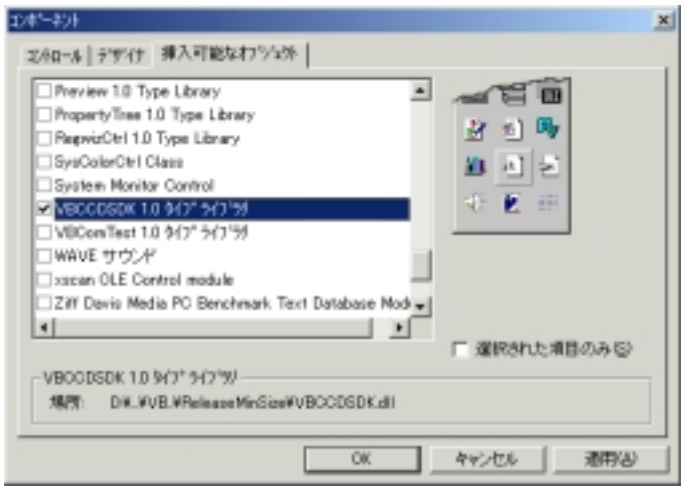
4-3. Visual Basic プログラム作成方法

（4-3-1）プログラム作成の準備

Visual BASIC でﾌﾟﾛｸﾞﾗﾑを作成する場合は最初に下記手順で CCD SDK ActiveX の組み込みを行います。

STEP.1

Microsoft Visual BASIC 6.0 を起動します。新しいﾌﾟﾛｼﾞｪｸﾄを作成します。「ﾌﾟﾛｼﾞｪｸﾄ」ﾒﾆｭｰの「ｺﾝﾎﾟｰﾈﾝﾄ」を選択します。右ﾀﾞｲｱﾛｸﾞの「挿入可能なｵﾌﾞｼﾞｪｸﾄ」ﾀｸﾞを開き、「VBCCDSDK 1.0 ﾀｲﾌﾟﾗｲﾌﾞﾗﾘ」をﾁｪｯｸし「OK」ﾎﾞﾀﾝをｸﾘｯｸします。

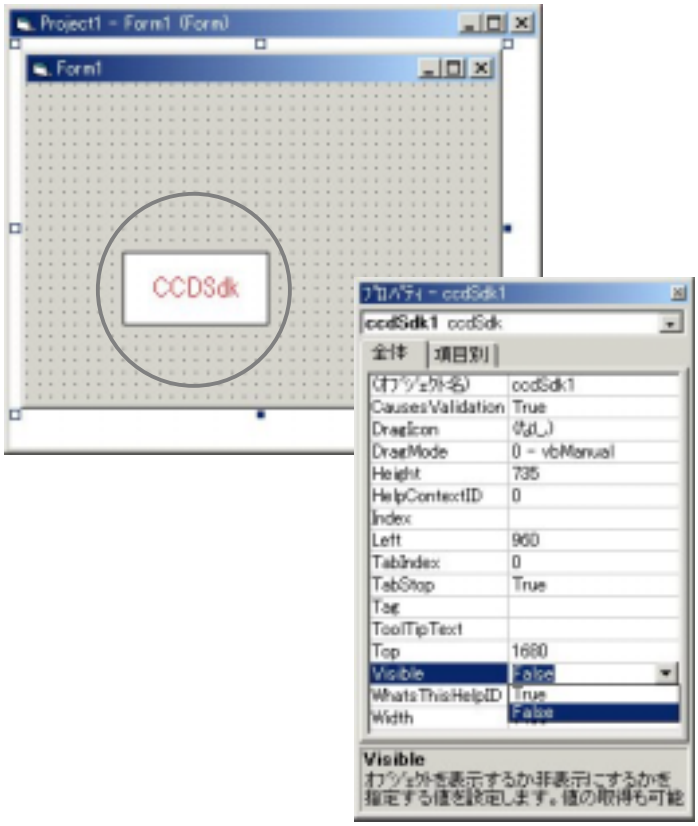


STEP.2

ｺﾝﾄﾛｰﾙﾊﾞｰに右のように CCD SDK ActiveX が追加されていることを確認します。

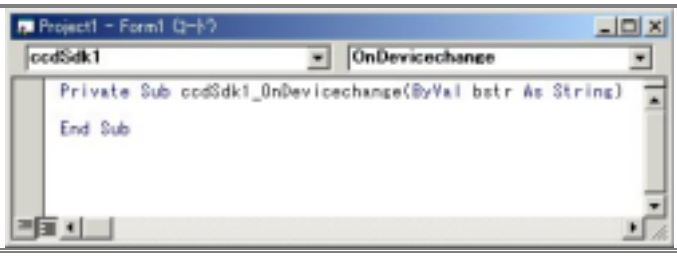
STEP.3

通常のﾎﾞﾀﾝｺﾝﾄﾛｰﾙと同じ要領でﾌｫｰﾑに CCD SDK を貼り付けます。ﾌﾟﾛｸﾞﾗﾑ実行時に CCD SDK が表示されないように、ﾌﾟﾛﾊﾟﾃｨの「Visible」を False に変更します。



