
Title: SX UVL Client SDK for Win32

Drawing Type: 機能仕様書

Drawing No.: SC02790XM

Date: April 16, 2020

Rev (版数)	Description (内容)	Date (日付)
XX	初版	05.07.13
XA	第 2 版 誤記修正	05.09.08
XB	第 3 版 API 追加	07.01.17
XC	第 4 版 API 追加	09.09.30
XD	第 5 版 SXUSBDEVICE 構造体変更	10.11.12
XE	第 6 版 対応 OS 追加	13.04.12
XF	第 7 版 システムステータス取得 API 追加 検索 API ホスト名の取得に対応	13.06.13
XG	第 8 版 対応 OS 追加	13.12.10
XH	第 9 版 切断要求 API 追加	13.12.18
XJ	第 10 版 インスタンス ID 取得 API 追加	14.03.19
XK	第 11 版 対応 OS 追加	15.07.21
XL	第 12 版 DS-600 対応	16.07.15
XM	第 13 版 DS-700 対応	20.04.16

1. 概要	5
2. 仕様	5
2.1. 動作環境	5
3. デバイスサーバ通信 API 仕様	6
3.1. デバイスサーバ検索処理	6
3.2. USB デバイス一覧取得処理	7
3.3. USB デバイス一覧取得処理	8
3.4. デバイス接続処理	10
3.5. デバイス接続処理	10
3.6. デバイス接続処理	11
3.7. デバイス接続処理	12
3.8. デバイス切断処理	13
3.9. デバイス切断処理	13
3.10. デバイス切断処理	14
3.11. ステータス取得処理	14
3.12. ステータス取得処理	15
3.13. デバイスサーバ情報取得処理	16
3.14. デバイスサーバ情報取得処理	16
3.15. デバイスサーバ検索ルール No.設定処理	17
3.16. デバイスサーバ再起動処理	17
3.17. デバイスサーバ再起動処理	18
3.18. システムステータス取得処理	18
4. 切断要求 API	19
4.1. 切断要求 API 初期化处理	19
4.2. 切断要求 API 終了処理	20
4.3. 切断要求の受信応答処理	20
4.4. 切断要求の許可応答処理	21
4.5. 切断要求の拒否応答処理	21
4.6. 切断処理の完了通知処理	21
4.7. 切断要求の処理	21
4.8. 切断要求の送信処理	22
4.9. 切断要求の送信処理	23
4.10. 切断要求のキャンセル処理	23
4.11. 切断要求ハンドルのクローズ処理	24
4.12. SRU_NOTIFY メッセージ	25
4.13. 切断要求メッセージ	26
5. デバイスドライバ API	27
5.1. インスタンス ID 取得処理	27

5.2.	インスタンス ID 取得処理	28
5.3.	インスタンス ID 解放処理	29

1. 概要

本仕様書は、サイレックス・テクノロジーのデバイスサーバを管理するユーティリティ **SX Virtual Link** の GUI 部を除く基本機能をユーザアプリケーションで制御を行うために開発された DLL の使用方法について記述します。

2. 仕様

2.1. 動作環境

XE
XG
XK
XM

項目	説明
OS	<p>Microsoft Windows 2000</p> <p>Microsoft Windows XP (32bit / 64bit)</p> <p>Microsoft Windows Vista (32bit / 64bit)</p> <p>Microsoft Windows 7 (32bit / 64bit)</p> <p>Microsoft Windows 8 (32bit / 64bit)*</p> <p>Microsoft Windows 8.1 (32bit / 64bit)*</p> <p>Microsoft Windows 10 (32bit / 64bit)*</p> <p>※クラシックデスクトップ用アプリケーションに対応しています。Modern UI 用アプリケーションには対応していません。</p> <p>※Windows RT には対応していません。</p>
コンパイラ	<p>Microsoft Visual C++ 6.0 SP6</p> <p>Microsoft Visual Studio 2005</p> <p>Microsoft Visual Studio 2008</p> <p>Microsoft Visual Studio 2010</p> <p>Microsoft Visual Studio 2012</p> <p>Microsoft Visual Studio 2013</p> <p>Microsoft Visual Studio 2017</p> <p>Microsoft Visual Studio 2019</p>

