

OMRON
オムロン株式会社
EMC カンパニー



HVC-C2W

API 仕様書

■改訂履歴

日付	Rev	内容	作成	照査	認可
2015/10/06	A	First release	村上	真鍋	松竹

■商標類

「OKAO」、「OKAO Vision」はオムロン株式会社の登録商標です。

■特記事項

本資料に記載した内容を、書面による当社への確認なしに転載することとはご遠慮ください。

本資料の内容は予告なく変更する場合があります。

目次

1 . ソフトウェア仕様	5
1.1 データ型定義	5
1.2 エラーコード定義	5
1.3 カメラ API 一覧	7
1.3.1 カメラ API 仕様	8
1.4 OKA0 API 一覧	23
1.4.1 OKA0 API 仕様	24
1.5 構造体定義	49
1.6 列挙型定義	59
1.7 コールバック型定義	67

1.ソフトウェア仕様

1.1 データ型定義

本ソフトウェアで利用されるデータ型について以下に示します。

データ型定義	説明
HVCW_CHAR	8bits 文字型
HVCW_UINT8	8bits 符号無文字型
HVCW_BYTE	8bits 符号無文字型
HVCW_BOOL	32bits 符号付整数型
HVCW_INT32	32bits 符号付整数型
HVCW_UINT32	32bits 符号無整数型
HVCW_INT16	16bits 符号付短長整数型
HVCW_UINT16	16bits 符号無短長整数型
HVCW_VOID	型無し

1.2 エラーコード定義

本ソフトウェアで利用されるエラーコードについて以下に示します。

エラーコード定義	説明	値
HVCW_SUCCESS	処理成功	1
HVCW_INVALID_PARAM	パラメータが不正	2
HVCW_NOT_READY	処理準備が出来ていない	3
HVCW_BUSY	処理を受け付け出来ない	4
HVCW_NOT_SUPPORT	要求された処理はサポートされない	5
HVCW_TIMEOUT	通信タイムアウト	6
HVCW_NOT_FOUND	処理対象が見つからない	7
HVCW_NOT_INITIALIZE	SDK が初期化されていない	11
HVCW_DISCONNECTED	カメラとの接続が切断されている	12
HVCW_NOHANDLE	ハンドルエラー	20
HVCW_NO_FACE	顔検出結果が0	30
HVCW_PLURAL_FACES	顔検出結果が複数存在する	31
HVCW_INVALID_RECEIVEDATA	受信データが不正	40
HVCW_FAILURE	その他エラー	8

1.3 コマンドステータス定義

本ソフトウェアで利用されるコマンドステータスについて以下に示します。

コマンドステータス	説明
0	正常終了
0xFF	不正なコマンド
0xFE	不正なデータ長
0xFD	データの内容が不正
0xFC	OKA0 モード時に受けつけられないコマンド
0xFB	スケジューラとイベントプログラムが有効時に ファイル削除、ファイルアップロードを実施
0xDF	ファイルが存在しない
0xDE	ファイル削除失敗
0xCF	SD カード未挿入
0xCE	SD カードの空き容量なし
0xCD	SD カード読み込みエラー
0xCC	SD カード書き込みエラー
0xBF	FLASH 読み込みエラー
0xBE	FLASH 書き込みエラー
0xAF	ドライバエラー
0x9F	システム不備によるエラー
1	顔検出なし
2	複数人検出

1.4 カメラ API 一覧

本ソフトウェアが提供する関数について以下に示します。

関数名	概要
HVCW_Connect	カメラに接続する
HVCW_Disconnect	カメラとの接続を切断する
HVCW_GetCameraVersion	カメラのファームウェアバージョンを取得する
HVCW_UpdateFirmware	カメラにファームウェア更新指示をする
HVCW_SetNightVisionMode	カメラの NightVision モード（手動・自動）を設定する
HVCW_GetNightVisionMode	カメラの NightVision モード（手動・自動）を取得する
HVCW_SetNightVisionStatus	カメラの NightVision ステータス（有効・無効）を設定する
HVCW_GetNightVisionStatus	カメラの NightVision ステータス（有効・無効）を取得する
HVCW_EnableEventMonitor	イベントの受信を開始する
HVCW_DisableEventMonitor	イベントの受信を停止する
HVCW_SetSpeakerVolume	カメラのスピーカーボリュームを設定する
HVCW_GetSpeakerVolume	カメラのスピーカーボリュームを取得する
HVCW_SetMicSensitivity	カメラのマイク感度を設定する
HVCW_GetMicSensitivity	カメラのマイク感度を取得する
HVCW_GetWiFiRSSI	カメラの WiFi-RSSI 値を取得する
HVCW_GenerateDataSoundFile	ネットワーク設定用音声ファイルを作成する
HVCW_RequestStorageFormat	カメラのストレージフォーマットを指示する
HVCW_GetConnectionType	カメラとの接続種別を取得する
HVCW_StartLive	ライブストリーミングを開始する
HVCW_StopLive	ライブストリーミングを停止する
HVCW_FreeDecodedVideoBuffer	ビデオフレームバッファの解放を行う
HVCW_FreeDecodedAudioBuffer	音声データの解放を行う
HVCW_EnterTalkMode	トークモードを開始する
HVCW_ExitTalkMode	トークモードを終了する
HVCW_TransferSoundData	端末マイクからの入力音声をカメラに転送し、 カメラ側にて再生を行う
HVCW_SetVideoResolution	ライブストリーミングの解像度を変更する
HVCW_GetVideoResolution	ライブストリーミングの解像度を取得する
HVCW_EnableSoundDetection	カメラの音声検知を開始する
HVCW_DisableSoundDetection	カメラの音声検知を停止する
HVCW_GetSoundDetection	カメラの音声検知設定を取得する
HVCW_EnableMotionDetection	カメラの動体検知を開始する
HVCW_DisableMotionDetection	カメラの動体検知を停止する
HVCW_GetMotionDetection	カメラの動体検知設定を取得する
HVCW_GetCameraMacAddress	カメラの MAC アドレスを取得する
HVCW_CheckNewFirmware	カメラ FW アップデート確認
HVCW_GetStorageInfo	ストレージ情報取得

1.4.1 カメラ API 仕様

●カメラ接続/切断

HVCW_INT32 HVCW_Connect (HHVC hHVC, HVCW_UINT8 *pucCameraId, HVCW_UINT8 *pucAccessToken)

引数	入力 : hHVC pucCameraId pucAccessToken	HVC ハンドル カメラ識別子 アクセストークン
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー その他エラー
説明	カメラに接続します。 アクセストークンは、システムへのログイン (CGI) にて取得可能です。	

HVCW_INT32 HVCW_Disconnect (HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー SDK 未初期化 その他エラー
説明	カメラとの接続を切断します。	

●バージョン取得

HVCW_INT32 HVCW_GetCameraVersion (HHVC hHVC, HVCW_UINT8 aucVersion[128])

引数	入力 : hHVC 出力 : aucVersion	HVC ハンドル バージョン
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM	正常終了 ハンドルエラー 引数エラー
説明	カメラのファームウェアバージョンを取得します。	

●ファームウェア更新

HVCW_INT32 HVCW_UpdateFirmware (HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_READY HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 ファームウェア最新 SDK 未初期化 その他エラー
説明	カメラにファームウェア更新の指示を行います。	

●カメラの NightVision モード(手動/自動)を設定/取得

HVCW_INT32 HVCW_SetNightVisionMode (HHVC hHVC, HVCW_NIGHT_VISION_MODE mode)

引数	入力 : hHVC mode	HVC ハンドル NightVision モード
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの NightVision モード（手動・自動）を設定します。 NightVision モードを自動に設定した場合、周囲の明るさに応じて自動的に IR LED の On/Off 制御を行います。 NightVision モードを手動に設定した場合、HVCW_SetNightVisionStatus () にて IR LED の On/Off 制御を行ってください。	
入力条件	mode : 0 (自動), 1 (手動)	
初期値	mode = 0 (自動)	

HVCW_INT32 HVCW_GetNightVisionMode (HHVC hHVC, HVCW_NIGHT_VISION_MODE *pMode)

引数	入力 : hHVC 出力 : pMode	HVC ハンドル NightVision モード
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの NightVision モード（手動・自動）を取得します。	

●カメラの NightVision ステータス (有効/無効) を設定/取得

HVCW_INT32 HVCW_SetNightVisionStatus (HHVC hHVC, HVCW_NIGHT_VISION_STATUS status)

引数	入力 : hHVC status	HVC ハンドル NightVision ステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_NOT_SUPPORT HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 モードエラー その他エラー
説明	カメラの NightVision ステータス (有効・無効) を設定します。 NightVision モードが手動の場合のみ設定可能となります。	
入力条件	status : 0 (無効), 1 (有効)	

HVCW_INT32 HVCW_GetNightVisionStatus (HHVC hHVC, HVCW_NIGHT_VISION_STATUS *pStatus)

引数	入力 : hHVC 出力 : pStatus	HVC ハンドル NightVision ステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの NightVision ステータス (有効・無効) を取得します。	

●イベントの受信を開始/停止

HVCW_INT32 HVCW_EnableEventMonitor

(HHVC hHVC, HVCW_VOID *pUserParam, HVCW_EventCallback callbackFunc)

引数	入力 : hHVC pUserParam callbackFunc	HVC ハンドル ユーザパラメータ イベント受信コールバック関数
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー SDK 未初期化 その他エラー
説明	イベント受信用コールバック関数を設定します。 イベントの詳細については HVCW_Event をご参照ください。 コールバック関数の詳細については HVCW_EventCallback をご参照ください。	

HVCW_INT32 HVCW_DisableEventMonitor (HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー SDK 未初期化 その他エラー
説明	イベント受信設定を解除します。	

●カメラのスピーカーボリュームを設定/取得

HVCW_INT32 HVCW_SetSpeakerVolume (HHVC hHVC, HVCW_UINT32 unVolume)

引数	入力 : hHVC unVolume	HVC ハンドル スピーカーボリューム
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラのスピーカーボリュームを設定します。	
入力条件	unVolume : 0~100	
初期値	unVolume = 86	

HVCW_INT32 HVCW_GetSpeakerVolume (HHVC hHVC, HVCW_UINT32 *punVolume)

引数	入力 : hHVC 出力 : punVolume	HVC ハンドル スピーカーボリューム
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラのスピーカーボリュームを取得します。	

●カメラのマイク感度を設定/取得

HVCW_INT32 HVCW_SetMicSensitivity(HHVC hHVC, HVCW_UINT32 unSensitivity)

引数	入力 : hHVC unSensitivity	HVC ハンドル カメラマイク感度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラのマイク感度を設定します。 設定値が大きくなれば感度が高まります。	
入力条件	unSensitivity : 0~9	
初期値	unSensitivity = 9	

HVCW_INT32 HVCW_GetMicSensitivity(HHVC hHVC, HVCW_UINT32 *punSensitivity)

引数	入力 : hHVC punSensitivity	HVC ハンドル カメラマイク感度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラのマイク感度を取得します。	

●カメラのWiFi-RSSI 値を取得

HVCW_INT32 HVCW_GetWiFiRSSI (HHVC hHVC, HVCW_INT32 *pnWifiRssi)

引数	入力 : hHVC 出力 : pnWifiRssi	HVC ハンドル カメラ Wi-Fi RSSI 値
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの WiFi-RSSI 値を取得します。 RSSI 値が大きいほどカメラの受信強度が強いことを意味します。	

●ネットワーク設定用音声ファイルを作成

HVCW_INT32 HVCW_GenerateDataSoundFile (HVCW_UINT8 *pucTargetFile,
HVCW_UINT8 *pucSSID, HVCW_UINT8 *pucPassword, HVCW_UINT8 *pusAccessToken)