STEP.4

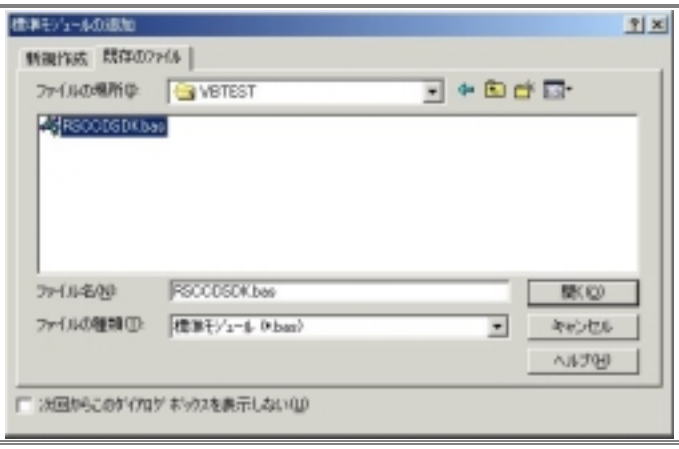
Step.3 のﾌｫｰﾑに貼り付けた「CCDSdk」をﾀﾞﾌﾞﾙｸﾘｯｸすると右のｺｰﾄﾞｳｨﾝﾄﾞｳが表示されます。ここには CCD ｶﾒﾗが取り外されたり、接続されたときのｲﾍﾞﾝﾄ処理を追加します。



STEP.5

CCD SDK CD-ROM の「ForVB¥Module」ﾌｫﾙﾀﾞにある「RSCCDSDK.BAS」をﾌﾟﾛｼﾞｪｸﾄに追加します。

「ﾌﾟﾛｼﾞｪｸﾄ」ﾒﾆｭｰの「標準ﾓｼﾞｭｰﾙの追加」を選択し、「既存のﾌｧｲﾙ」ﾀｸﾞを開きます。ﾌﾟﾛｼﾞｪｸﾄﾌｫﾙﾀﾞに予めｺﾋﾟｰしておいた「RSCCDSDK.BAS」を選択し「開く」ﾎﾞﾀﾝをｸﾘｯｸします。



以上で基本的な準備作業は終了です。

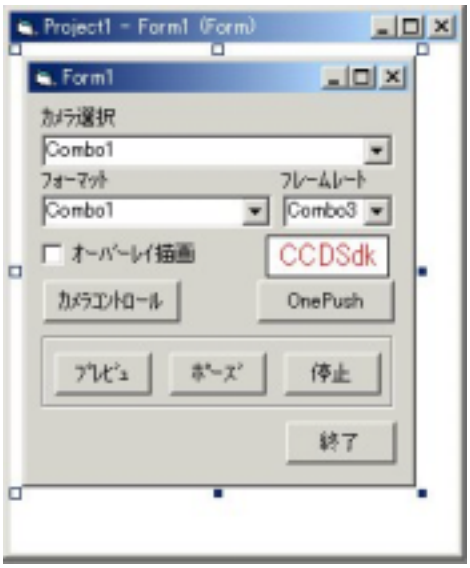
(4-3-2) VB サンプルプログラム解説

EnumCamera ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM\ForVB\Sample\EnumCamera\Form1.frm

EnumCamera ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑは接続されている CCD ｶﾒﾗをｽｷｬﾝし、ｶﾒﾗ選択ｺﾝﾎﾞﾎﾞｯｸｽに CCD ｶﾒﾗのﾕﾆｰｸな識別名称を、ﾌｫｰﾏｯﾄ及びﾌﾚｰﾑﾚｰﾄｺﾝﾎﾞﾎﾞｯｸｽに選択されているｶﾒﾗがｻﾎﾟｰﾄしているﾌｫｰﾏｯﾄ・ﾌﾚｰﾑﾚｰﾄをｾｯﾄします。

また、ｶﾒﾗの取り外しもしくは新たに接続された時に、CCD SDK ActiveX より送られるｲﾍﾞﾝﾄに対する応答処理を行います。ﾌﾟﾚﾋﾞｭｰﾎﾞﾀﾝにより、選択されている CCD ｶﾒﾗのﾌﾟﾚﾋﾞｭｰを行います。

以下に、EnumCamera ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑの骨格について解説します。ﾌﾟﾛｸﾞﾗﾑの詳細についてはｿｰｽﾞｺｰﾄﾞを参照して下さい。



ﾌｫｰﾑがﾛｰﾄﾞされた時の処理

RSCCDSDK.BAS のﾌｧﾝｸｼｮﾝ ccdGetConnectedCameraInfo を呼び出すことにより、tCameraInfo 型構造体に配列 gCameraInfo(MAX_NUMBER_OF_CAMERAS)に接続されている全てのｶﾒﾗの情報がｾｯﾄされます。各ｺﾝﾎﾞﾎﾞｯｸｽのｽﾄﾘｰﾑﾌｫｰﾏｯﾄ情報は gCameraInfo にｾｯﾄされた内容を元にして更新表示することになります。

```
Option Explicit                                          変数未定義ｴﾗｰとする
Option Base 0                                            配列を 0 ｵﾘｼﾞﾝとする
Private Sub Form_Load()
    ccdSdk1.ccdInitialize                                CCD SDK の初期化
    Call Refresh_CameraInfo
End Sub
Public Sub Refresh_CameraInfo()
    Call ccdGetConnectedCameraInfo                       RSCCDSDK.BAS のﾌｧﾝｸｼｮﾝｺｰﾙ
                                                         ｶﾒﾗ情報を取得し gCameraInfo にｾｰﾌﾞ
    Call Refresh_cboCameraName                           ｶﾒﾗ選択ｺﾝﾎﾞの内容作成
    CameraIndex = 0
    cboCameraName.ListIndex = CameraIndex                1 台目のｶﾒﾗを選択

    Status = ccdSdk1.ccdGetCurrentFormat(CameraIndex,    1 台目のｶﾒﾗから現在設定されているｽﾄﾘｰﾑと
                        CurrentStreamIndex, FrameRateIndex)  ﾌﾚｰﾑﾚｰﾄを取得
    Call Refresh_cboFormat(CameraIndex)
    Call Refresh_cboFrameRate(CameraIndex, CurrentStreamIndex)  ﾌｫｰﾏｯﾄｺﾝﾎﾞの内容作成
End Sub                                                  ﾌﾚｰﾑﾚｰﾄｺﾝﾎﾞﾎﾞｯｸｽの内容作成
```