3. デバイスサーバ通信 API 仕様

3.1. デバイスサーバ検索処理

SxuptEnumDeviceServers

SxuptEnumDeviceServers は、指定されたブロードキャストアドレスに対して検索を行い、ネットワーク上のデバイスサーバを列挙します。

BOOL SxuptEnumDeviceServers(

```
LPDWORD lpdwBroadcasts, /* ブロードキャストアドレスの配列のアドレス */
DWORD dwCount, /* ブロードキャストアドレスの個数 */
LPVOID lpbServers, /* 構造体の配列のアドレス */
DWORD cbBuf, /* 配列のバイト数 */
LPDWORD lpdwReaded, /* コピーされたバイト数が返される変数のアドレス */
LPDWORD lpdwReturned /* コピーされた構造体の個数が返される変数のアドレス */
);
```

パラメータ	意味
lpdwBroadcasts	ブロードキャストアドレスの配列へのポインタです。この変数には、デバイスサーバを検索するブロードキャストアドレスを指定します。ブロードキャストアドレスは、DWORD 値の配列として複数指定することができます。NULL を指定した場合は、ローカルアドレス (255.255.255.255) のデバイスサーバを検索します。 本パラメータは、IPv4 用のパラメータです。
dwCount	lpdwBroadcasts で指定したブロードキャストアドレスの個数を指定します。lpdwBroadcasts を NULL で指定した場合は、この値は無視されます。
lpbServers	SXPSEVER 構造体の配列を示すポインタです。デバイスサーバ検索結果を受け取ります。
cbBuf	lpbServers が指す配列のサイズをバイト単位で指定します。
lpdwReaded	lpbServers にコピーされたバイト数を示します。
lpdwReturned	検索されたデバイスサーバの個数が返される変数を指すポインタです。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

備考

SXPSEVER 構造体

```
typedef struct _SXPSEVER {
    BYTE bNodeaddr[6]; /* Ethernet アドレス */
    DWORD dwIp; /* IPv4 アドレス */
    BYTE bIpAddress[16]; /* IPv6 アドレス */
    char szMachineType[16]; /* デバイスサーバ機種名 */
    char szHostName[16]; /* ホスト名 */
} SXPSEVER, *LPSXPSEVER;
```

XF
XM

3.2. USB デバイス一覧取得処理

SxuptpEnumDevices

SxuptpEnumDevices は、指定したデバイスサーバに接続されている USB デバイスの一覧を列挙します。

BOOL SxuptpEnumDevices(

DWORD dwIpaddr, /* デバイスサーバの IP アドレス */

LPVOID lpbDevices, /* 構造体の配列のアドレス */

DWORD cbBuf, /* 配列のバイト数 */

LPDWORD lpdwReaded, /* コピーされたバイト数が返される変数のアドレス */

LPDWORD lpdwReturned /* コピーされた構造体の個数が返される変数のアドレス */

);

パラメータ	意味
dwIpaddr	USB デバイスを列挙するデバイスサーバの IP アドレスを指定します。
lpbDevices	SXUSBDEVICE 構造体の配列を示すポインタです。USB デバイス一覧を受け取ります。
cbBuf	lpbDevices が指す配列のサイズをバイト単位で指定します。
lpdwReaded	lpbDevices にコピーされたバイト数を示します。
lpdwReturned	列挙された USB デバイスの個数が返される変数を指すポインタです。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XM

3.3. USB デバイス一覧取得処理

SxuptpEnumDevicesV6

SxuptpEnumDevicesV6 は、指定したデバイスサーバに接続されている USB デバイスの一覧を列挙します。この関数は IPv4/IPv6 通信に対応しています。

BOOL SxuptpEnumDevicesV6(

```
LPBYTE lpbIpAddr, /* デバイスサーバの IP アドレス */
int iIpAddrLen, /* IP アドレス長 */
LPVOID lpbDevices, /* 構造体の配列のアドレス */
DWORD cbBuf, /* 配列のバイト数 */
LPDWORD lpdwReaded, /* コピーされたバイト数が返される変数のアドレス */
LPDWORD lpdwReturned /* コピーされた構造体の個数が返される変数のアドレス */
);
```

パラメータ	意味
lpbIpAddr	USB デバイスを列挙するデバイスサーバの IP アドレスを指定します。IPv4 アドレス、IPv6 アドレスを指定することができます。
iIpAddrLen	指定した IP アドレスの長さを指定します。
lpbDevices	SXUSBDEVICE 構造体の配列を示すポインタです。USB デバイス一覧を受け取ります。
cbBuf	lpbDevices が指す配列のサイズをバイト単位で指定します。
lpdwReaded	lpbDevices にコピーされたバイト数を示します。
lpdwReturned	列挙された USB デバイスの個数が返される変数を指すポインタです。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