引数	入力 : pucTargetFile pucSSID pucPassword pusAccessToken	音声ファイル保存先ファイル ネットワーク名 (SSID) パスワード アクセストークン
戻り値	HVCW_SUCCESS HVCW_INVALID_PARAM HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 引数エラー SDK 未初期化 その他エラー
説明	ネットワーク設定用音声ファイルを作成します。 作成した音声ファイルはアプリケーションにて削除してください。 ネットワーク設定に必要な情報は、ネットワーク名 (SSID)、パスワード、アクセストークンとなります。 アクセストークンは、システムへのログイン (CGI) にて取得可能です。	

●ストレージフォーマット

HVCW_INT32 HVCW_RequestStorageFormat
(HHVC hHVC, HVCW_STORAGE_FORMAT_RESULT_CODE *pResultCode)

引数	入力 : hHVC 出力 : pResultCode	HVC ハンドル フォーマット結果詳細
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラのストレージフォーマットを指示します。 <u>※ストレージに保存されている全てのデータは消去されますのでご注意ください。</u> <u>Flash に保存されている設定ファイルに影響はありません。</u> フォーマット結果詳細は HVCW_StorageFormatResultCode をご参照ください。	

●接続種別を取得

HVCW_INT32 HVCW_GetConnectionType(HHVC hHVC, HVCW_CONNECTION_TYPE *pConnType)

引数	入力 : hHVC 出力 : pConnType	HVC ハンドル 接続種別
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラとの接続種別を取得します。 接続種別は P2P 接続、リレーサーバー経由接続、ローカル LAN 接続のいずれかを取得できます。	

●ライブストリーミング開始/停止

HVCW_INT32 HVCW_StartLive(HHVC hHVC, HVCW_VOID *pUserParam,
HVCW_VIDEO_RESOLUTION videoResolution, HVCW_RequestRenderingCallback renderingCallback,
HVCW_LiveEventCallback eventCallback)

引数	入力 : hHVC pUserParam videoResolution renderingCallback eventCallback	HVC ハンドル ユーザーパラメータ 解像度 ビデオフレーム ライブイベント受信コールバック関数
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_BUSY HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 処理受け付け不可 その他エラー
説明	<p>ライブストリーミングを開始します。</p> <p>複数のカメラとの同時ライブストリーミングは可能ですが、端末の処理負荷によりフレームレートの低下が発生する可能性があります。その場合はアプリケーション側で、解像度の調整を行ってください。</p> <p>デコード済みのビデオフレーム、及び音声データはコールバックにて通知されます。通知したデコード済みデータは、アプリケーションにて描画、及び再生を行う必要があります。ビデオフレーム、音声データの詳細は HVCW_RequestRenderingCallback をご参照ください。</p> <p>通知時点でビデオフレームと音声データは同期されます。</p> <p>ライブストリーミングの解像度はネットワーク状況、及び端末の処理速度によって自動的に切り替わります。</p> <p>ライブイベントはコールバックにて通知されます。</p> <p>ライブイベントの詳細は HVCW_EventCallback をご参照ください。</p>	
入力条件	videoResolution : 0(1280x720), 1(640x360), 2(320x180)	

HVCW_INT32 HVCW_StopLive(HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_NOT_READY HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 ライブストリーミング未開始 その他エラー
説明	ライブストリーミングを停止します。	

●ビデオフレームバッファの解放

HVCW_INT32 HVCW_FreeDecodedVideoBuffer (HHVC hHVC, const HVCW_VOID **pBuffer)

引数	入力 : hHVC pBuffer	HVC ハンドル ビデオフレームバッファ
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM	正常終了 ハンドルエラー 引数エラー
説明	ビデオフレームバッファの解放をします。	

●音声データの解放

HVCW_INT32 HVCW_FreeDecodedAudioBuffer (HHVC hHVC, const HVCW_VOID **pBuffer)

引数	入力 : hHVC pBuffer	HVC ハンドル 音声データバッファ
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM	正常終了 ハンドルエラー 引数エラー
説明	音声データの解放をします。	

●トークモードの開始/終了

HVCW_INT32 HVCW_EnterTalkMode (HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_NOT_READY HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 ライブストリーミング未開始 その他エラー
説明	<p>トークモードを開始します。</p> <p>トークモードとは、ライブストリーミング中に端末マイクから入力された音声をカメラ側にて再生するモードのことです。</p> <p>本関数は、カメラ側で再生した音のループバックを抑止するためのものであるため、トークモードを開始する際は必ず実行してください。</p>	

HVCW_INT32 HVCW_ExitTalkMode (HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_NOT_READY HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 ライブストリーミング未開始 その他エラー
説明	トークモードを終了します。	

●端末マイクからの入力音声をカメラに転送し、カメラ側にて再生を行う

HVCW_INT32 HVCW_TransferSoundData

(HHVC hHVC, HVCW_BYTE *pucSoundData, HVCW_UINT32 unSoundLen)

引数	入力 : hHVC pucSoundData unSoundLen	HVC ハンドル 転送データ 音声データサイズ
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_NOT_READY HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 ライブストリーミング未開始 その他エラー
説明	<p>端末マイクからの入力音声をカメラに転送し、カメラ側にて再生を行います。</p> <p>音声データ転送前に、必ずトークモードに設定してください。再生音声の連続再生を継続的に行う必要がある場合には、アプリケーションにて取得したマイク音声データを速やかに本API を利用して転送する必要があります。音声データの転送はライブストリーミング中のみ可能です。音声データの転送は「1024」バイト単位で行う必要があります。1024バイト未満のデータはゼロサブレスでパディングしてください。</p>	
入力条件	unSoundLen : 1024 (固定)	

●ライブストリーミングの解像度を変更/取得

HVCW_INT32 HVCW_SetVideoResolution(HHVC hHVC, HVCW_VIDEO_RESOLUTION videoResolution)

引数	入力 : hHVC videoResolution	HVC ハンドル 解像度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_NOT_READY HVCW_NOT_SUPPORT HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 ライブストリーミング未開始 指定解像度に変更不可 その他エラー
説明	ライブストリーミングの解像度を変更します。	
入力条件	videoResolution : 0 (1280x720), 1 (640x360), 2 (320x180)	

HVCW_INT32 HVCW_GetVideoResolution(HHVC hHVC, HVCW_VIDEO_RESOLUTION *pVideoResolution)

引数	入力 : hHVC 出力 : pVideoResolution	HVC ハンドル 解像度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_NOT_READY HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 ライブストリーミング未開始 その他エラー
説明	ライブストリーミングの解像度を取得します。	

●カメラの音声検知を開始/終了/取得

HVCW_INT32 HVCW_EnableSoundDetection(HHVC hHVC, HVCW_UINT32 unSensitivity)

引数	入力 : hHVC unSensitivity	HVC ハンドル 音声検知感度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの音声検知を開始します。 設定値が大きくなれば感度が高まります。	
入力条件	unSensitivity : 0~4	

HVCW_INT32 HVCW_DisableSoundDetection(HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの音声検知を終了します。	

HVCW_INT32 HVCW_GetSoundDetection(HHVC hHVC, HVCW_BOOL *pbOn, HVCW_UINT32 *punSensitivity)

引数	入力 : hHVC 出力 : pbOn punSensitivity	HVC ハンドル 有効/無効 音声検知感度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの音声検知機能の状態（有効・無効）、及び感度を取得します。	

●カメラの動体検知を開始/停止

HVCW_INT32 HVCW_EnableMotionDetection(HHVC hHVC, HVCW_UINT32 unDetectionParamsCount,
const HVCW_DETECTIONPARAM aDetectionParams[10])

引数	入力 : hHVC unDetectionParamsCount aDetectionParams	HVC ハンドル 動体検知矩形数 動体検知範囲、感度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	<p>カメラの動体検知を開始します。</p> <p>最大 10 個まで動体検知範囲を設定することができます。</p> <p>sensitivity が大きくなれば感度が高まります。</p> <p>unDetectionParamsCount = 2 とした場合は、 aDetectionParams[0], aDetectionParams[1] が有効となり、それ以降は無視されます。。</p> <p>画像領域外部分を設定した場合は、画像領域外部分のみ無視されます。 座標の指定は 16 の倍数で丸め込まれます。</p>	
入力条件	unDetectionParamsCount : 0~10 aDetectionParams.rect.nX : 0~1919 aDetectionParams.rect.nY : 0~1079 aDetectionParams.rect.nWidth : 1~1920 aDetectionParams.rect.nHeight : 1~1080 aDetectionParams.sensitivity : 0~4	

HVCW_INT32 HVCW_DisableMotionDetection(HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの動体検知を停止します。	

●カメラの動体検知設定を取得

HVCW_INT32 HVCW_GetMotionDetection(HHVC hHVC, HVCW_BOOL *pbOn,
HVCW_UINT32 *punDetectionParamCount, HVCW_DETECTIONPARAM aDetectionParams[10])

引数	入力 : hHVC 出力 : pbOn 入力 : punDetectionParamCount 出力 : aDetectionParams	HVC ハンドル 有効/無効 領域数 動体検知範囲、感度
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	カメラの動体検知機能の状態（有効・無効）、及び感度を取得します。	

●カメラの MAC アドレスを取得

HVCW_INT32 HVCW_GetCameraMacAddress(HHVC hHVC, HVCW_UINT8 aucMACAddress[32])

引数	入力 : hHVC 出力 : aucMACAddress	HVC ハンドル MAC アドレス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM	正常終了 ハンドルエラー 引数エラー
説明	カメラの MAC アドレスを取得します。	

●カメラ FW アップデートを確認

HVCW_INT32 HVCW_CheckNewFirmware(HHVC hHVC, HVCW_UINT8 aucVersion[128])

引数	入力 : hHVC 出力 : aucVersion	HVC ハンドル 最新バージョン情報
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM	正常終了 ハンドルエラー 引数エラー
説明	カメラのファームウェアのアップデートを確認します。 ファームウェアアップデートする最新のバージョンがあった場合はバージョン情報を返します。 カメラが最新バージョンであった場合は aucVersion の中身に NULL を返します。	

●ストレージ情報を取得

HVCW_INT32 HVCW_GetStorageInfo (HHVC hHVC, HVCW_STORAGEINFO *pStorageInfo)

引数	入力 : hHVC 出力 : pStorageInfo	HVC ハンドル ストレージ情報
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_TIMEOUT HVCW_DISCONNECTED HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー 引数エラー 通信タイムアウト 接続切断 SDK 未初期化 その他エラー
説明	ストレージの情報を取得します。	

1.5 OKAO API 一覧

本ソフトウェアが提供する関数について以下に示します。

関数名	概要
HVCW_GetVersion	SDK のバージョンを取得
HVCW_CreateHandle	HVC ハンドル作成
HVCW_DeleteHandle	HVC ハンドル削除
HVCW_SetCameraHandle	カメラハンドルの設定
HVCW_SetAppID	アプリケーション ID 設定
HVCW_GetAppID	アプリケーション ID 取得
HVCW_SetOkaoMode	OKAO モード設定
HVCW_GetOkaoMode	OKAO モード取得
HVCW_GetFileCount	ファイル数取得
HVCW_GetFileInfo	ファイル情報取得
HVCW_DownloadFile	ファイルダウンロード
HVCW_UploadFile	ファイルアップロード
HVCW_DeleteFile	ファイルの削除
HVCW_GetLastOkaoImageSize	最新の OKAO 実行画像サイズ取得
HVCW_GetLastOkaoImage	最新の OKAO 実行画像の取得
HVCW_TakePicture	撮影
HVCW_OKAO_SetThreshold	しきい値設定
HVCW_OKAO_GetThreshold	しきい値取得
HVCW_OKAO_SetSizeRange	検出サイズ設定
HVCW_OKAO_GetSizeRange	検出サイズ取得
HVCW_OKAO_SetDetectionAngle	検出角度の設定
HVCW_OKAO_GetDetectionAngle	検出角度の取得
HVCW_OKAO_Execute	OKAO 実行
HVCW_ALBUM_Register	アルバム登録
HVCW_ALBUM_SetUserName	登録者名変更
HVCW_ALBUM_GetUserName	登録者名取得
HVCW_ALBUM_DeleteData	登録データ削除
HVCW_ALBUM_DeleteAllData	すべての登録情報をクリア
HVCW_ALBUM_GetRegistrationStatus	登録状態取得
HVCW_ALBUM_GetSize	アルバムサイズ取得
HVCW_ALBUM_Download	アルバムのダウンロード
HVCW_ALBUM_Upload	アルバムのアップロード
HVCW_ALBUM_Save	アルバムの Flash 書き込み
HVCW_SetScheduler	スケジュール設定
HVCW_GetScheduler	スケジュール取得
HVCW_SetEventProgram	イベントプログラム設定
HVCW_GetEventProgram	イベントプログラム取得