ｶﾒﾗｲﾍﾞﾝﾄの処理

新しくｶﾒﾗが接続させると bstr に文字列"CCD_ATTACH"がｾｯﾄされ ccdSdk1_OnDevicechange の処理が呼び出されます。ｶﾒﾗが取り外された場合は、bstr には"CCD_REMOVE"がｾｯﾄされます。

```
Private Sub ccdSdk1_OnDevicechange(ByVal bstr As String)
    If StrComp(bstr, strCcdAttach, vbTextCompare) = 0 Then   RSCCDSDK.BAS で定義された文字列
                                                             strCcdAttach と比較し一致すれば接続発生
        ccdSdk1.ccdFreeCamera                                全てのﾘｿｰｽを解放
        Call Refresh_CameraInfo                              ｶﾒﾗの列挙処理
        Exit Sub
    ElseIf StrComp(bstr, strCcdRemove, vbTextCompare) = 0 Then   StrCcdRemove と一致すれば取り外し発生
        ccdSdk1.ccdFreeCamera                                全てのﾘｿｰｽを解放
```

```
        Call Refresh_CameraInfo                          ｶﾒﾗの列挙処理
        Exit Sub
    End If
End Sub
```

```
Public Sub Refresh_CameraInfo()
    On Error GoTo ErrorHandler
    Call ccdGetConnectedCameraInfo                       現在接続されている全てのｶﾒﾗ情報を取得し
                                                         gCameraInfo にｾｰﾌﾞ'
                                                         RSCCDSDK.BAS で定義されているﾌｧﾝｸｼｮﾝｺｰﾙ
    Call Refresh_cboCameraName                           ｶﾒﾗｺﾝﾎﾞﾎﾞｯｸｽのﾘｽﾄ内容作成
    CameraIndex = 0
    cboCameraName.ListIndex = CameraIndex

    Status = ccdSdk1.ccdGetCurrentFormat(CameraIndex,    一台目のｶﾒﾗについて現在設定されているﾌｫｰ
                        CurrentStreamIndex, FrameRateIndex)  ﾏｯﾄを取得
    Call Refresh_cboFormat(CameraIndex)                  ﾌｫｰﾏｯﾄｺﾝﾎﾞﾎﾞｯｸｽの内容作成
    cboFormat.ListIndex = CurrentStreamIndex
    Call Refresh_cboFrameRate(CameraIndex, CurrentStreamIndex)  ﾌﾚｰﾑﾚｰﾄｺﾝﾎﾞﾎﾞｯｸｽの内容作成
    cboFrameRate.ListIndex = FrameRateIndex
    Exit Sub
ErrorHandler:
    MsgBox "Refresh camera info failed"
End Sub
```

ｶﾒﾗﾌﾟﾚﾋﾞｭｰの処理

```
Private Sub btnPreview_Click()
    CameraIndex = cboCameraName.ListIndex
    If chkOverlay.Value = 1 Then
        Status = ccdSdk1.ccdSetOverlay(CameraIndex, True)    ｵｰﾊﾞｰﾚｲ描画機能をｲﾈｰﾌﾞﾙ
    Else
        Status = ccdSdk1.ccdSetOverlay(CameraIndex, False)
    End If
    Status = ccdSdk1.ccdPreviewStart(CameraIndex, 0)     ﾌﾟﾚﾋﾞｭｰ開始
End Sub
```

```
Private Sub btnStop_Click()
    CameraIndex = cboCameraName.ListIndex
    Status = ccdSdk1.ccdPreviewStop(CameraIndex)
End Sub
```

WhiteBalance の OnePush 自動調整処理

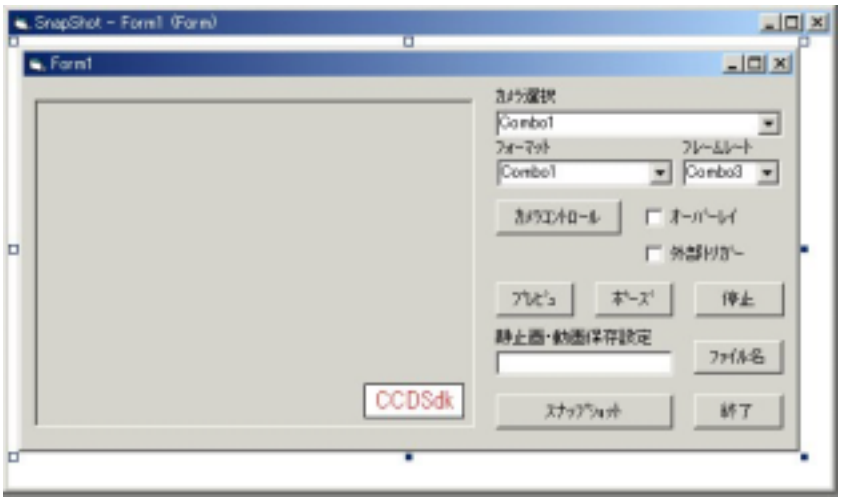
```
Private Sub btnOnePush_Click()
    CameraIndex = cboCameraName.ListIndex
    Status = ccdSdk1.ccdGetColor(CameraIndex, UB, colVal, 0)   現在の U_Value/B_Value 値取得
    Status = ccdSdk1.ccdSetColor(CameraIndex, UB, colVal,      OnePush 自動調整
                                 fCameraControlOnePush)
    Status = ccdSdk1.ccdGetColor(CameraIndex, VR, colVal, 0)   現在の V_Value/R_Value 値取得
    Status = ccdSdk1.ccdSetColor(CameraIndex, VR, colVal,      OnePush 自動調整
                                 fCameraControlOnePush)
End Sub
```