備考

XD

SXUSBDEVICE 構造体

XE

typedef struct _SXUSBDEVICE {

XL

char szPortName[32]; /* USB 論理ポート名 */

XM

char szDeviceName[64]; /* デバイス名 */

DWORD dwIpPc; /* 接続先 PC の IPv4 アドレス */

BYTE bIpAddrPc[16]; /* 接続先 PC の IPv6 アドレス */

WORD wStatus; /* ステータス情報 */

WORD wVid; /* デバイスの VID */

WORD wPid; /* デバイスの PID */

char szLocation[16]; /* デバイス接続位置情報 */

WORD wClass; /* USB Interface Class 情報 */

} SXUSBDEVICE, *LPSXUSBDEVICE;

パラメータ	意味
szPortName	サーバの USB 論理ポート名。接続・切断時に必要なパラメータです。
szDeviceName	USB デバイスの名称。最大 63 文字となります。
dwIpPc	デバイスを専有している PC の IPv4 アドレス。未接続時は 0.0.0.0 となります。
bIpAddrPc	デバイスを専有している PC の IPv6 アドレス。未接続時は 0 クリアされます。

wStatus	デバイス状態を下記の BIT 状態で示します。	
	0x8000	
	0x4000	このビットが low の場合は、デバイスサーバと USB デバイス間で USB 通信を行っています。
	0x2000	暗号通信中。
	0x1000	暗号化通信対応製品は high となります。
	0x0800	無線 LAN 動作中。有線 LAN 製品は low 固定。
	0x0400	アイソクロナス転送に対応したエンドポイントを有する USB デバイス。
	0x0200	USB 異常発生。
	0x0100	
	0x0080	high 固定。
	0x0040	low 固定。
	0x0020	(未使用)
	0x0010	接続パスワードが有効。USB 接続を接続するには、パスワード認証が必要です。
	0x0008	接続パスワード対応製品は high となります。
	0x0004	
	0x0002	
	0x0001	USB デバイス検出あり。
wVid	USB デバイスの VID を格納します。	
wPID	USB デバイスの PID を格納します。	
szLocation	接続されているデバイスの位置情報を示す(NULL ターミネイト文字)。先頭 1 バイトに接続スピードを示す文字(H : High/F : Full/L:Low)に続き、ホストコントローラデバイス番号、ルートハブポート番号、ハブポート番号を順に示します。各番号は文字 0~9,A~F を使用します。ホストコントローラデバイス番号 1+ROOT HUB ポート番号 1+HUB ポート 1 に接続されている場合は「H111」となります。	
wClass	USB デバイスの USB Interface Class 情報を格納します。 ※USB Interface Class は USB.org で定義されている値が格納されます。	

3.4. デバイス接続処理

SxuftpDeviceConnect

SxuftpDeviceConnect は、指定したデバイスサーバに接続されているデバイスと接続処理を行います。

SxuftpDeviceConnect は、旧バージョンのデバイスサーバ用の関数です。通常は SxuftpDeviceConnectEx を使用してください。

```

BOOL SxuftpDeviceConnect(
    DWORD      dwIpaddr,      /* デバイスサーバの IP アドレス */
    LPBYTE     lpbNodeaddr,   /* Ethernet アドレスを格納する変数へのポインタ */
    LPSTR      lpszPortName,  /* USB 論理ポート名 */
    BOOL       bFlags        /* オプション */
);

```

パラメータ	意味
dwIpaddr	デバイスサーバの IP アドレスを指定します。
lpbNodeaddr	接続要求を行うデバイスサーバの Ethernet アドレスを指定します。
lpszPortName	接続するデバイスの USB 論理ポート名を指定します。
bFlags	常に 0 を指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XB

3.5. デバイス接続処理

SxuftpDeviceConnectEx

SxuftpDeviceConnectEx は、指定したデバイスサーバに接続されているデバイスと接続処理を行います。

```

BOOL SxuftpDeviceConnectEx(
    DWORD      dwIpaddr,      /* デバイスサーバの IP アドレス */
    LPBYTE     lpbNodeaddr,   /* Ethernet アドレスを格納する変数へのポインタ */
    LPSXUSBDEVICE lpDevice,   /* USB デバイス情報 */
    BOOL       bFlags        /* オプション */
);

```

パラメータ	意味
dwIpaddr	デバイスサーバの IP アドレスを指定します。
lpbNodeaddr	接続要求を行うデバイスサーバの Ethernet アドレスを指定します。
lpDevice	接続する USB デバイスの SXUSBDEVICE 構造体を示すポインタを指定します。
bFlags	常に 0 を指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XL