1.5.1 OKAO API 仕様

●バージョン取得

HVCW_INT32 HVCW_GetVersion

(HVCW_UINT8 *pucMajor, HVCW_UINT8 *pucMinor, HVCW_UINT8 *pucRelease)

引数	出力 : pucMajor pucMinor pucRelease	メジャーバージョン マイナーバージョン リリースバージョン
戻り値	HVCW_SUCCESS HVCW_INVALID_PARAM	正常終了 引数エラー
説明	SDK のバージョンを取得します。	

●ハンドル作成/削除

HHVC HVCW_CreateHandle(void)

引数	なし	
戻り値	not NULL NULL	HVC ハンドル 失敗
説明	HVC ハンドルを作成します。	

HVCW_INT32 HVCW_DeleteHandle(HHVC hHVC)

引数	入力 : hHVC	HVC ハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_FAILURE	正常終了 ハンドルエラー (ハンドル引数の内容が不正) Finalize 失敗
説明	HVC ハンドルを削除します。	

●カメラハンドルの設定

HVCW_INT32 HVCW_SetCameraHandle(HHVC hHVC, C2W_HANDLE hCameraHandle)

引数	入力 : hHVC hCameraHandle	HVC ハンドル C2W カメラハンドル
戻り値	HVCW_SUCCESS HVCW_NOHANDLE	正常終了 ハンドルエラー (ハンドル引数の内容が不正)
説明	HVC ハンドルにカメラハンドルを設定します。 本関数には、C2W ライブラリの C2W_Connect() によりカメラ接続に成功したカメラハンドルを入力してください。 本関数に設定できるカメラハンドルは1つのみです。 hCameraHandle に NULL を設定すると、設定済みカメラハンドルを解除できます。	
初期値	hCameraHandle : NULL	

●アプリケーション ID 設定/取得

HVCW_INT32 HVCW_SetAppID(HHVC hHVC, HVCW_INT32 nAppID, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC nAppID 出力 : pucReturnStatus	HVC ハンドル アプリケーション ID コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー (ハンドル引数の内容が不正) 引数エラー ・ 引数のポインタが NULL ・ 引数値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>アプリケーション ID を設定します。</p> <p>アプリケーション ID を使用すると、アプリケーション毎にカメラに保存される情報 (※1) を管理することができます。 アプリケーション ID は予め一意の ID 番号を取得しておく必要があります。</p> <p>※1) アプリケーション毎に管理する情報 : ログ、画像、イベント用音声データ、通知メッセージ アプリケーション毎に管理されない情報 : 上記以外の全てのパラメータ (顔認証アルバムデータ、OKAO 設定パラメータ、イベントスケジュール、カメラの設定値など)</p> <p>※下記のアプリケーション ID が予約されています。 0～99: 予約</p>	
入力条件	nAppID : -1～INT32_MAX (0x7FFFFFFF)	
初期値	nAppID = -1 (未登録)	

HVCW_INT32 HVCW_GetAppID(HHVC hHVC, HVCW_INT32 *pnAppID, HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC 出力 : pnAppID pucReturnStatus	HVC ハンドル アプリケーション ID コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー (ハンドルの内容が不正) 引数エラー (引数のポインタが NULL) カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	設定されているアプリケーション ID を取得します。	

●OKAO モード設定/取得

HVCW_INT32 HVCW_SetOkaoMode(HHVC hHVC, HVCW_BOOL bOkaoMode, HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC bOkaoMode 出力 : pucReturnStatus	HVC ハンドル OKAO モード設定 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>OKAO モードの有効/無効を設定します。</p> <p>OKAO モードを有効にした場合： ライブラリミニング、イベントプログラムやスケジュール機能が無効になり OKAO 処理のみ行うモードになります。</p> <p>OKAO モードを無効にした場合： 全ての機能が有効になります。</p> <p>TRUE : OKAO モードを有効にする FALSE : OKAO モードを無効にする</p>	
入力条件	bOkaoMode : FALSE bOkaoMode : TRUE	
初期値	bOkaoMode = FALSE	

HVCW_INT32 HVCW_GetOkaoMode(HHVC hHVC, HVCW_BOOL *pbOkaoMode, HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC 出力 : pbOkaoMode pucReturnStatus	HVC ハンドル OKAO モード設定 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	設定されている OKAO モードの状態を取得します。	

●ファイル数取得

HVCW_INT32 HVCW_GetFileCount (HHVC hHVC, HVCW_FILE_EXT fileExt, HVCW_UINT32 *punFileCount, HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC fileExt 出力 : punFileCount pucReturnStatus	HVC ハンドル ファイル拡張子 ファイル数 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	カメラ SD カードに保存されているファイル数を取得します。 fileExt に取得したいファイル拡張子を設定します。 pnFileCount に指定したファイル拡張子のファイル数が格納されます。	
入力条件	fileExt = 2 : ログファイル(log) fileExt = 3 : メッセージテキストファイル(txt) fileExt = 4 : 音声ファイル(wav) fileExt = 6 : 画像ファイル(jpg) fileExt = 8 : サムネイル画像ファイル(jpg)	

●ファイル情報取得

HVCW_INT32 HVCW_GetFileInfo(HHVC hHVC, HVCW_FILE_EXT fileExt,
HVCW_UINT32 unFileIndex, HVCW_FILEINFO *pFileInfo, HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC fileExt unFileIndex 出力 : pFileInfo pucReturnStatus	HVC ハンドル ファイル拡張子 ファイル index ファイル情報 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	カメラ SD カードに保存されているファイルの情報（ファイル名、ファイルサイズ）を取得します。 nFileIndex に情報を取得したいファイルの index を指定します。	
入力条件	fileExt = 2 : ログファイル(log) fileExt = 3 : メッセージテキストファイル(txt) fileExt = 4 : 音声ファイル(wav) fileExt = 6 : 画像ファイル(jpg) fileExt = 8 : サムネイル画像ファイル(jpg)	

●ファイルダウンロード

HVCW_INT32 HVCW_DownloadFile(HHVC hHVC, const HVCW_FILEINFO *pFileInfo,
HVCW_UINT8 *pucBuffer, HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC pFileInfo 出力 : pucBuffer pucReturnStatus	HVC ハンドル ファイル情報 ファイルデータ格納 Buffer コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	カメラ SD カードに保存されているファイルをダウンロードします。 HVCW_GetFileInfo() で取得した pFileInfo を格納してください。 ダウンロードしたファイルは pucBuffer に格納されます。	

●ファイルアップロード

HVCW_INT32 HVCW_UploadFile(HHVC hHVC, HVCW_CHAR acFileName[40], HVCW_INT32 nBufferSize,
const HVCW_UINT8 *pucBuffer, HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC acFileName nBufferSize pucBuffer 出力 : pucReturnStatus	HVC ハンドル ファイル名 バッファサイズ ファイルデータ格納 buffer コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	カメラ SD カードにファイルをアップロードします。 acFileName に使用できる文字列は英数字、[・]（ドット）、[-]（ハイフン）、[_]（アンダーバー）です。上記以外の文字は使用しないでください。 アップロード可能なファイルは以下の通りです。 ログファイル(log) メッセージテキストファイル(txt) 音声ファイル(wav) 画像ファイル(jpg) 画像ファイル(yuv) アップロード可能なメッセージテキストファイルの長さは 127byte です。 スケジューラ、イベントプログラム有効時にはファイルアップロードに失敗します。 スケジューラ、イベントプログラムを無効にしてから本関数を実行してください。	

●ファイルの削除

HVCW_INT32 HVCW_DeleteFile(HHVC hHVC, const HVCW_FILEINFO *pFileInfo,
HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC pFileInfo 出力 : pucReturnStatus	HVC ハンドル ファイル情報 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	カメラ SD カード内のファイルを削除します。 スケジューラ、イベントプログラム有効時にはファイル削除に失敗します。 スケジューラ、イベントプログラムを無効にしてから本関数を実行してください。	

●最新の OKAO 実行時の画像サイズを取得

HVCW_INT32 HVCW_GetLastOkaoImageSize(HHVC hHVC, HVCW_INT32 *pnImgBufSize,
HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC 出力 : pnImgBufSize pucReturnStatus	HVC ハンドル 画像格納バッファサイズ コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>OKAO を実行した際の最新画像の画像格納バッファサイズを取得します。</p> <p>本関数で取得した画像格納バッファサイズは、HVCW_GetLastOkaoImage 関数で取得する画像のバッファサイズです。</p> <p><注意事項> 最新の OKAO 実行時の画像を取得する場合は、必ず本関数により画像格納バッファサイズを取得後、HVCW_GetLastOkaoImage 関数を用いて画像を取得して下さい。</p>	

●最新の OKAO 実行時の画像取得

HVCW_INT32 HVCW_GetLastOkaoImage(HHVC hHVC, HVCW_INT32 nImgBufSize, HVCW_UINT8 *pucImage,
HVCW_UINT8 * pucReturnStatus)

引数	入力 : hHVC nImgBufSize 出力 : pucImage pucReturnStatus	HVC ハンドル 画像格納バッファサイズ 画像格納バッファ コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 画像格納バッファサイズの値が不正 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>OKAO を実行した際の最新画像を取得します。</p> <p>画像フォーマットは JPEG 形式です。 取得に必要なメモリは事前に確保し、本関数を呼び出してください。 取得に必要なメモリサイズは HVCW_GetLastOkaoImageSize にて取得してください。</p> <p><注意事項> 本関数を呼び出す前には、必ず HVCW_GetLastOkaoImageSize を呼び出して下さい。</p>	
入力条件	nImgBufSize : HVCW_GetLastOkaoImageSize() で取得した画像格納バッファサイズ以上	

●撮影

HVCW_INT32 HVCW_TakePicture(HHVC hHVC, HVCW_FILEINFO *pFileInfo,
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC 出力 : pFileInfo pucReturnStatus	HVC ハンドル 撮影した画像ファイル情報 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>画像を撮影します。</p> <p>撮影した画像はカメラの SD カードに保存されます。 保存されたファイルを取得する場合は、HVCW_DownloadFile 関数を使用してファイル を取得してください。</p>	

●しきい値設定/取得

HVCW_INT32 HVCW_OKAO_SetThreshold(HHVC hHVC, const HVCW_OKAO_THRESHOLD *pThreshold,
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC pThreshold 出力 : pucReturnStatus	HVC ハンドル しきい値設定構造体 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>人体検出、手検出、ペット検出、顔検出、顔認証時の検出/認証しきい値を設定します。</p> <p>人体検出、手検出、ペット検出、顔検出： しきい値を大きくした場合、誤検出数は少なくなりますが正解数も少なくなる傾向があります。</p> <p>顔認証： しきい値を大きくした場合、他人を棄却する可能性が高まりますが、本人を受け入れなくなる傾向があります。</p> <p>しきい値設定構造体については HVCW_OKAO_THRESHOLD をご参照ください。</p>	
入力条件	pThreshold.nBody, pThreshold.nHand, pThreshold.nPet, pThreshold.nFace : 1～1000 pThreshold.nRecognition : 0～1000	
初期値	pThreshold.nBody = 500 pThreshold.nHand = 500 pThreshold.nPet = 500 pThreshold.nFace = 500 pThreshold.nRecognition = 500	