ﾌﾟﾛｸﾞﾗﾑ終了処理

ﾌﾟﾛｸﾞﾗﾑ終了時にはﾌﾟﾚﾋﾞｭｰ等の動作を停止した後に、ｶﾒﾗﾘｿｰｽの解放、CCD SDK の終了処理が必要です。

```
Private Sub Form_Unload(Cancel As Integer)
    CameraIndex = cboCameraName.ListIndex
    Status = ccdSdk1.ccdPreviewStop(CameraIndex)         念の為にﾌﾟﾚﾋﾞｭｰ停止
    ccdSdk1.ccdFreeCamera                                ｶﾒﾗﾘｿｰｽの解放
    ccdSdk1.ccdUninitialize                              CCD SDK の終了処理
    Unload Me
End Sub
```

SnapShot ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM¥ForVB¥Sample¥Snapshot¥Form1.frm



ﾌﾟﾚﾋﾞｭｰ状態で保存ﾌｧｲﾙ名をを拡張子「.bmp」として入力し、「ｽﾅｯﾌﾟｼｮｯﾄ」ﾎﾞﾀﾝを押すと静止画が保存され、同時にﾋﾟｸﾁｬｰﾎﾞｯｸｽに保存したBMPﾌｧｲﾙが表示されます。
ｵｰﾊﾞｰﾚｲと外部ﾄﾘｶﾞｰは両方同時にｲﾈｰﾌﾞﾙにすることはできません。

以下、EnumCameraｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑと異なる点のみを解説します。

ﾌﾟﾚﾋﾞｭｰ実行時の処理

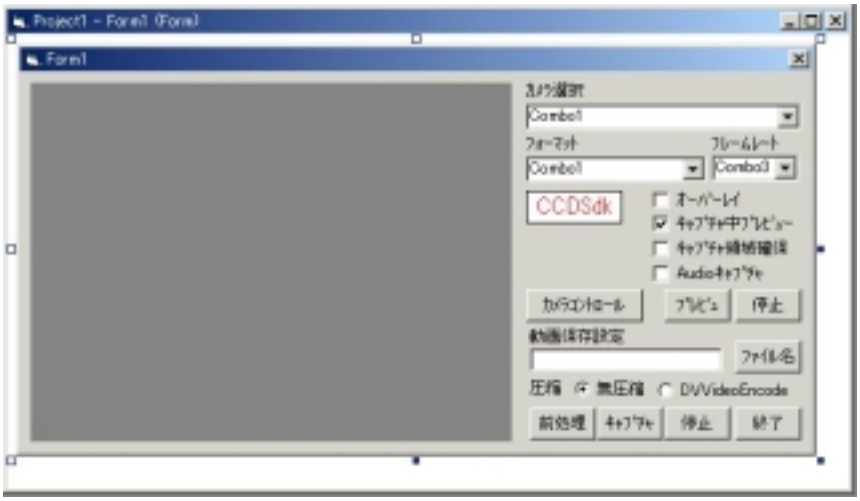
```
Private Sub btnPreview_Click()
    CameraIndex = cboCameraName.ListIndex
    If chkOverlay.Value = 1 Then
        Status = ccdSdk1.ccdSetOverlay(CameraIndex, True)
    Else
        Status = ccdSdk1.ccdSetOverlay(CameraIndex, False)
    End If
    If ChkTrigger.Value = 0 Then
        Status = ccdSdk1.ccdSetTrigger(CameraIndex, False)     外部ﾄﾘｶﾞｰﾓｰﾄﾞ設定
    Else
        Status = ccdSdk1.ccdSetTrigger(CameraIndex, True)
    End If
    Status = ccdSdk1.ccdPreviewStart(CameraIndex, 0)
End Sub
```

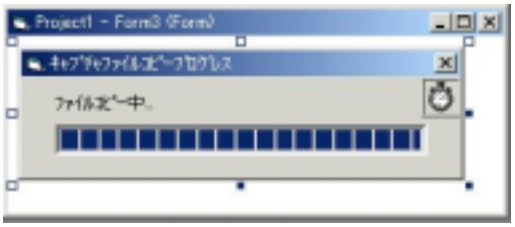
ｽﾅｯﾌﾟｼｮｯﾄ実行時の処理

```
Private Sub btnSnapShot_Click()
    CameraIndex = cboCameraName.ListIndex
    bmpfile = editFileName.Text
    Status = ccdSdk1.ccdSnapshot(CameraIndex, bmpfile)     1 ﾌﾚｰﾑを BMP ﾌｧｲﾙに保存

    ImagePicture.Picture = LoadPicture()
    ImagePicture.Stretch = True
    ImagePicture.Picture = LoadPicture(editFileName)     BMP ﾌｧｲﾙの表示
End Sub
```