3.6. デバイス接続処理

SxuptpDeviceConnectEx2

SxuptpDeviceConnectEx2 は、指定したデバイスサーバに接続されているデバイスと接続処理を行います。

SxuptpDeviceConnectEx2 は、USB データ暗号化、USB デバイス接続パスワードに対応したデバイスサーバとの接続を行うことができます。

DWORD SxuptpDeviceConnectEx2(

 DWORD dwIpaddr, /* デバイスサーバの IP アドレス */
 LPBYTE lpbNodeaddr, /* Ethernet アドレスを格納する変数へのポインタ */
 LPSXUSBDEVICE lpDevice, /* USB デバイス情報 */
 DWORD dwFlags, /* オプションフラグ */
 LPVOID lpbOption /* オプション情報 */

);

パラメータ	意味	
dwIpaddr	デバイスサーバの IP アドレスを指定します。	
lpbNodeaddr	接続要求を行うデバイスサーバの Ethernet アドレスを指定します。	
lpDevice	接続する USB デバイスの SXUSBDEVICE 構造体を示すポインタを指定します。	
dwFlags	接続方法に関するフラグを指定します。フラグは複数指定することができます。	
	値	内容
	SVLC_ENCRYPT	USB デバイスとのデータ通信を暗号化します。
	SVLC_AUTH	接続パスワードを指定します。接続パスワードが有効になっている USB デバイスと接続する場合は、このフラグを指定する必要があります。 このフラグを指定した場合は、lpbOption にパスワード文字列(CHAR 型)を格納した NULL 終端文字列のポインタを指定してください。
lpbOption	dwFlags の値に応じた追加情報を指定します。	

戻り値

関数が正常に終了した場合は 0 を返します。失敗した場合は 1 を返します。認証に失敗した場合は、2 を返します。

XM

3.7. デバイス接続処理

SxuptyDeviceConnectV6

SxuptyDeviceConnectV6 は、指定したデバイスサーバに接続されているデバイスと接続処理を行います。この関数は IPv4/IPv6 通信に対応しています。

SxuptyDeviceConnectV6 は、USB データ暗号化、USB デバイス接続パスワードに対応したデバイスサーバとの接続を行うことができます。

DWORD SxuptyDeviceConnectV6(

LPBYTE lpbIpAddr, /* デバイスサーバの IP アドレス */
int iIpAddrlen, /* IP アドレス長 */
LPBYTE lpbNodeaddr, /* Ethernet アドレスを格納する変数へのポインタ */
LPSXUSBDEVICE lpDevice, /* USB デバイス情報 */
DWORD dwFlags, /* オプションフラグ */
LPVOID lpbOption /* オプション情報 */
);

パラメータ	意味	
lpbIpAddr	デバイスサーバの IP アドレスを指定します。 IPv4 アドレス、IPv6 アドレスを指定することができます。	
iIpAddrlen	指定した IP アドレスの長さを指定します。	
lpbNodeaddr	接続要求を行うデバイスサーバの Ethernet アドレスを指定します。	
lpDevice	接続する USB デバイスの SXUSBDEVICE 構造体を示すポインタを指定します。	
dwFlags	接続方法に関するフラグを指定します。フラグは複数指定することができます。	
	値	内容
	SVLC_ENCRYPT	USB デバイスとのデータ通信を暗号化します。
	SVLC_AUTH	接続パスワードを指定します。接続パスワードが有効になっている USB デバイスと接続する場合は、このフラグを指定する必要があります。 このフラグを指定した場合は、lpbOption にパスワード文字列(CHAR 型)を格納した NULL 終端文字列のポインタを指定してください。
lpbOption	dwFlags の値に応じた追加情報を指定します。	

戻り値

関数が正常に終了した場合は 0 を返します。失敗した場合は 1 を返します。認証に失敗した場合は、2 を返します。

3.8. デバイス切断処理

SxuptyDeviceDisconnect

SxuptyDeviceDisconnect は、指定したデバイスサーバに接続されているデバイスと切断処理を行います。

SxuptyDeviceDisconnect は、旧バージョンのデバイスサーバ用の関数です。通常は SxuptyDeviceDisconnectEx を使用してください。

```

BOOL SxuptyDeviceDisconnect(
    DWORD      dwIpaddr,      /* デバイスサーバの IP アドレス */
    LPBYTE     lpbNodeaddr,   /* Ethernet アドレスを格納する変数へのポインタ */
    LPSTR      lpszPortName   /* USB 論理ポート名 */
);

```

パラメータ	意味
dwIpaddr	デバイスサーバの IP アドレスを指定します。
lpbNodeaddr	切断要求を行うデバイスサーバの Ethernet アドレスを指定します。
lpszPortName	切断するデバイスの USB 論理ポート名を指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XB