HVCW_INT32 HVCW_OKAO_GetThreshold(HHVC hHVC, HVCW_OKAO_THRESHOLD *pThreshold,
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC 出力 : pThreshold pucReturnStatus	HVC ハンドル しきい値 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	設定されている人体検出、手検出、ペット検出、顔検出、顔認証のしきい値を取得します。	

●検出サイズ設定/取得

HVCW_INT32 HVCW_OKAO_SetSizeRange(HHVC hHVC, const HVCW_OKAO_SIZE_RANGE *pSizeRange,
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC pSizeRange 出力 : pucReturnStatus	HVC ハンドル 検出サイズ コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>人体検出、手検出、ペット検出、顔検出の検出サイズを設定します。</p> <p>最小検出サイズ (nMin) を小さく設定し、最大検出サイズ (nMax) を大きく設定すれば、処理時間はかかりますが、多くのサイズの検出が可能となります。</p> <p>検出サイズ設定構造体については HVCW_OKAO_SIZE_RANGE をご参照ください。</p>	
入力条件	nMin : 20～8192 nMax : 20～8192 かつ nMin ≤ nMax	
初期値	pSizeRange.body.nMin = 30 pSizeRange.body.nMax = 8192 pSizeRange.hand.nMin = 40 pSizeRange.hand.nMax = 8192 pSizeRange.pet.nMin = 40 pSizeRange.pet.nMax = 8192 pSizeRange.face.nMin = 64 pSizeRange.face.nMax = 8192	

HVCW_INT32 HVCW_OKAO_GetSizeRange(HHVC hHVC, HVCW_OKAO_SIZE_RANGE *pSizeRange,
HVCW_UINT8 *pucReturnStatus)

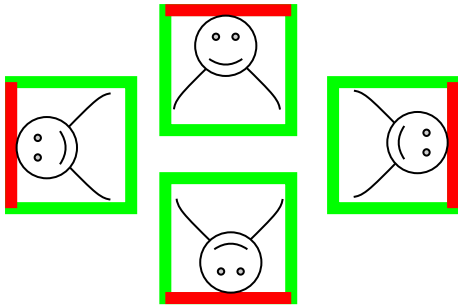
引数	入力 : hHVC 出力 : pSizeRange pucReturnStatus	HVC ハンドル 検出サイズ コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	設定されている人体検出、手検出、ペット検出、顔検出の検出サイズを取得します。	

●検出角度の設定/取得

HVCW_INT32 HVCW_OKAO_SetDetectionAngle (HHVC hHVC,

const HVCW_OKAO_DETECTION_ANGLE *pDetectionAngle, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC pDetectionAngle 出力 : pucReturnStatus	HVC ハンドル 検出角度 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>人体検出、手検出、ペット検出、顔検出の検出角度を設定します。</p> <p>■顔検出 YAW 方向 (nPose)、ROLL 方向 (nAngle) の検出角度の設定が可能です。</p> <p>顔検出の nPose には、以下の 3 シンボルのうち、検出したい Yaw 方向を指定してください。</p> <p>正面方向 : HVCW_POSE_ANGLE_FRONT (0) 正面、左右斜め方向 : HVCW_POSE_ANGLE_HALF_PROFILE (1) 正面、左右斜め、横方向 : HVCW_POSE_ANGLE_PROFILE (2)</p> <p>nAngle には検出する ROLL 方向を指定します。 検出する角度の設定は、下図の 12 のシンボルのうち、検出したい Roll 方向を OR 演算子 () で繋いで指定することができます。</p> <div data-bbox="427 1267 1272 1749" data-label="Diagram"> <p>The diagram illustrates the 12 ROLL angles for face detection. A central face icon is surrounded by 12 radial lines, each pointing to a label: HVCW_ROLL_ANGLE_0 (top), HVCW_ROLL_ANGLE_1 (top-right), HVCW_ROLL_ANGLE_2 (right), HVCW_ROLL_ANGLE_3 (bottom-right), HVCW_ROLL_ANGLE_4 (bottom), HVCW_ROLL_ANGLE_5 (bottom-left), HVCW_ROLL_ANGLE_6 (left), HVCW_ROLL_ANGLE_7 (top-left), HVCW_ROLL_ANGLE_8 (left), HVCW_ROLL_ANGLE_9 (top-left), HVCW_ROLL_ANGLE_10 (top), and HVCW_ROLL_ANGLE_11 (top-left). A red box highlights the face, and a black arrow points upwards towards HVCW_ROLL_ANGLE_0.</p> </div> <p>■ペット検出 ROLL 方向のみ検出角度の設定が可能です。</p> <p>nPet に検出する ROLL 方向を指定します。</p> <p>ペット検出の角度の設定は、顔検出と同じ上図の 12 のシンボルのうち、検出したい Roll 方向を OR 演算子 () で繋いで指定することができます。</p>	

	<p>■人体検出、手検出</p> <p>ROLL 方向のみ検出角度の設定が可能です。 nBody, nHand に検出する ROLL 方向を指定します。</p> <p>人体検出、手検出の角度の設定は、下図の4のシンボルのうち、検出したい Roll 方向を OR 演算子 () で繋いで指定することができます。</p> <div style="text-align: center;"><p>HVCW_ROLL_ANGLE_UP (ROLL_ANGLE_0)</p><p>HVCW_ROLL_ANGLE_LEFT (ROLL_ANGLE_9) HVCW_ROLL_ANGLE_RIGHT (ROLL_ANGLE_3)</p><p>HVCW_ROLL_ANGLE_DOWN (ROLL_ANGLE_6)</p></div> <p>検出角度設定構造体については HVCW_OKAO_DETECTION_ANGLE をご参照ください。</p>
入力条件	<p>人体検出、手検出、ペット検出 nBody, nHand, nPet: 上記の説明欄をご参照ください。</p> <p>顔検出 nPose : HVCW_POSE_ANGLE_FRONT or HVCW_POSE_ANGLE_HALF_PROFILE or HVCW_POSE_ANGLE_PROFILE nAngle : 上記の説明欄をご参照ください。</p>
初期値	<p>pDetectionAngle.nBody = HVCW_ROLL_ANGLE_UP pDetectionAngle.nHand = HVCW_ROLL_ANGLE_UP pDetectionAngle.nPet = HVCW_ROLL_ANGLE_0 pDetectionAngle.nFace.nPose = HVCW_POSE_ANGLE_FRONT pDetectionAngle.nFace.nAngle = HVCW_ROLL_ANGLE_0</p>

HVCW_INT32 HVCW_OKAO_GetDetectionAngle (HHVC hHVC,

HVCW_OKAO_DETECTION_ANGLE *pDetectionAngle, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC 出力 : pDetectionAngle pucReturnStatus	HVC ハンドル 検出角度 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	設定されている人体検出、手検出、ペット検出、顔検出の検出角度を取得します。	

●OKAO 実行

HVCW_INT32 HVCW_OKAO_Execute (HVC hHVC, HVCW_BOOL abUseFunction[HVCW_OkaoFunction_Max],
HVCW_OKAO_RESULT *pResult, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC abUseFunction 出力 : pResult pucReturnStatus	HVC ハンドル 実行機能フラグ 実行結果 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>OKAO の処理を実行します。</p> <p>abUseFunction には実行機能毎の実行フラグを指定してください。 実行機能配列の順番は以下の HVCW_OKAO_FUNCTION をご参照ください。</p> <pre> /* 実行機能インデックス */ typedef enum HVCW_OKAO_FUNCTION { HVCW_OkaoFunction_Body = 0, /*人体検出*/ HVCW_OkaoFunction_Hand, /*手検出*/ HVCW_OkaoFunction_Pet, /*ペット検出*/ HVCW_OkaoFunction_Face, /*顔検出*/ HVCW_OkaoFunction_Direction, /*顔向き推定*/ HVCW_OkaoFunction_Age, /*年齢推定*/ HVCW_OkaoFunction_Gender, /*性別推定*/ HVCW_OkaoFunction_Gaze, /*視線推定*/ HVCW_OkaoFunction_Blink, /*目つむり推定*/ HVCW_OkaoFunction_Expression, /*表情推定*/ HVCW_OkaoFunction_Recognition, /*顔認証*/ HVCW_OkaoFunction_Max }; </pre> <p>実行結果構造体については HVCW_OKAO_RESULT をご参照ください。</p>	
入力条件	abUseFunction : TRUE (1) or FALSE (0)	
初期値	abUseFunction : 全て FALSE (0)	

●アルバム登録

HVCW_INT32 HVCW_ALBUM_Register (HHVC hHVC, HVCW_INT32 nUserID, HVCW_INT32 nDataID,
HVCW_OKAO_RESULT_DETECTION *pFaceResult, HVCW_FILEINFO *pFileInfo,
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC nUserID nDataID 出力 : pFaceResult pFileInfo pucReturnStatus	HVC ハンドル ユーザ ID データ ID 顔検出結果 登録した画像のファイル情報 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NO_FACE HVCW_PLURAL_FACES HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 顔検出結果が 0 顔検出結果が複数存在する ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	アルバムに顔を登録します。 既に登録済みの場合、上書き登録します。 同一人物の特徴量データを複数枚登録する場合、同じユーザーID の異なるデータ ID に登録してください。 顔が検出されない、複数の顔が存在する場合は登録できません。 顔検出結果構造体については HVCW_OKAO_RESULT_DETECTION をご参照ください。	
入力条件	nUserID : 0～499 nDataID : 0～9	

●登録者名変更/取得

HVCW_INT32 HVCW_ALBUM_SetUserName (HHVC hHVC, HVCW_INT32 nUserID, const HVCW_CHAR acName[44],
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC nUserID acName 出力 : pucReturnStatus	HVC ハンドル ユーザ ID ユーザ名 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	登録者の名前を変更します。 アルバムに顔が登録されていない ID に対しても名前変更が可能です。	
入力条件	nUserID : 0～499	
初期値	acName = ""	

HVCW_INT32 HVCW_ALBUM_GetUserName (HHVC hHVC, HVCW_INT32 nUserID, HVCW_CHAR acName[44],
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC nUserID 出力 : acName pucReturnStatus	HVC ハンドル ユーザ ID ユーザ名 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	指定した ID に登録されている名前を取得します。	
入力条件	nUserID : 0～499	

●登録データ削除

HVCW_INT32 HVCW_ALBUM_DeleteData(HHVC hHVC, HVCW_INT32 nUserID, HVCW_INT32 nDataID,
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC nUserID nDataID 出力 : pucReturnStatus	HVC ハンドル ユーザ ID データ ID コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	アルバムから指定ユーザーの指定データを削除します。	
入力条件	nUserID : 0～499 nDataID : 0～9	

●すべての登録情報をクリア

HVCW_INT32 HVCW_ALBUM_DeleteAllData(HHVC hHVC, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC 出力 : pucReturnStatus	HVC ハンドル コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	アルバムに登録されている全てのアルバムデータを削除します。	

●登録状態取得

HVCW_INT32 HVCW_ALBUM_GetRegistrationStatus (HHVC hHVC, HVCW_INT32 nUserID,
HVCW_BOOL abExist[10], HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC nUserID 出力 : abExist pucReturnStatus	HVC ハンドル ユーザ ID 登録位置フラグ コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	指定された指定ユーザーの登録状況を取得します。 nUserID には登録状況を確認したいユーザの ID を設定してください。 登録されている場合は abExist に TRUE を、未登録の場合は FALSE を返します。	
入力条件	nUserID : 0~499	