Capture ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM¥ForVB¥Sample¥Capture¥Form1.frm

「ﾌﾟﾚﾋﾞｭｰ」を実行するとﾀﾞｲｱﾛｸﾞ内のﾋﾟｸﾁｬｰﾎﾞｯｸｽに動画が表示されます。「ｷｬﾌﾟﾁｬ」ﾎﾞﾀﾝをｸﾘｯｸする前に保存する動画ﾌｧｲﾙ名を入力し「前処理」ﾎﾞﾀﾝをｸﾘｯｸします。
前処理を実行すると「ｷｬﾌﾟﾁｬ領域確保」のﾁｪｯｸﾎﾞｯｸｽにﾁｪｯｸを入れている場合は、予めﾊｰﾄﾞﾃﾞｨｽｸに 200M ﾊﾞｲﾄのﾌｧｲﾙ領域が確保されます。ｷｬﾌﾟﾁｬ実行時、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙに一時的に動画が保存されます。ｷｬﾌﾟﾁｬを停止すると、ﾃﾝﾎﾟﾗﾘﾌｧｲﾙからｷｬﾌﾟﾁｬﾌｧｲﾙ名で指定したﾌｧｲﾙに動画ﾃﾞｰﾀがｺﾋﾟｰされます。





EnumCamera ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑと異なる点のみを解説します。

前処理ﾎﾞﾀﾝを押した時の処理

```
Private Sub btnPreCapture_Click()

    CameraIndex = cboCameraName.ListIndex

    avifile = editFileName.Text
    Status = ccdSdk1.ccdSetCaptureFile                    ｷｬﾌﾟﾁｬﾌｧｲﾙ設定。ｷｬﾌﾟﾁｬ時ﾃﾝﾎﾟﾗﾘﾌｧｲﾙを使用
                          (CameraIndex, avifile, True)    する。

    If chkAllocate.Value <> 0 Then                        ﾃﾝﾎﾟﾗﾘﾌｧｲﾙの設定
        AllocMBytes = 200                                 ﾃﾝﾎﾟﾗﾘﾌｧｲﾙ領域は仮に 200M とする
        tempfile = "c:\temp.avi"
        Status = ccdSdk1.ccdSetCaptureTempFile
                  (CameraIndex, tempfile, True, AllocMBytes)
    Else
        tempfile = ""
        Status = ccdSdk1.ccdSetCaptureTempFile            ﾃﾝﾎﾟﾗﾘﾌｧｲﾙは使用するが、領域確保はせず、
                  (CameraIndex, tempfile, False, 0)       保存場所は環境変数 TEMP で示された場所
    End If

    If OptDvEncode.Value = 1 Then
        Status = ccdSdk1.ccdSetMovieCompressor
                        (CameraIndex, NoCompressor, 0)    圧縮ﾌｫｰﾏｯﾄ指定
    Else
        Compressor = "DV Video Encoder"
        Status = ccdSdk1.ccdSetMovieCompressor
                        (CameraIndex, SetByName, Compressor)
    End If
```

```
    If ChkAudio.Value <> 0 Then                          Audio ｷｬﾌﾟﾁｬの設定
        Status = ccdSdk1.ccdSetAudio(CameraIndex, True)
    Else
        Status = ccdSdk1.ccdSetAudio(CameraIndex, False)
    End If
End Sub
```

■ ｷｬﾌﾟﾁｬｰ実行時の処理

```
Private Sub btnCaptureStart_Click()
    CameraIndex = cboCameraName.ListIndex

    If ChkPreview.Value <> 0 Then                        ｷｬﾌﾟﾁｬｰ実行とｷｬﾌﾟﾁｬｰ時のﾌﾟﾚﾋﾞｭｰ設定
        Status = ccdSdk1.ccdCaptureStart                 ｷｬﾌﾟﾁｬ開始
                    (CameraIndex, True, picVideoWindow.hWnd)
    Else
        Status = ccdSdk1.ccdCaptureStart
                    (CameraIndex, False, picVideoWindow.hWnd)
    End If
End Sub
```

■ ｷｬﾌﾟﾁｬｰ停止時の処理

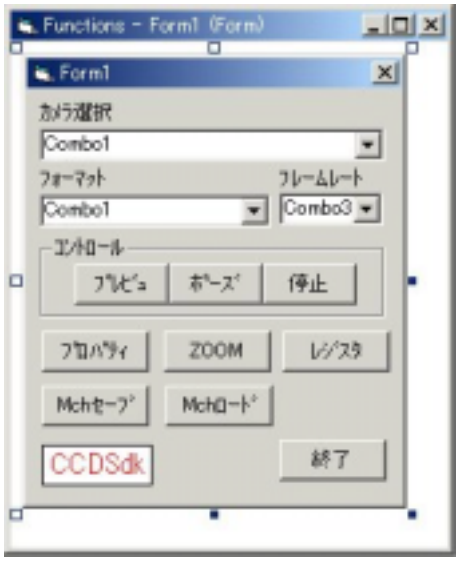
```
Private Sub btnCaptureStop_Click()
    CameraIndex = cboCameraName.ListIndex
    Status = ccdSdk1.ccdCaptureStop(CameraIndex)         ｷｬﾌﾟﾁｬ停止

    Form3.Show                                           ｺﾋﾟｰﾌﾟﾛｸﾞﾚｽﾀﾞｲｱﾛｸﾞ表示
End Sub
```