3.9. デバイス切断処理

SxuptyDeviceDisconnectEx

SxuptyDeviceDisconnectEx は、指定したデバイスサーバに接続されているデバイスと切断処理を行います。

```

BOOL SxuptyDeviceDisconnectEx(
    DWORD      dwIpaddr,      /* デバイスサーバの IP アドレス */
    LPBYTE     lpbNodeaddr,   /* Ethernet アドレスを格納する変数へのポインタ */
    LPSXUSBDEVICE lpDevice    /* USB デバイス情報 */
);

```

パラメータ	意味
dwIpaddr	デバイスサーバの IP アドレスを指定します。
lpbNodeaddr	切断要求を行うデバイスサーバの Ethernet アドレスを指定します。
lpDevice	切断する USB デバイスの SXUSBDEVICE 構造体を示すポインタを指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XM

3.10. デバイス切断処理

SxuptyDeviceDisconnectV6

SxuptyDeviceDisconnectV6 は、指定したデバイスサーバに接続されているデバイスと切断処理を行います。この関数は IPv4/IPv6 通信に対応しています。

BOOL SxuptyDeviceDisconnectV6(

LPBYTE lpbIpAddr, /* デバイスサーバの IP アドレス */
int iIpAddrLen, /* IP アドレス長 */
LPBYTE lpbNodeAddr, /* Ethernet アドレスを格納する変数へのポインタ */
LPSXUSBDEVICE lpDevice /* USB デバイス情報 */
);

パラメータ	意味
lpbIpAddr	デバイスサーバの IP アドレスを指定します。 IPv4 アドレス、IPv6 アドレスを指定することができます。
iIpAddrLen	指定した IP アドレスの長さを指定します。
lpbNodeAddr	切断要求を行うデバイスサーバの Ethernet アドレスを指定します。
lpDevice	切断する USB デバイスの SXUSBDEVICE 構造体を示すポインタを指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

3.11. ステータス取得処理

SxuptyGetDeviceStatus

SxuptyGetDeviceStatus は、指定したデバイスサーバに接続されている USB デバイスとの接続状況を取得します。

DWORD SxuptyGetDeviceStatus(

DWORD dwIpAddr, /* デバイスサーバの IP アドレス */
LPSTR lpszPortName /* デバイスの USB 論理ポート名を指定します */
);

パラメータ	意味
dwIpAddr	デバイスサーバの IP アドレスを指定します。
lpszPortName	デバイスの USB 論理ポート名を指定します。

戻り値

関数が正常に終了した場合は、下記のステータス情報を返します。

値	意味
0x00000001	USB デバイスは接続可能です。
0x00000002	USB デバイスはコンピュータに接続されています。
0x00000003	USB デバイスは他のユーザーが接続中です。
0x00000004	USB デバイスは検出されていません。
0xFFFFFFFF	ネットワーク通信エラーです。

XM

3.12. ステータス取得処理

SxuptyGetDeviceStatusV6

SxuptyGetDeviceStatusV6 は、指定したデバイスサーバに接続されている USB デバイスとの接続状況を取得します。この関数は IPv4/IPv6 通信に対応しています。

DWORD SxuptyGetDeviceStatusV6(

LPBYTE lpbIpAddr, /* デバイスサーバの IP アドレス */

int iIpAddrLen, /* IP アドレス長 */

LPSTR lpszPortName /* デバイスの USB 論理ポート名を指定します */

);

パラメータ	意味
lpbIpAddr	デバイスサーバの IP アドレスを指定します。 IPv4 アドレス、IPv6 アドレスを指定することができます。
iIpAddrLen	指定した IP アドレスの長さを指定します。
lpszPortName	デバイスの USB 論理ポート名を指定します。

戻り値

関数が正常に終了した場合は、下記のステータス情報を返します。

値	意味
0x00000001	USB デバイスは接続可能です。
0x00000002	USB デバイスはコンピュータに接続されています。
0x00000003	USB デバイスは他のユーザーが接続中です。
0x00000004	USB デバイスは検出されていません。
0xFFFFFFFF	ネットワーク通信エラーです。

XA

3.13. デバイスサーバ情報取得処理

SxuptyGetDeviceServerInfo

SxuptyGetDeviceServerInfo は、指定された IP アドレスのデバイスサーバの情報を取得します。

```

BOOL SxuptyGetDeviceServerInfo(
    DWORD    dwIpaddr,          /* デバイスサーバの IP アドレス */
    LPVOID    lpbServers,       /* 構造体の配列のアドレス */
    DWORD     cbBuf             /* 配列のバイト数 */
);