●アルバムサイズ取得

HVCW_INT32 HVCW_ALBUM_GetSize(HHVC hHVC, HVCW_INT32 *pnAlbumSize,
HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC 出力 : pnAlbumSize pucReturnStatus	HVC ハンドル アルバムサイズ コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	アルバムのサイズを取得します。	

●アルバムのダウンロード

HVCW_INT32 HVCW_ALBUM_Download (HHVC hHVC, HVCW_UINT8 *pucAlbum, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC 出力 : pucAlbum pucReturnStatus	HVC ハンドル アルバムデータ コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	現在カメラが保持しているアルバムをダウンロードします。	

●アルバムのアップロード

HVCW_INT32 HVCW_ALBUM_Upload (HHVC hHVC, HVCW_INT32 nAlbumSize, const HVCW_UINT8 *pucAlbum, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC nAlbumSize pucAlbum 出力 : pucReturnStatus	HVC ハンドル アルバムサイズ アルバム コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	アルバムをカメラ側に保存します。	

●アルバムの Flash 書き込み

HVCW_INT32 HVCW_ALBUM_Save (HHVC hHVC, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC 出力 : pucReturnStatus	HVC ハンドル コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	アルバムデータをカメラの Flash ROM に書き込みます。	

●スケジュール設定/取得

HVCW_INT32 HVCW_SetScheduler (HVC hHVC, HVCW_SCHEDULER_TYPE schedulerType,
HVCW_BOOL bEnable, const HVCW_SCHEDULE_INFO *pSchedule, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC schedulerType bEnable pSchedule 出力 : pucReturnStatus	HVC ハンドル スケジュール スケジュール有効フラグ スケジュール情報 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	スケジュールを設定します。 schedulerType = 0 Onetime スケジュール設定を行うことができます。 Onetime スケジュールは、指定した時間のみ動作します。 schedulerType = 1 Repeat スケジュール設定を行うことができます。 Repeat スケジュールは、指定した時間帯（開始時間～終了時間）で動作します。 スケジュール情報構造体の詳細については HVCW_SCHEDULE_INFO をご参照ください。	
入力条件	schedulerType : 0 (Onetime), 1 (Repeat) bEnable : TRUE or FALSE pSchedule.nIndex : 0～34 (schedulerType = 0 のとき) pSchedule.frequency : 0 (Once), 1 (daily), 2 (weekday) pSchedule.bWeekday[7] : TRUE or FALSE (pSchedule.frequency = 2 のとき) pSchedule.startTime.nMonth : 1～12 pSchedule.startTime.nDay : 1～31 pSchedule.startTime.nHour : 0～23 pSchedule.startTime.nMinute : 0～59 pSchedule.endTime.nMonth : 1～12 pSchedule.endTime.nDay : 1～31 pSchedule.endTime.nHour : 0～23 pSchedule.endTime.nMinute : 0～59 pSchedule.nInterval : 1～86400 (60*60*24) pSchedule.abFunction[HVCW_OkaoFunction_Max] : TRUE or FALSE pSchedule.saveLog : 0 (None), 1 (All), 2 (Detection) pSchedule.saveImage : 0 (None), 1 (All), 2 (Detection)	
初期値	bEnable = FALSE pSchedule.frequency = 0 pSchedule.bWeekday[7] = FALSE pSchedule.startTime.nYear = 2000 pSchedule.startTime.nMonth = 1 pSchedule.startTime.nDay = 1 pSchedule.startTime.nHour = 0 pSchedule.startTime.nMinute = 0	

	pSchedule.endTime.nYear = 2000 pSchedule.endTime.nMonth = 1 pSchedule.endTime.nDay = 1 pSchedule.endTime.nHour = 1 pSchedule.endTime.nMinute = 1 pSchedule.nInterval = 1 pSchedule.abFunction[HVCW_OkaoFunction_Max] = FALSE pSchedule.saveLog = 0 pSchedule.saveImage = 0
--	--

HVCW_INT32 HVCW_GetScheduler (HHVC hHVC, HVCW_SCHEDULER_TYPE schedulerType,
HVCW_BOOL *pbEnable, HVCW_SCHEDULE_INFO *pSchedule, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC scheduler 出力 : pbEnable pSchedule pucReturnStatus	HVC ハンドル スケジュール指定 スケジュール有効フラグ スケジュール情報 コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	設定されているスケジュールを取得します。 OneTimeSchedule の設定値を取得する場合は、pSchedule->nIndex に値を設定してから本関数を実行してください。	
入力条件	schedulerType : 0 (Onetime), 1 (Repeat) pSchedule.nIndex : 0～34	

● イベントプログラム設定/取得

HVCW_INT32 HVCW_SetEventProgram(HHVC hHVC, HVCW_EVENT_PROGRAM_TYPE eventProgramType, HVCW_BOOL bEnable, const HVCW_EVENT_PROGRAM *pEventProgram, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC eventProgramType bEnable pEventProgram 出力 : pucReturnStatus	HVC ハンドル イベントプログラム種別 スケジュール有効フラグ イベントプログラム コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	<p>イベントプログラムを設定します。</p> <p>eventProgramType で音声検知イベント、動体検知イベント、タイマーイベントのうちのどのイベントを設定するか選択してください。</p> <p>※動体検知、音声検知イベントを設定するときの注意事項※ 本関数を実行しても以下は設定されません bEnable pEventProgram->eventProgram-> motionEvent.area.nX pEventProgram->eventProgram-> motionEvent.area.nY pEventProgram->eventProgram-> motionEvent.area.nWidth pEventProgram->eventProgram-> motionEvent.area.nHeight pEventProgram->eventProgram->motionEvent.nSensitivity pEventProgram->eventProgram->soundEvent.nSensitivity</p> <p>動体検知を開始、設定変更を行う際は HVCW_EnableMotionDetection() を実行してください。終了する際は HVCW_DisableMotionDetection() を実行してください。 音声検知を開始、設定変更を行う際は HVCW_EnableSoundDetection() を実行してください。終了する際は HVCW_DisableSoundDetection() を実行してください。</p>	
入力条件	eventProgramType : 0(音声検知), 1(動体検知), 2(タイマー) bEnable : TRUE or FALSE pEventProgram->nIndex : 0~9 (動体検知 設定時のみ) <u><pEventProgram->eventProgram(HVCW_EVENT_PROGRAM_MOTION)></u> motionEvent.area.nX : 0~1919 motionEvent.area.nY : 0~1079 motionEvent.area.nWidth : 1~1920 motionEvent.area.nHeight : 1~1080 motionEvent.nSensitivity : 0~4 motionEvent.unDuration : 1~10 motionEvent.unRatio : 1~100 motionEvent.unOffPeriod : 0~86400 (60*60*24) <u><pEventProgram->eventProgram(HVCW_EVENT_PROGRAM_SOUND)></u> soundEvent.nSensitivity : 0~4 soundEvent.unDuration : 1~10	

	<p>soundEvent.unRatio : 1~100 soundEvent.unOffPeriod : 0~86400 (60*60*24)</p> <p><pEventProgram->eventProgram(HVCW_EVENT_PROGRAM_TIMER)> timerEvent.unInterval : 1~86400 (60*60*24)</p> <p><pEventProgram->eventProgram(共通)> okaoProcess.body.bEnable : TRUE or FALSE okaoProcess.body.nCount : 0~35 okaoProcess.pet.bEnable : TRUE or FALSE okaoProcess.pet.nCount : 0~10 okaoProcess.face.bEnable : TRUE or FALSE okaoProcess.face.nCount : 0~35 okaoProcess.age.bEnable : TRUE or FALSE okaoProcess.age.stAgeRange.nMin : 0~75 okaoProcess.age.stAgeRange.nMax : 0~75 okaoProcess.age.nConfidence : 0~1000 okaoProcess.gender.bEnable : TRUE or FALSE okaoProcess.gender.nGender : 0 (Female), 1 (Male), -1 (無視) okaoProcess.gender.nConfidence : 0~1000 okaoProcess.expression.bEnable : TRUE or FALSE okaoProcess.expression.expression : 0 (無表情), 1 (喜び), 2 (驚き), 3 (怒り), 4 (悲しみ), -1 (無視) okaoProcess.expression.nScore : 0~100 okaoProcess.expression.degreeRange.nMin : -100~100 okaoProcess.expression.degreeRange.nMax : -100~100 okaoProcess.recognition.bEnable : TRUE or FALSE postProcess.saveLog : 0 (None), 1 (All), 2 (Detection) postProcess.saveImage : 0 (None), 1 (All), 2 (Detection) postProcess.notification.pushAlert.bEnable : TRUE or FALSE postProcess.notification.sound.bEnable : TRUE or FALSE frPostProcess.saveImage : 0 (None), 1 (All), 2 (本人), 3 (他人) frPostProcess.notification.pushAlert.knownUser.nUserID : 0~499, -1 (無視) frPostProcess.notification.pushAlert.unknownUser.bEnable : TRUE or FALSE frPostProcess.notification.sound.knownUser.nUserID : 0~499, -1 (無視) frPostProcess.notification.sound.unknownUser.bEnable : TRUE or FALSE</p>
初期値	<p>bEnable = FALSE</p> <p><pEventProgram->eventProgram(HVCW_EVENT_PROGRAM_MOTION)> motionEvent.area.nX = 0 motionEvent.area.nY = 0 motionEvent.area.nWidth = 1920 motionEvent.area.nHeight = 1080 motionEvent.nSensitivity = 3 motionEvent.unDuration = 1 motionEvent.unRatio = 100 motionEvent.unOffPeriod = 0</p> <p><pEventProgram->eventProgram(HVCW_EVENT_PROGRAM_SOUND)> soundEvent.nSensitivity = 3 soundEvent.unDuration = 1 soundEvent.unRatio = 100 soundEvent.unOffPeriod = 0</p> <p><pEventProgram->eventProgram(HVCW_EVENT_PROGRAM_TIMER)> timerEvent.unInterval = 1</p>

	<p><pEventProgram->eventProgram(共通)></p> <pre> okaoProcess.body.bEnable = FALSE okaoProcess.body.nCount = 1 okaoProcess.pet.bEnable = FALSE okaoProcess.pet.nCount = 1 okaoProcess.face.bEnable = FALSE okaoProcess.face.nCount = 1 okaoProcess.age.bEnable = FALSE okaoProcess.age.stAgeRange.nMin = 0 okaoProcess.age.stAgeRange.nMax = 75 okaoProcess.age.nConfidence = 0 okaoProcess.gender.bEnable = FALSE okaoProcess.gender.nGender = -1 okaoProcess.gender.nConfidence = 0 okaoProcess.expression.bEnable = FALSE okaoProcess.expression.expression = -1 okaoProcess.expression.nScore = 0 okaoProcess.expression.degreeRange.nMin = -100 okaoProcess.expression.degreeRange.nMax = 100 okaoProcess.recognition.bEnable = FALSE postProcess.saveLog = 0 postProcess.saveImage = 0 postProcess.notification.pushAlert.bEnable = FALSE postProcess.notification.sound.bEnable = FALSE frPostProcess.saveImage = 0 frPostProcess.notification.pushAlert.knownUser.nUserID = -1 frPostProcess.notification.pushAlert.unknownUser.bEnable = FALSE frPostProcess.notification.sound.knownUser.nUserID = -1 frPostProcess.notification.sound.unknownUser.bEnable = FALSE </pre>
--	---

HVCW_INT32 HVCW_GetEventProgram(HHVC hHVC, HVCW_EVENT_PROGRAM_TYPE eventProgramType,
HVCW_BOOL *pbEnable, HVCW_EVENT_PROGRAM *pEventProgram, HVCW_UINT8 *pucReturnStatus)