■ ｺﾋﾟｰﾌﾟﾛｸﾞﾚｽﾀｲﾏｰｲﾍﾞﾝﾄ処理

```
Private Sub CopyTimer_Timer()
    Dim Status As Long
    Dim CopyProgress As Long

    CameraIndex = Form1.cboCameraName.ListIndex
    Status = Form1.ccdSdk1.ccdGetCopyFileProgress        ｺﾋﾟｰ進捗情報取得
                                (CameraIndex, CopyProgress)
    If CopyProgress >= 100 Then                          完了判定
        CopyTimer.Interval = 0
        Unload Form3
    Else
        CopyProgressBar.Value = CopyProgress             ｺﾋﾟｰﾌﾟﾛｸﾞﾚｽ表示更新
    End If
End Sub
```

Functions ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM¥ForVB¥Sample¥Functions¥Form1.frm

その他の CCD SDK ActiveX のｲﾝﾀｰﾌｪｲｽの使い方について解説します。



ｶﾒﾗﾌﾟﾛﾊﾟﾃｨのｺﾝﾄﾛｰﾙ

ZOOM ﾎﾞﾀﾝをｸﾘｯｸすることにより右のﾌｫｰﾑが表示されます。Zoom,Focus のﾌﾟﾛﾊﾟﾃｨ制御方法を例にしたｻﾝﾌﾟﾙです。全てのﾌﾟﾛﾊﾟﾃｨは全く同様の手法でｺﾝﾄﾛｰﾙすることができます。

ccdGetPropertyRange, ccdGetProperty, ccdSetProperty, ccdGetColorRange ccdGetColor, ccdSetColor のｲﾝﾀｰﾌｪｲｽに関するｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑです。


```
Private Sub Form_Load()
    CameraIndex = Form1.cboCameraName.ListIndex
    Status = Form1.ccdSdk1.ccdGetPropertyRange                    ｽﾞｰﾑのﾚﾝｼﾞ取得
                          (CameraIndex,
                          Zoom, _
                          tZoomRange.Min, _
                          tZoomRange.Max, _
                          tZoomRange.SteppingDelta, _
                          tZoomRange.Default, _
                          tZoomRange.Default)
    txtZoomMin = tZoomRange.Min                                   ｽｸﾛｰﾙﾊﾞｰ設定
    txtZoomMax = tZoomRange.Max

    Status = Form1.ccdSdk1.ccdGetProperty                         ｽﾞｰﾑの現在値取得
                      (CameraIndex, Zoom, CurrentValue, 0)
    HScrollZoom.Min = tZoomRange.Min
    HScrollZoom.Max = tZoomRange.Max                              ｽﾞｰﾑｽｸﾛｰﾙﾊﾞｰ設定
    HScrollZoom.Value = CurrentValue
End Sub
```

```
Private Sub HScrollZoom_Change()
    CameraIndex = Form1.cboCameraName.ListIndex
    sVal = HScrollZoom.Value
    Status = Form1.ccdSdk1.ccdSetProperty
                              (lCameraIndex, Zoom, sVal, 0)
End Sub
```

ｽｸﾛｰﾙﾊﾞｰの設定値を取得し Zoom を行う

```
Private Sub btnOnePush_Click()
    CameraIndex = Form1.cboCameraName.ListIndex
    Status = Form1.ccdSdk1.ccdGetColor
                               (lCameraIndex, UB, colVal, 0)
    Status = Form1.ccdSdk1.ccdSetColor
              (lCameraIndex, UB, colVal, fCameraControlOnePush)

    Status = Form1.ccdSdk1.ccdGetColor
                               (lCameraIndex, VR, colVal, 0)
    Status = Form1.ccdSdk1.ccdSetColor
              (lCameraIndex, VR, colVal, fCameraControlOnePush)
End Sub
```

WhiteBalance の OnePush 自動調整を行う

現在の WhiteBalance B-gain(Blue)を取得

B-gain の OnePush 自動調整実行

現在の WhiteBalance R-gain(Red)を取得

R-gain の OnePush 自動調整実行

Camera Control & Status ﾚｼﾞｽﾀ直接ﾘｰﾄﾞﾗｲﾄによるｶﾒﾗｺﾝﾄﾛｰﾙ

ｽﾞｰﾑ制御には「ﾘﾌﾚｯｼｭ」ﾎﾞﾀﾝ実行により ccdGetPropertyRange,ccdGetProperty で取得した現在の Zoom 情報が表示されます。「ﾘｰﾄﾞ」ﾎﾞﾀﾝにより Camera Control & Status ﾚｼﾞｽﾀのｵﾌｾｯﾄ「F0F00880h」にﾏｯﾋﾟﾝｸﾞされている Zoom の Value (Bit20 - 31)が表示されます。「ﾗｲﾄ」ﾎﾞﾀﾝにより「Value Write 値」のｴﾃﾞｨｯﾄﾎﾞｯｸｽに入力した値が書き込まれます。


```
Private Sub btnRead_Click()
    CameraIndex = Form1.cboCameraName.ListIndex
    lOffset = CLng("&H" + txtOffset.Caption)
    Status = Form1.ccdSdk1.ccdReadRegister
                                (CameraIndex, lOffset,
                                readVal)

    txtReadVal = CLng(readVal And &HFFF)
End Sub
```