```

パラメータ	意味
dwIpaddr	デバイスサーバの IP アドレスを指定します。
lpbServers	SXPSEVER 構造体の配列を示すポインタです。デバイスサーバの情報を受け取ります。
cbBuf	lpbServers が指す配列のサイズをバイト単位で指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XM

3.14. デバイスサーバ情報取得処理

SxuptyGetDeviceServerInfoV6

SxuptyGetDeviceServerInfoV6 は、指定された IP アドレスのデバイスサーバの情報を取得します。この関数は IPv4/IPv6 通信に対応しています。

```

BOOL SxuptyGetDeviceServerInfoV6(
    LPBYTE     lpbIpAddr,       /* デバイスサーバの IP アドレス */
    int         iIpAddrLen,     /* IP アドレス長 */
    LPVOID     lpbServers,      /* 構造体の配列のアドレス */
    DWORD      cbBuf           /* 配列のバイト数 */
);

```

パラメータ	意味
lpbIpAddr	デバイスサーバの IP アドレスを指定します。 IPv4 アドレス、IPv6 アドレスを指定することができます。
iIpAddrLen	指定した IP アドレスの長さを指定します。
lpbServers	SXPSEVER 構造体の配列を示すポインタです。デバイスサーバの情報を受け取ります。
cbBuf	lpbServers が指す配列のサイズをバイト単位で指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

備考

XF

XM

SXPSEVER 構造体

typedef struct _SXPSEVER {
 BYTE bNodeaddr[6]; /* Ethernet アドレス */
 DWORD dwIp; /* IPv4 アドレス */
 BYTE bIpAddr[16]; /* IPv6 アドレス */
 char szMachineType[16]; /* デバイスサーバ機種名 */
 char szHostName[16]; /* ホスト名 */
} SXPSEVER, *LPSXPSEVER;

XB

3.15. デバイスサーバ検索ルール No.設定処理

SxuptpSetSrchrRuleCookie

SxuptpSetSrchrRuleCookie は、デバイスサーバ検索時に使用するベンダー固有のチェック No. を設定します。ルール No.が一致するデバイスサーバのみ SxuptpEnumDeviceServers で検索されます。

BOOL SxuptpSetSrchrRuleCookie(
 WORD wRuleCookie /* ルール No. */
);

パラメータ	意味
wRuleCookie	ルール No.を指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XC

3.16. デバイスサーバ再起動処理

SxuptpRebootDeviceServer

SxuptpRebootDeviceServer は、デバイスサーバ本体をリモートで再起動することができます。

BOOL SxuptpRebootDeviceServer(
 DWORD dwIpaddr, /* デバイスサーバの IP アドレス */
 LPBYTE lpbNodeaddr /* Ethernet アドレスを格納する変数へのポインタ */
);

パラメータ	意味
dwIpaddr	デバイスサーバの IP アドレスを指定します。
lpbNodeaddr	再起動を行うデバイスサーバの Ethernet アドレスを指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XM

3.17. デバイスサーバ再起動処理

SxuftpRebootDeviceServerV6

SxuftpRebootDeviceServerV6 は、デバイスサーバ本体をリモートで再起動することができます。
この関数は IPv4/IPv6 通信に対応しています。

BOOL SxuftpRebootDeviceServerV6(

```
    LPBYTE    lpbIpAddr,        /* デバイスサーバの IP アドレス */
    int        iIpAddrLen,      /* IP アドレス長 */
    LPBYTE    lpbNodeAddr      /* Ethernet アドレスを格納する変数へのポインタ */
);
```

パラメータ	意味
lpbIpAddr	デバイスサーバの IP アドレスを指定します。 IPv4 アドレス、IPv6 アドレスを指定することができます。
iIpAddrLen	指定した IP アドレスの長さを指定します。
lpbNodeAddr	再起動を行うデバイスサーバの Ethernet アドレスを指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

XF

3.18. システムステータス取得処理

SxuftpGetSystemStatus

SxuftpGetSystemStatus は、指定したデバイスサーバのシステムステータス情報を取得します。

BOOL SxuftpGetSystemStatus(

```
    DWORD      dwIpaddr,        /* デバイスサーバの IP アドレス */
    LPSTR       lpszStatus,      /* システムステータスを格納する変数へのポインタ */
    DWORD      cbBuf            /* 配列のバイト数 */
);
```