引数	入力 : hHVC eventProgramType 出力 : pbEnable pEventProgram pucReturnStatus	HVC ハンドル イベントプログラム種別 スケジュール有効フラグ イベントプログラム コマンドステータス
戻り値	HVCW_SUCCESS HVCW_NOHANDLE HVCW_INVALID_PARAM HVCW_DISCONNECTED HVCW_TIMEOUT HVCW_INVALID_RECEIVEDATA HVCW_NOT_INITIALIZE HVCW_FAILURE	正常終了 ハンドルエラー（ハンドルの内容が不正） 引数エラー ・ 引数のポインタが NULL ・ 設定値が仕様範囲外 カメラ接続エラー 通信タイムアウト 受信データが不正 SDK 未初期化 その他エラー
説明	イベントプログラムを取得します。 動体検知イベントの設定値を取得する場合は、pEventProgram->nIndex に値を設定してから本関数を実行してください。	
入力条件	eventProgramType : 0(音声検知), 1(動体検知), 2(タイマー) pEventProgram->nIndex : 0~9	

1.6 構造体定義

●ビデオフレーム

HVCW_VideoFrame

メンバ	HVCW_BYTE *buffer[3] HVCW_UINT8 stride[3] HVCW_UINT8 width HVCW_UINT8 height	輝度情報、色差情報を格納 各 buffer のストライド情報を格納 横幅 縦幅
説明	YUV420P で出力されるビデオフレームを格納する構造体です。	

●座標点

HVCW_POINT

メンバ	HVCW_INT32 nX HVCW_INT32 nY	X 座標 Y 座標
説明	座標点を格納する構造体です。	

●矩形

HVCW_RECT

メンバ	HVCW_INT32 nX HVCW_INT32 nY HVCW_INT32 nWidth HVCW_INT32 nHeight	起点 x 座標 起点 y 座標 横幅 縦幅
説明	矩形を表現する構造体です。	

●動体検知パラメータ

HVCW_DETECTIONPARAM

メンバ	HVCW_RECT rect HVCW_UINT8 sensitivity	矩形 感度
説明	動体検知の範囲、及び感度を格納する構造体です。	

●スケジュール情報

HVCW_SCHEDULE_INFO

メンバ	<div>HVCW_INT32 nIndex</div> <div>HVCW_SCHEDULE_FREQUENCY frequency</div> <div>HVCW_BOOL bWeekday[7]</div> <div>HVCW_SCHEDULE_TIME startTime</div> <div>HVCW_SCHEDULE_TIME endTime</div> <div>HVCW_INT32 nInterval</div> <div>HVCW_BOOL</div> <div>abFunction[HVCW_OkaoFunction_Max]</div> <div>HVCW_SAVE_RESULT saveLog</div> <div>HVCW_SAVE_RESULT saveImage</div>	<div>Onetime スケジューラのインデックス (Onetime の場合のみ)</div> <div>スケジュールのサイクル</div> <div>曜日設定 (frequency が " weekday " の場合のみ)</div> <div>開始時間</div> <div>終了時間 (Repeat の場合のみ)</div> <div>処理のインターバル</div> <div>OKAO の使用機能フラグ</div> <div>ログの保存方法</div> <div>画像の保存方法</div>												
説明	<p>スケジュール情報を管理する構造体です。</p> <p>nIndex は Onetime スケジューラを設定する場合にのみ有効です。</p> <p>frequency はスケジュールのサイクルを設定します。 0 (Once), 1 (daily), 2 (weekday) の設定が可能です。ScheduleType に応じて下記動作をします。</p> <table border="1"> <thead> <tr> <th></th><th>Onetime</th><th>Repeat</th></tr> </thead> <tbody> <tr> <td>Once</td><td>一回 (開始日時) 実行</td><td>一期間 (開始日時～終了日時) 実行</td></tr> <tr> <td>daily</td><td>毎日決まった時間実行</td><td>毎日決まった時間帯 (開始時間～終了時間) 実行</td></tr> <tr> <td>weekday</td><td>指定曜日の決まった時間実行</td><td>指定曜日の決まった時間帯 (開始時間～終了時間) 実行</td></tr> </tbody> </table> <p>bWeekday はどの曜日に実行するかを指定します。は frequency で " weekday " の場合のみ指定可能です。</p> <p>nInterval は処理を行ってから、次の処理を行うまでの期間を設定します。</p> <p>abFunction は実行する OKAO 機能を設定します。 実行可能な機能については HVCW_OKAO_FUNCTION をご確認ください。</p> <p>saveLog、saveImage はログ、画像を保存する条件を設定します。 下記の設定が可能です。 0 (None : 保存しない) 1 (All : 常に保存する) 2 (Detection : abFunction で指定した機能で検出できれば保存する)</p>			Onetime	Repeat	Once	一回 (開始日時) 実行	一期間 (開始日時～終了日時) 実行	daily	毎日決まった時間実行	毎日決まった時間帯 (開始時間～終了時間) 実行	weekday	指定曜日の決まった時間実行	指定曜日の決まった時間帯 (開始時間～終了時間) 実行
	Onetime	Repeat												
Once	一回 (開始日時) 実行	一期間 (開始日時～終了日時) 実行												
daily	毎日決まった時間実行	毎日決まった時間帯 (開始時間～終了時間) 実行												
weekday	指定曜日の決まった時間実行	指定曜日の決まった時間帯 (開始時間～終了時間) 実行												

●イベントプログラム

HVCW_EVENT_PROGRAM

メンバ	HVCW_INT32 nIndex	動体検知イベントのインデックス (動体検知の場合のみ)
	HVCW_VOID* eventProgram	イベント設定
説明	すべてのイベントプログラムを格納する構造体です。	

●動体検知イベントプログラム

HVCW_EVENT_PROGRAM_MOTION

メンバ	HVCW_EP_MOTION_EVENT motionEvent	動体検知イベント設定
	HVCW_EP_OKAO_PROCESS okaoProcess	OKAO 処理設定
	HVCW_EP_POST_PROCESS postProcess	通知処理設定
	HVCW_EP_FR_POST_PROCESS frPostProcess	FR 専用通知処理設定
説明	動体検知イベントプログラムを格納する構造体です。	

●音声検知イベントプログラム

HVCW_EVENT_PROGRAM_SOUND

メンバ	HVCW_EP_SOUNDEVENT soundEvent	音声検知イベント設定
	HVCW_EP_OKAO_PROCESS okaoProcess	OKAO 処理設定
	HVCW_EP_POST_PROCESS postProcess	通知処理設定
	HVCW_EP_FR_POST_PROCESS frPostProcess	FR 専用通知処理設定
説明	音声検知イベントプログラムを格納する構造体です。	

●タイマーイベントプログラム

HVCW_EVENT_PROGRAM_TIMER

メンバ	HVCW_EP_TIMER_EVENT timerEvent	タイマーイベント設定
	HVCW_EP_OKAO_PROCESS okaoProcess	OKAO 処理設定
	HVCW_EP_POST_PROCESS postProcess	通知処理設定
	HVCW_EP_FR_POST_PROCESS frPostProcess	FR 専用通知処理設定
説明	タイマーイベントプログラムを格納する構造体です。	

●動体検知イベント

HVCW_EP_MOTION_EVENT

メンバ	HVCW_RECT area	動体検知エリア
	HVCW_INT32 nSensitivity	動体検知感度
	HVCW_UINT32 unDuration	動体検知判定時間
	HVCW_UINT32 unRatio	動体検知判定割合
	HVCW_UINT32 unOffPeriod	イベントのインターバル
説明	<p>動体検知イベント時の時間、割合、インターバルなどの情報を格納する構造体です。</p> <p>unDuration で動体検知を判定する間隔（秒）を設定します。</p> <p>unRatio は unDuration 期間中に何割の反応が発生した際に動体を検知したと判定するかの割合を設定します。</p> <p>unOffPeriod で動体を検知した後に次の動体検知が起こるまでのインターバルを設定します。インターバルを長くすると動体検知イベントの発生頻度を低下させることができます。</p>	

●音声検知イベント

HVCW_EP_SOUND_EVENT

メンバ	HVCW_INT32 nSensitivity HVCW_UINT32 unDuration HVCW_UINT32 unRatio HVCW_UINT32 unOffPeriod	音声検知感度 音声検知判定時間 音声検知判定割合 イベントのインターバル
説明	<p>音声検知イベントの時間、割合、インターバルなどの情報を格納する構造体です。</p> <p>unDuration で音声検知を判定する間隔（秒）を設定します。 unRatio は unDuration 期間中に何割の反応が発生した際に音声を検知したと判定するかの割合を設定します。 unOffPeriod で音声を検知した後に次の音声検知が起こるまでのインターバルを設定します。インターバルを長くすると音声検知イベントの発生頻度を低下させることができます。</p>	

●タイマーイベント

HVCW_EP_TIMER_EVENT

メンバ	HVCW_UINT32 unInterval	イベント インターバル
説明	<p>タイマーイベントのイベントのインターバルを格納する構造体です。</p> <p>unInterval で次のイベントが起こるまでのインターバルを設定します。 インターバルを長くするとタイマーイベントの発生頻度を低下させることができます。</p>	

●OKAO 処理

HVCW_EP_OKAO_PROCESS

メンバ	HVCW_EP_BODY body HVCW_EP_PET pet HVCW_EP_FACE face HVCW_EP_AGE age HVCW_EP_GENDER gender HVCW_EP_EXPRESSION expression HVCW_EP_RECOGNITION recognition	人体の判定値 ペットの判定値 顔の判定値 年齢の判定値 性別の判定値 表情の判定値 認証の判定値
説明	イベント処理時に設定する OKAO の判定条件を格納する構造体です。	

●人体検出

HVCW_EP_BODY

メンバ	HVCW_BOOL bEnable HVCW_INT32 nCount	人体検出の実行フラグ 検出数の判定値
説明	<p>bEnable を TRUE にすると人体検出の条件が有効になります。 nCount に設定した数値以上の人体を検出した場合に条件を満たすことになります。</p>	

●ペット検出

HVCW_EP_PET

メンバ	HVCW_BOOL bEnable HVCW_INT32 nCount	ペット検出の実行フラグ 検出数の判定値
説明	<p>bEnable を TRUE にするとペット検出の条件が有効になります。 nCount に設定した数値以上のペットを検出した場合に条件を満たすことになります。</p>	

●顔検出

HVCW_EP_FACE

メンバ	HVCW_BOOL bEnable HVCW_INT32 nCount	顔検出の実行フラグ 検出数の判定値
説明	bEnable を TRUE にすると顔検出の条件が有効になります。 nCount に設定した数値以上の顔を検出した場合に条件を満たすことになります。	

●年齢推定

HVCW_EP_AGE

メンバ	HVCW_BOOL bEnable HVCW_RANGE stAgeRange HVCW_INT32 nConfidence	年齢推定の実行フラグ 年齢の判定値 信頼度しきい値
説明	bEnable を TRUE にすると年齢推定の条件が有効になります。 stAgeRange に設定した数値の年齢であった場合、nConfidence に設定した数値以上の信頼度であった場合に条件を満たすことになります。	

●性別推定

HVCW_EP_GENDER

メンバ	HVCW_BOOL bEnable HVCW_INT32 nGender HVCW_INT32 nConfidence	性別推定の実行フラグ 性別の判定値 信頼度しきい値
説明	bEnable を TRUE にすると性別推定の条件が有効になります。 stAgeRange に設定した数値以上の性別であった場合、nConfidence に設定した数値以上の信頼度であった場合に条件を満たすことになります。	