CSR ｵﾌｾｯﾄ 880h にﾏｯﾋﾟﾝｸﾞされている ZOOM ﾚｼﾞｽﾀを Quadlet ﾘｰﾄﾞ

Bit20-31 の現在値 Value を取得

```
Private Sub btnWrite_Click()
    CameraIndex = Form1.cboCameraName.ListIndex

    lOffset = CLng("&H" + txtOffset.Caption)
    Status = Form1.ccdSdk1.ccdReadRegister
                                (CameraIndex, lOffset,
                                readVal)
    txtReadVal = CLng(readVal And &HFFF)
    writeVal = CLng(txtWriteVal.Text)
    writeVal = (readVal And &HFFFFF000) Or (writeVal And &HFFF)

    Status = Form1.ccdSdk1.ccdWriteRegister
                                (CameraIndex, lOffset,
                                writeVal)
End Sub
```

CSR ｵﾌｾｯﾄ 880h にﾏｯﾋﾟﾝｸﾞされている ZOOM ﾚｼﾞｽﾀを Quadlet ﾘｰﾄﾞ

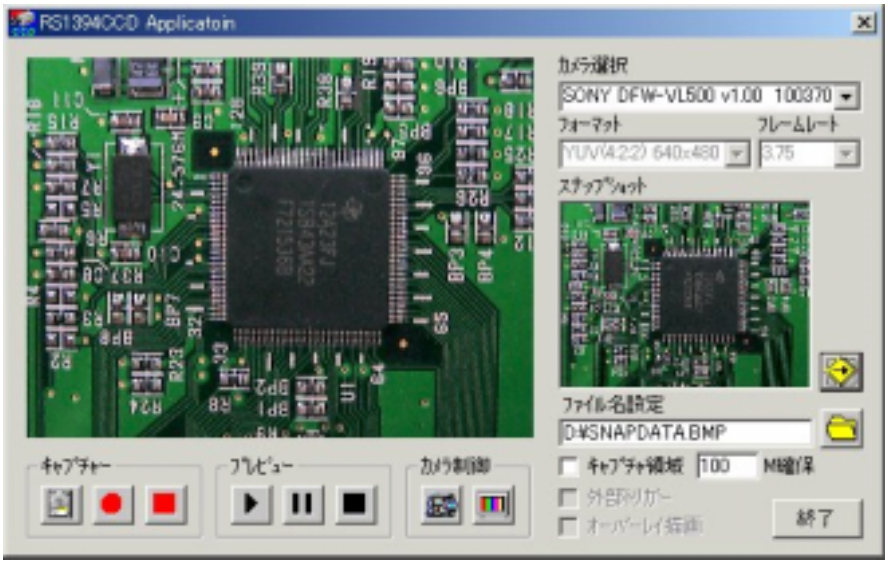
Bit20-31 の現在値 Value を取得
Bit 0-19 はﾘｰﾄﾞした値を、Bit20-31 に入力値をｾｯﾄ

CSR ｵﾌｾｯﾄ 880h にﾏｯﾋﾟﾝｸﾞされている ZOOM ﾚｼﾞｽﾀに Quadlet ﾗｲﾄ

Rsi-CCD ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ：CD-ROM¥ForVB¥Sample¥RSi-CCD¥ RSi-CCD.exe

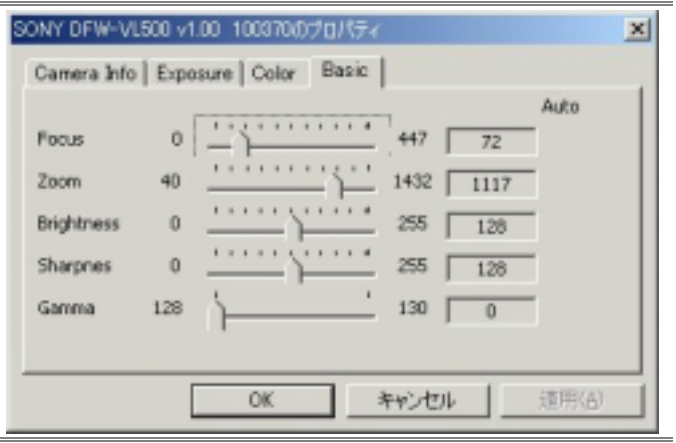
本ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑここまで説明した各ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑの機能を総合的にまとめたものになります。ここでは、ｿｰｽｺｰﾄﾞ詳細説明は省略します。機能詳細は MFC のページを参照願います。

「ﾌﾟﾚﾋﾞｭｰ」ﾎﾞﾀﾝにより左のﾋﾟｸﾁｬｰﾎﾞｯｸｽに CCD ｶﾒﾗからの動画が表示されます。
ﾌｧｲﾙ名を指定して「ｽﾅｯﾌﾟ」ﾎﾞﾀﾝをｸﾘｯｸすることにより右下のﾋﾟｸﾁｬｰﾎﾞｯｸｽに保存されたﾋﾞｯﾄﾏｯﾌﾟﾌｧｲﾙがｽﾄﾚｯﾁ表示されます。
同様に「ｷｬﾌﾟﾁｬｰ」ﾎﾞﾀﾝにより動画が保存されます。