パラメータ	意味
dwIpaddr	デバイスサーバの IP アドレスを指定します。
lpszStatus	デバイスサーバから取得したシステムステータスの文字列を格納する配列のポインタを指定します。
cbBuf	lpszStatus が指す配列のサイズをバイト単位で指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。
システムステータスは、デバイスサーバの機種によっては対応していません。その場合は、関数は失敗します。また、システムステータスの内容は、デバイスサーバの機種によって異なります。

XH

4. 切断要求 API

切断要求 API を使用する場合は、SruInitialize 関数をプログラム内で 1 度呼び出してください。
また、切断要求 API の使用が完了した場合は、SruUninitialize 関数を呼び出してください。

SruInitialize 関数を呼び出すと、切断要求を使用する準備が整います。切断要求のイベントは、Windows メッセージでプログラムに通知します。メッセージを受け取ったプログラムは、メッセージの内容に応じた API を呼び出すことで、切断要求の各機能を実行することができます。

切断要求 API は、ワーカースレッドからの呼び出しには対応していません。必ずメインスレッドから呼び出すようにしてください。

4.1. 切断要求 API 初期化处理

SruInitialize

SruInitialize は、切断要求 API を使用するために必要な初期化处理を行います。切断要求 API を使用するアプリケーションは、SruInitialize 関数を呼び出す必要があります。

```

BOOL SruInitialize(
    HINSTANCE      hInst,          /* インスタンス ハンドル */
    HWND           hParent,        /* 切断要求イベント通知ウィンドウハンドル */
    LPCSTR         lpszMyName     /* 通信時に使用する名前 */
);
    
```

パラメータ	意味
hInst	切断要求 API を使用するアプリケーションのインスタンス ハンドルを指定します。
hParent	切断要求に関するイベントが発生した場合に通知する Windows メッセージの通知先ウィンドウのハンドルを指定します。
lpszMyName	切断要求の送信や他者からの切断要求に応答する際に、通信相手に通知する名前を任意の文字列を指定します。 lpszMyName は、最大 64 バイト(NULL 含む)まで指定可能です。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

4.2. 切断要求 API 終了処理

SruUninitialize

SruUninitialize は、切断要求 API の終了処理を行います。アプリケーションは、切断要求 API を使用する必要がなくなった場合は、SruUninitialize 関数を呼び出してください。

```
BOOL SruUninitialize(
    HINSTANCE      hInst          /* インスタンス ハンドル */
);
```

パラメータ	意味
hInst	SruInitialize 関数の呼び出し時に指定したアプリケーションのインスタンス ハンドルを指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。

4.3. 切断要求の受信応答処理

SruSendReply

SruSendReply は、切断要求の送信者に対して、切断要求を受信したことを応答します。

```
BOOL SruSendReply(
    WORD          wAutoDisconnect, /* 自動切断許可フラグ */
    DWORD         dwTimeout        /* 自動切断タイムアウト値 */
);
```

パラメータ	意味
wAutoDisconnect	自動切断を許可するか指定します。自動切断を許可する場合は、1 を指定してください。許可しない場合は 0 を指定してください。
dwTimeout	自動切断までの時間(秒)を指定します。 wAutoDisconnect の値を 1 にした場合、dwTimeout で指定した秒数が経過した後、切断要求の受信したプログラムは、対象 USB デバイスを切断する必要があります。

戻り値

関数が正常に終了した場合 TRUE を返します。それ以外の場合は FALSE を返します。

解説

切断要求受信メッセージ(SRU_REQUEST)を受け取ったアプリケーションは、切断要求の対象 USB デバイスを接続中か確認する必要があります。USB デバイスを接続中の場合は、アプリケーションは直ちに SruSendReply を使用して、応答を返す必要があります。

4.4. 切断要求の許可応答処理

SruAcceptRequest

SruAcceptRequest は、切断要求を受信したアプリケーションが、要求を許可する場合に呼び出します。

BOOL SruAcceptRequest(VOID);

戻り値

関数が正常に終了した場合 TRUE を返します。それ以外の場合は FALSE を返します。

解説

切断要求を許可するアプリケーションは、SruAcceptRequest 関数の呼び出し後、対象 USB デバイスの切断処理を行う必要があります。また、切断処理の完了後、アプリケーションは SruDeviceDisconnected 関数を呼び出し、切断処理が完了したことを通知する必要があります。

4.5. 切断要求の拒否応答処理

SruRefuseRequest

SruRefuseRequest は、切断要求を受信したアプリケーションが、要求を拒否する場合に呼び出します。

BOOL SruRefuseRequest(VOID);

戻り値

関数が正常に終了した場合 TRUE を返します。それ以外の場合は FALSE を返します。

4.6. 切断処理の完了通知処理

SruDisconnectedDevice

SruDisconnectedDevice は、切断要求を許可後に、対象 USB デバイスの切断処理が完了した場合に、切断要求側のアプリケーションに通知するために使用します。