●表情推定

HVCW_EP_EXPRESSION

メンバ	HVCW_BOOL bEnable HVCW_EXPRESSION expression HVCW_INT32 nScore HVCW_RANGE degreeRange	表情推定の実行フラグ 表情の判定 スコアの判定しきい値 ネガポジの判定値
説明	bEnable を TRUE にすると表情推定の条件が有効になります。 expression に設定した表情が nScore で設定したスコア以上であった場合、またネガポジの値が degreeRange に設定した数値であった場合に条件を満たすことになります。	

●ポスト処理

HVCW_EP_POST_PROCESS

メンバ	HVCW_SAVE_RESULT saveLog HVCW_SAVE_RESULT saveImage HVCW_EP_NOTIFICATION notification	ログの保存 画像の保存 通知設定
説明	イベント設定時のログ、画像保存の条件を設定します。	

●通知設定

HVCW_EP_NOTIFICATION

メンバ	HVCW_EP_PUSH_ALERT pushAlert HVCW_EP_SOUND sound	プッシュ通知設定 音声出力設定
説明	イベント設定時のプッシュ通知、音声出力の条件を設定します。	

●プッシュ通知

HVCW_EP_PUSH_ALERT

メンバ	HVCW_BOOL bEnable HVCW_FILEINFO fileInfo	プッシュ通知フラグ プッシュ通知のメッセージファイル
説明	イベント設定時のプッシュ通知の条件を設定します。 bEnable でプッシュ通知の有効/無効を設定します。 TRUE : 有効 FALSE : 無効 fileInfo に通知メッセージファイルを指定します。	

●音声出力

HVCW_EP_SOUND

メンバ	HVCW_BOOL bEnable HVCW_FILEINFO fileInfo	音声出力フラグ 音声出力の音声ファイル
説明	イベント設定時の音声出力の条件を設定します。 bEnable で音声出力の有効/無効を設定します。 TRUE : 有効 FALSE : 無効 fileInfo に音声ファイルを指定します。	

●顔認証

HVCW_EP_RECOGNITION

メンバ	HVCW_BOOL bEnable	顔認証の実行フラグ
説明	bEnable を TRUE にすると顔認証の条件が有効になります	

●顔認証ポスト処理

HVCW_EP_FR_POST_PROCESS

メンバ	HVCW_FR_SAVE_RESULT saveImage HVCW_EP_FR_NOTIFICATION notification	画像の保存 顔認証通知設定
説明	顔認証時の画像保存、顔認証通知を設定します。	

●顔認証通知設定

HVCW_EP_FR_NOTIFICATION

メンバ	HVCW_EP_FR_PUSH_ALERT pushAlert HVCW_EP_FR_SOUND sound	プッシュ通知設定 音声出力設定
説明	顔認証時のプッシュ通知、音声出力の条件を設定します。	

●顔認証プッシュ通知

HVCW_EP_FR_PUSH_ALERT

メンバ	HVCW_EP_KNOWNUSER knownUser[20] HVCW_EP_PUSH_ALERT unknownUser	登録している人物の場合の処理 登録していない人物の場合の処理
説明	顔認証のプッシュ通知の条件を設定します。 登録している人物の条件、登録していない人物の条件をそれぞれ設定します。	

●顔認証音声出力

HVCW_EP_FR_SOUND

メンバ	HVCW_EP_KNOWNUSER knownUser [20]	登録している人物
	HVCW_EP_SOUND unknownUser	登録していない人物
説明	顔認証の音声出力の条件を設定します。 登録している人物の条件、登録していない人物の条件をそれぞれ設定します。	

●登録ユーザ

HVCW_EP_KNOWNUSER

メンバ	HVCW_INT32 nUserID	ユーザ ID
	HVCW_FILEINFO fileInfo	対象のファイル
説明	fileInfo に音声出力 or プッシュ通知のファイルを指定します。	

●最小最大値設定構造体

HVCW_RANGE

メンバ	HVCW_INT32 nMin	最小値
	HVCW_INT32 nMax	最大値

●検出しきい値設定構造体

HVCW_OKAO_THRESHOLD

メンバ	HVCW_INT32 nBody	人体検出のしきい値
	HVCW_INT32 nHand	手検出のしきい値
	HVCW_INT32 nPet	ペット検出のしきい値
	HVCW_INT32 nFace	顔検出のしきい値
	HVCW_INT32 nRecognition	認証のしきい値

●検出サイズ設定構造体

HVCW_OKAO_SIZE_RANGE

メンバ	HVCW_RANGE body	人体検出の検出サイズ
	HVCW_RANGE hand	手検出の検出サイズ
	HVCW_RANGE pet	ペット検出の検出サイズ
	HVCW_RANGE face	顔検出の検出サイズ

●検出角度設定構造体

HVCW_OKAO_DETECTION_ANGLE

メンバ	HVCW_UINT32 nBody	人体検出の検出角度
	HVCW_UINT32 nHand	手検出の検出角度
	HVCW_UINT32 nPet	ペット検出の検出角度
	HVCW_OKAO_DETECTION_ANGLE_FACE face	顔検出の検出角度

●顔検出角度設定構造体

HVCW_OKAO_DETECTION_ANGLE_FACE

メンバ	HVCW_UINT32 nPose	顔の方向
	HVCW_UINT32 nAngle	顔角度

●検出結果

HVCW_OKAO_RESULT

メンバ	HVCW_OKAO_RESULT_BODYYS bodys	人体の結果
	HVCW_OKAO_RESULT_HANDS hands	手の結果
	HVCW_OKAO_RESULT_PETS pets	ペットの結果
	HVCW_OKAO_RESULT_FACES faces	顔の結果

●人体検出結果

HVCW_OKAO_RESULT_BODYYS

メンバ	HVCW_INT32 nCount	人体検出数
	HVCW_OKAO_RESULT_DETECTION body[35]	人体の結果

●手検出結果

HVCW_OKAO_RESULT_HANDS

メンバ	HVCW_INT32 nCount	手検出数
	HVCW_OKAO_RESULT_DETECTION hand[35]	手の結果

●ペット検出結果

HVCW_OKAO_RESULT_PETS

メンバ	HVCW_INT32 nCount	ペット検出数
	HVCW_OKAO_RESULT_PET pet[10]	ペットの結果

●顔検出結果

HVCW_OKAO_RESULT_FACES

メンバ	HVCW_INT32 nCount	顔検出数
	HVCW_OKAO_RESULT_FACE face[35]	顔の結果

●人体/手検出結果詳細

HVCW_OKAO_RESULT_DETECTION

メンバ	HVCW_POINT center	中心点
	HVCW_INT32 nSize	サイズ
	HVCW_INT32 nConfidence	信頼度

●ペット検出結果詳細

HVCW_OKAO_RESULT_PET

メンバ	HVCW_POINT ptCenter	中心点
	HVCW_INT32 nSize	サイズ
	HVCW_INT32 nConfidence	信頼度
	HVCW_INT32 nPetType	ペットの種類

●顔検出結果詳細

HVCW_OKAO_RESULT_FACE

メンバ	HVCW_POINT center	中心点
	HVCW_INT32 nSize	サイズ
	HVCW_INT32 nConfidence	信頼度
	HVCW_OKAO_RESULT_DIRECTION direction	顔向き推定結果
	HVCW_OKAO_RESULT_AEG age	年齢推定結果
	HVCW_OKAO_RESULT_GENDER gender	性別推定結果
	HVCW_OKAO_RESULT_GAZE gaze	視線推定結果
	HVCW_OKAO_RESULT_BLINK blink	目つむり推定結果
	HVCW_OKAO_RESULT_EXPRESSION expression	表情推定結果
	HVCW_OKAO_RESULT_RECOGNITION recognition	顔認証結果

●角度

HVCW_OKAO_RESULT_DIRECTION

メンバ	HVCW_INT32 nLR	左右角度
	HVCW_INT32 nUD	上下角度
	HVCW_INT32 nRoll	ロール角度

●年齢推定結果

HVCW_OKAO_RESULT_AEG

メンバ	HVCW_INT32 nAge	年齢
	HVCW_INT32 nConfidence	信頼度

●性別推定結果

HVCW_OKAO_RESULT_GENDER

メンバ	HVCW_INT32 nGender	性別
	HVCW_INT32 nConfidence	信頼度

●視線推定結果

HVCW_OKAO_RESULT_GAZE

メンバ	HVCW_INT32 nLR	左右
	HVCW_INT32 nUD	上下角度

●目つむり推定結果

HVCW_OKAO_RESULT_BLINK

メンバ	HVCW_INT32 nLeftEye	左目の目つむり度合
	HVCW_INT32 nRightEye	右目の目つむり度合

●表情推定結果

HVCW_OKAO_RESULT_EXPRESSION

メンバ	HVCW_INT32 anScore[HVCW_Expression_Max]	表情ごとのスコア
	HVCW_INT32 nDegree	ネガティブ/ポジティブ度合

●顔認証スコア

HVCW_OKAO_RESULT_RECOGNITION

メンバ	HVCW_INT32 nUID	ユーザ ID
	HVCW_INT32 nScore	スコア

●ファイル情報

HVCW_FILEINFO

メンバ	HVCW_CHAR acName[40]	ファイル名称
	HVCW_INT32 nSize	ファイルのサイズ
	HVCW_UINT32 Reserved	Reserved

●スケジュール時間

HVCW_SCHEDULE_TIME

メンバ	HVCW_INT32 nYear	年 (4 ケタ)
	HVCW_INT32 nMonth	月
	HVCW_INT32 nDay	日
	HVCW_INT32 nHour	時
	HVCW_INT32 nMinute	分

●ストレージ情報

HVCW_STORAGEINFO

メンバ	HVCW_UINT32 ucTotalSize	ストレージ総容量 (KB)
	HVCW_UINT32 ucUsedSize	使用済みストレージ容量 (KB)
	HVCW_UINT32 ucFreeSize	空きストレージ容量 (KB)
	HVCW_STORAGE_STATUS storageStatus	ストレージステータス

1.7 列举型定義

●ビデオ解像度

```
typedef enum HVCW_VideoResolution{  
    HVCW_VideoResolution_High,  
    HVCW_VideoResolution_Middle,  
    HVCW_VideoResolution_Low  
}HVCW_VIDEO_RESOLUTION;
```

定義名	説明
HVCW_VideoResolution_High	1280x720
HVCW_VideoResolution_Middle	640x360
HVCW_VideoResolution_Low	320x180

●NightVision モード

```
typedef enum HVCW_NightVisionMode{  
    HVCW_NightVisionMode_Auto,  
    HVCW_NightVisionMode_Manual  
}HVCW_NIGHT_VISION_MODE;
```

定義名	説明
HVCW_NightVisionMode_Auto	NightVision の有効・無効を周囲の明るさによって自動制御するモード
HVCW_NightVisionMode_Manual	NightVision の有効・無効を手動設定するモード

●NightVision ステータス

```
typedef enum HVCW_NightVisionStatus{  
    HVCW_NightVisionStatus_Off,  
    HVCW_NightVisionStatus_On  
}HVCW_NIGHT_VISION_STATUS;
```

定義名	説明
HVCW_NightVisionStatus_Off	NightVision 無効
HVCW_NightVisionStatus_On,	NightVision 有効

●イベント

```
typedef enum HVCW_Event{  
    HVCW_Event_ConnectionNum,  
    HVCW_Event_StreamingNum,  
    HVCW_Event_NightVisionMode,  
    HVCW_Event_NightVisionStatus,  
    HVCW_Event_SpeakerVolume,  
    HVCW_Event_Disconnected,  
    HVCW_Event_Reconnected,  
    HVCW_Event_StorageStatu  
} HVCW_EVENT;
```