ｶﾒﾗﾌﾟﾛﾊﾟﾃｨのｺﾝﾄﾛｰﾙ

上記ﾒｲﾝｳｨﾝﾄﾞｳの「ｶﾒﾗ調整」ﾎﾞﾀﾝをｸﾘｯｸすると、右のｶﾒﾗﾌﾟﾛﾊﾟﾃｨ調整ﾀﾞｲｱﾛｸﾞが表示されます。



第五章 サポート情報

5-1. FAQ

REX-CFW3H パフォーマンス

REX-CFW3H CardBus カードの転送速度が著しく遅い。例えば、CCD カメラを接続してプレビューを行うと、プレビュー画像が途切れ途切れになる。

カメラドライバキットユーザーズマニュアルで説明している、CBPRSET 補助ドライバのセットアップを行って下さい。

動画のプレビューができない

CCD カメラが接続されているのにも関わらずプレビューとかカメラコントロールができない。

1. CCD カメラのランプがオレンジのままになっていないか確認して下さい。オレンジの場合は一度 1394 ケーブルの挿抜を行って下さい。
2. パソコンを再起動して下さい。
3. REX-CFW3H をご利用の場合は AC アダプタが接続されているか確認して下さい。

オーバーレイ表示が動作しない

オーバーレイをイネーブルにしてプレビューを行い、手動でプレビュー画面の拡大・縮小を行うとパソコンがフリーズする。また、指定のピクチャーウィンドウ内に表示されなかったり、表示された画像に縦線が入ったりする。

一部のグラフィックカードでオーバーレイ表示が正常に動作しないことを確認しています。その場合はオーバーレイ機能をディセーブルにしてご使用頂く必要があります。これは、本製品の問題ではなくグラフィックカードの問題です。

カメラコントロールを行うとカメラの取り外しが発生

CCD カメラのプロパティページより手動で Zoom/Focus 等のプロパティを変更していると、カメラの取り外しが発生する。

動画プレビューを行った状態で Zoom/Focus 等のプロパティを変更すると、1394 のバス上に Isocronous データと Asyncronous データが混在して流れます。頻繁に Asyncronous データを混在させると 1394 ドライバに不具合が発生しカメラの取り外しが起こることがあります。高い頻度で、Zoom/Focus 等のプロパティを変更しないように注意して下さい。

■ ハードディスクへの動画および静止画保存

キャプチャした動画を再生するとフレーム落ちが発生している。静止画を指定したインターバルで保存できない。

動画・静止画ファイルを HDD に保存する場合のパフォーマンス可否について以下の点より総合的に判断する必要があります。

(1) CCD カメラからのフレームサイズを調べる

CCD カメラの設定データフォーマットより 1 フレームのデータサイズは下記のようになります。

ﾌｫｰﾏｯﾄ形式		ﾌﾚｰﾑｻｲｽﾞ	ﾌｫｰﾏｯﾄ	bit/Pixel	ﾌﾚｰﾑｻｲｽﾞ(ﾊﾞｲﾄ)
Format0	Mode0	160×120	YUV (4:4:4)	24	57,600
	Mode1	320×240	YUV (4:2:2)	16	153,600
	Mode2	640×480	YUV (4:1:1)	12	460,800
	Mode2	640×480	YUV (4:2:2)	16	614,400

・・・

(2)ハードディスクのライトパフォーマンス

ディスクパフォーマンスツールにより HDD のライト速度を計測します。下記はフリーソフトウェア HDBench を使って計測した結果です。1 秒間の転送バイトが 12,641(KBytes/s)であることが分かります。尚、HDBench は http://www.hdbench.net よりダウンロードできます。

(3) 1394 バス上のフレーム転送レート

各フレームレートでの 1 フレームの転送に要する時間は下表のようになります。1394 バス上では 30fps の設定が行われている場合、アイソクロナス通信により 125 マイクロ秒毎に 1 パケットデータが送られます。従って、1 フレームの転送に要する時間は 30msec になります。

ﾌﾚｰﾑﾚｰﾄ	30fps	15fps	7.5fps	3.75fps
ﾊﾟｹｯﾄ/ﾌﾚｰﾑ	240	480	960	1920
1ﾌﾚｰﾑ転送時間	30 msec	60 msec	120 msec	240 msec

以上より上記の環境では、1 フレームのデータを保存するのにおおよそ 48.6msec 程度のオーバーヘッドが発生します。従って、フレームレート 30fps では書き込みに遅れが発生し、せいぜい 15fps が限界であると判断できます。

5-2. ユーザサポートについて

RS1394CCD SDK ユーザサポート申し込みに際してはユーザ登録書が必要になります。 ユーザ登録書をお持ちでない場合は、本製品添付の使用許諾書をお読み頂き、ユーザ登録の申し込みを行う必要があります。弊社より返送されたユーザ登録書に記載された期間において、E-mail および FAX にてサポートを行います。

ご質問の内容について

- COM ドライバ・ActiveX ドライバが公開するインターフェイスについての質問
- カメラドライバ・COM ドライバ・ActiveX ドライバの動作・使用方法について

1394 規格に関する内容、Microsoft から提供されているドライバに関する内容、Visual C/C++・Visual Basic の言語使用に関する質問は受け付けられないことがございますので、予めご了承願います。

ご質問方法について

弊社より返送されたユーザ登録書に記載された E-mail アドレスもしくは FAX 番号にご質問をお送り願います。それ以外の場所に質問された場合は基本的に受け付けられません。

ご質問される場合には、ご利用の環境・不具合の内容等詳しく連絡願います。プログラム上の問題等については、可能な範囲で不具合の再現可能なソースコードを添付願います。

ご返答について

弊社５営業日以内に連絡を頂いた E-mail アドレスもしくは FAX 番号にご返送致します。調査に時間を要する場合は別途ご連絡差し上げます。