定義名	説明
HVCW_Event_ConnectionNum	0=カメラへの接続数に変更された場合に通知する。
HVCW_Event_StreamingNum	1=カメラのライブストリーミング数に変更された場合に通知する。
HVCW_Event_NightVisionMode	6=NightVision モードに変更された場合に通知する。
HVCW_Event_NightVisionStatus	7=NightVision ステータスに変更された場合に通知する。
HVCW_Event_SpeakerVolume	9=カメラのスピーカーボリュームに変更された場合に通知する。
HVCW_Event_Disconnected	13=カメラとの接続が切断した場合に通知する。
HVCW_Event_Reconnected	14=切断されたカメラとの接続が回復した場合に通知する。
HVCW_Event_StorageStatus	18=カメラのストレージステータスに変更された場合に通知する。

●ストレージフォーマット結果

```
typedef enum HVCW_StorageFormatResultCode{  
    HVCW_StorageFormatResultCode_Success,  
    HVCW_StorageFormatResultCode_RemovedStorage,  
    HVCW_StorageFormatResultCode_NotSupportStorage,  
    HVCW_StorageFormatResultCode_Timeout,  
    HVCW_StorageFormatResultCode_AlreadyRunning,  
    HVCW_StorageFormatResultCode_CheckDisk,  
    HVCW_StorageFormatResultCode_FatalError  
} HVCW_STORAGE_FORMAT_RESULT_CODE;
```

定義名	説明
HVCW_StorageFormatResultCode_Success	フォーマット成功
HVCW_StorageFormatResultCode_RemovedStorage	フォーマット中にストレージが抜かれた
HVCW_StorageFormatResultCode_NotSupportStorage	サポートしていないストレージ
HVCW_StorageFormatResultCode_Timeout	フォーマット中にタイムアウトが発生
HVCW_StorageFormatResultCode_AlreadyRunning	既にフォーマットを実行中
HVCW_StorageFormatResultCode_CheckDisk	チェックディスク実行中
HVCW_StorageFormatResultCode_FatalError	上記以外のエラー

●接続種別

```
typedef enum HVCW_ConnectionType{  
    HVCW_ConnectionType_P2P,  
    HVCW_ConnectionType_Relay,  
    HVCW_ConnectionType_Local,  
    HVCW_ConnectionType_Disconnect  
}HVCW_CONNECTION_TYPE;
```

定義名	説明
HVCW_ConnectionType_Disconnect	未接続
HVCW_ConnectionType_P2P	P2P 接続
HVCW_ConnectionType_Relay	Relay 接続
HVCW_ConnectionType_Local	Local 接続

●ライブイベント

```
typedef enum HVCW_LiveEvent{
    HVCW_LiveEvent_Started,
    HVCW_LiveEvent_Stopped,
    HVCW_LiveEvent_Disconnected,
    HVCW_LiveEvent_Error,
    HVCW_LiveEvent_FPS,
    HVCW_LiveEvent_ChangeResolution
} HVCW_LIVE_EVENT;
```

定義名	説明	付加情報
HVCW_LiveEvent_Started	ライブストリーミング開始後、ビデオのデコードを開始したタイミングで通知する。	NULL
HVCW_LiveEvent_Stopped	ライブストリーミング停止時に通知する。	NULL
HVCW_LiveEvent_Disconnected	カメラとの接続が切断された場合に通知する。 HVCW_StopLive() を使用してライブストリーミングを停止する必要がある。	NULL
HVCW_LiveEvent_Error	ライブストリーミング継続不能なエラーが発生した場合に通知する。 HVCW_StopLive() を使用してライブストリーミングを停止する必要がある。	NULL
HVCW_LiveEvent_FPS	FPS 情報を通知する。	FPS
HVCW_LiveEvent_ChangeResolution	ライブストリーミングの解像度が変更された場合に通知する。	HVCW_VideoResolution

●スケジューラタイプ設定

```
typedef enum HVCW_ScheduleType{
    HVCW_ScheduleType_OneTime,
    HVCW_ScheduleType_Repeat,
    HVCW_ScheduleType_Max
} HVCW_SCHEDULER_TYPE;
```

定義名	説明
HVCW_ScheduleType_OneTime	Onetime スケジューラ
HVCW_ScheduleType_Repeat	Repeat スケジューラ

●イベントタイプ設定

```
typedef enum HVCW_EventProgramType {
    HVCW_EventProgramType_Sound,
    HVCW_EventProgramType_Motion,
    HVCW_EventProgramType_Timer,
    HVCW_EventProgramType_Max
} HVCW_EVENT_PROGRAM_TYPE;
```

定義名	説明
HVCW_EventProgramType_Sound	音声検知イベント
HVCW_EventProgramType_Motion	動体検知イベント
HVCW_EventProgramType_Timer	タイマーイベント

●スケジュール頻度

```
typedef enum HVCW_ScheduleFrequency{  
    HVCW_ScheduleFrequency_Once,  
    HVCW_ScheduleFrequency_Daily,  
    HVCW_ScheduleFrequency_Weekday,  
    HVCW_ScheduleFrequency_Max  
} HVCW_SCHEDULE_FREQUENCY;
```

定義名	説明
HVCW_ScheduleFrequency_Once	0=一回実行
HVCW_ScheduleFrequency_Daily	1=毎日実行
HVCW_ScheduleFrequency_Weekday	2=指定曜日実行

●画像保存条件

```
typedef enum HVCW_FRSaveResult{  
    HVCW_FRSaveResult_None,  
    HVCW_FRSaveResult_All,  
    HVCW_FRSaveResult_Known,  
    HVCW_FRSaveResult_Unknown,  
    HVCW_FRSaveResult_Max  
} HVCW_FR_SAVE_RESULT;
```

定義名	説明
HVCW_FRSaveResult_None	0=保存なし
HVCW_FRSaveResult_All	1=すべて保存
HVCW_FRSaveResult_Known	2=本人の場合保存
HVCW_FRSaveResult_Unknown	3=他人の場合保存

●画像保存条件

```
typedef enum HVCW_SaveResult{  
    HVCW_SaveResult_None,  
    HVCW_SaveResult_All,  
    HVCW_SaveResult_Detection,  
    HVCW_SaveResult_Max  
}HVCW_SAVE_RESULT;
```

定義名	説明
HVCW_SaveResult_None	0=保存しない
HVCW_SaveResult_All	1=すべて保存
HVCW_SaveResult_Detection	2=OKAO の条件を満たした場合に保存

●ファイル種別

```
typedef enum HVCW_FileExt{  
    HVCW_FileExt_Log,  
    HVCW_FileExt_MessageText,  
    HVCW_FileExt_Sound,  
    HVCW_FileExt_JpgImage,  
    HVCW_FileExt_YUVImage,  
    HVCW_FileExt_ThumbnailJpgImage  
}HVCW_FILE_EXT;
```

定義名	説明
HVCW_FileExt_Log	2=ログファイル(log)
HVCW_FileExt_MessageText	3=メッセージテキストファイル(txt)
HVCW_FileExt_Sound	4=音声ファイル(wav)
HVCW_FileExt_JpgImage	6=画像ファイル(jpg)
HVCW_FileExt_YUVImage	7=画像ファイル(yuv)
HVCW_FileExt_ThumbnailJpgImage	8=サムネイル画像ファイル(jpg)

●OKAO 機能

```
typedef enum HVCW_OkaoFunction{
    HVCW_OkaoFunction_Body,
    HVCW_OkaoFunction_Hand,
    HVCW_OkaoFunction_Pet,
    HVCW_OkaoFunction_Face,
    HVCW_OkaoFunction_Direction,
    HVCW_OkaoFunction_Age,
    HVCW_OkaoFunction_Gender,
    HVCW_OkaoFunction_Gaze,
    HVCW_OkaoFunction_Blink,
    HVCW_OkaoFunction_Expression,
    HVCW_OkaoFunction_Recognition,
    HVCW_OkaoFunction_Max
} HVCW_OKAO_FUNCTION;
```

定義名	説明
HVCW_OkaoFunction_Body	0=人体検出
HVCW_OkaoFunction_Hand	1=手検出
HVCW_OkaoFunction_Pet	2=ペット検出
HVCW_OkaoFunction_Face	3=顔検出
HVCW_OkaoFunction_Direction	4=顔向き推定
HVCW_OkaoFunction_Age	5=年齢推定
HVCW_OkaoFunction_Gender	6=性別推定
HVCW_OkaoFunction_Gaze	7=視線推定
HVCW_OkaoFunction_Blink	8=目つむり推定
HVCW_OkaoFunction_Expression	9=表情推定
HVCW_OkaoFunction_Recognition	10=顔認証

●表情種別

```
typedef enum HVCW_Expression{
    HVCW_Expression_Neutral,
    HVCW_Expression_Happiness,
    HVCW_Expression_Surprise,
    HVCW_Expression_Anger,
    HVCW_Expression_Sadness,
    HVCW_Expression_Max
} HVCW_EXPRESSION;
```

定義名	説明
HVCW_Expression_Neutral	0=表情：無表情
HVCW_Expression_Happiness	1=表情：喜び
HVCW_Expression_Surprise	2=表情：驚き
HVCW_Expression_Anger	3=表情：怒り
HVCW_Expression_Sadness	4=表情：悲しみ
HVCW_Expression_Max	

●ストレージステータスの列挙型定義

```
typedef enum HVCW_StorageStatus{  
    HVCW_StorageStatus_NotInsert,  
    HVCW_StorageStatus_NotReady,  
    HVCW_StorageStatus_NeedFormat,  
    HVCW_StorageStatus_Normal,  
    HVCW_StorageStatus_Error,  
    HVCW_StorageStatus_Formatting,  
    HVCW_StorageStatus_NotSupport  
} HVCW_STORAGE_STATUS;
```

定義名	説明
HVCW_StorageStatus_NotInsert	0=ストレージ未検出
HVCW_StorageStatus_NotReady	1=ストレージ検出済みだが使用不可状態
HVCW_StorageStatus_NeedFormat	2=ストレージはフォーマットが必要な状態
HVCW_StorageStatus_Normal	3=ストレージ使用可能
HVCW_StorageStatus_Error	4=ストレージにエラーが発生し使用不可能状態
HVCW_StorageStatus_Formatting	6=ストレージフォーマット中
HVCW_StorageStatus_NotSupport	7=サポート対象外ストレージ

1.8 コールバック型定義

●イベント通知

HVCW_INT32 (*HVCW_EventCallback) (HVCW_INT32 nEventId, HVCW_VOID *pUserParam,
HVCW_VOID *pEventInfo)

引数	入力 : nEventId pUserParam pEventInfo	イベント ID ユーザーパラメータ イベント付加情報
戻り値	HVCW_SUCCESS	正常終了
説明	イベント通知用コールバック関数です。	

●レンダリング要求

HVCW_INT32 (*HVCW_RequestRenderingCallback) (HVCW_BOOL bVideo, HVCW_VOID *pUserParam,
HVCW_VOID *pRenderInfo, HVCW_UINT32 unInfoLen, HVCW_UINT32 unTimeStamp)

引数	入力 : bVideo pUserParam pRenderInfo unInfoLen unTimeStamp	HVCW_TRUE : ビデオ描画要求 HVCW_FALSE : 音声データ再生要求 ユーザーパラメータ ビデオフレーム、又は音声データ renderInfo サイズ タイムスタンプ
戻り値	HVCW_SUCCESS	正常終了
説明	ライブストリーミング中にビデオフレームの描画、及び音声データの再生を要求するコールバック関数です。	

●ライブイベント通知

HVCW_INT32 (*HVCW_LiveEventCallback) (HVCW_INT32 nEventId, HVCW_VOID *pUserParam,
HVCW_VOID *pEventInfo)

引数	入力 : nEventId pUserParam pEventInfo	イベント ID ユーザーパラメータ イベント付加情報
戻り値	HVCW_SUCCESS	正常終了
説明	ライブイベント通知用コールバック関数です。	