BOOL SruDisconnectedDevice(VOID);

戻り値

関数が正常に終了した場合 TRUE を返します。それ以外の場合は FALSE を返します。

4.7. 切断要求の処理

SruDeviceNotFound

SruDeviceNotFound は、切断要求を許可後に、対象 USB デバイスのステータスが未検出に変わった場合や切断処理に失敗した場合に、切断要求側のアプリケーションに通知するために使用します。

BOOL SruDeviceNotFound(VOID);

戻り値

関数が正常に終了した場合 TRUE を返します。それ以外の場合は FALSE を返します。

4.8. 切断要求の送信処理

SruSendRequest

SruSendRequest は、USB デバイスを接続しているユーザーに対して、USB デバイスの切断要求を送信します。

HANDLE SruSendRequest(

HWND	hParent,	/* 切断要求イベント通知ウィンドウハンドル */
DWORD	dwIp,	/* デバイスサーバの IP アドレス */
LPBYTE	lpbNodeaddr,	/* デバイスサーバの Ethernet アドレス */
LPSXUSBDEVICE	lpDevice,	/* USB デバイス情報 */
DWORD	dwTimeout	/* タイムアウト値 */

);

パラメータ	意味
hParent	切断要求イベントを通知するウィンドウのハンドルを指定します。イベントは Windows メッセージで行われます。
dwIp	切断要求対象の USB デバイスが接続されているデバイスサーバの IP アドレスを指定します。
lpbNodeaddr	切断要求対象の USB デバイスが接続されているデバイスサーバの Ethernet アドレスを指定します。
lpDevice	切断要求対象の USB デバイスの SXUSBDEVICE 構造体を示すポインタを指定します。
dwTimeout	切断要求の通信タイムアウトを指定します。タイムアウト値は秒(s)単位で指定します。0 を指定した場合はタイムアウトなしで動作します。指定した時間内に通信が正常に行われなかった場合は、切断要求は通信エラーで失敗します。 タイムアウト値は、0 ～ 60 の範囲で指定可能です。

戻り値

関数が正常に終了した場合は、切断要求オブジェクトを表すハンドルが返ります。関数が失敗した場合は、NULL が返ります。

XM

4.9. 切断要求の送信処理

SruSendRequestV6

SruSendRequestV6 は、USB デバイスを接続しているユーザーに対して、USB デバイスの切断要求を送信します。この関数は IPv4/IPv6 通信に対応しています。

HANDLE SruSendRequestV6(

HWND	hParent,	/* 切断要求イベント通知ウィンドウハンドル */
LPBYTE	lpbIpAddr,	/* デバイスサーバの IP アドレス */
int	iIpAddrLen,	/* デバイスサーバの IP アドレスの長さ */
LPBYTE	lpbNodeAddr,	/* デバイスサーバの Ethernet アドレス */
LPSXUSBDEVICE	lpDevice,	/* USB デバイス情報 */
DWORD	dwTimeout	/* タイムアウト値 */

);

パラメータ	意味
hParent	切断要求イベントを通知するウィンドウのハンドルを指定します。イベントは Windows メッセージで行われます。
lpbIpAddr	切断要求対象の USB デバイスが接続されているデバイスサーバの IP アドレスを指定します。
iIpAddrLen	デバイスサーバの IP アドレスの長さを指定します。
lpbNodeAddr	切断要求対象の USB デバイスが接続されているデバイスサーバの Ethernet アドレスを指定します。
lpDevice	切断要求対象の USB デバイスの SXUSBDEVICE 構造体を示すポインタを指定します。
dwTimeout	切断要求の通信タイムアウトを指定します。タイムアウト値は秒(s)単位で指定します。0 を指定した場合はタイムアウトなしで動作します。指定した時間内に通信が正常に行われなかった場合は、切断要求は通信エラーで失敗します。 タイムアウト値は、0 ～ 60 の範囲で指定可能です。

戻り値

関数が正常に終了した場合は、切断要求オブジェクトを表すハンドルが返ります。関数が失敗した場合は、NULL が返ります。

4.10. 切断要求のキャンセル処理

SruCancelRequest

SruCancelRequest は、SruSendRequest で送信した切断要求をキャンセルします。

BOOL SruCancelRequest(

HANDLE	hRequest	/* 切断要求オブジェクトのハンドル */
--------	----------	-----------------------

);

パラメータ	意味
hRequest	SruSendRequest 関数によって返された切断要求オブジェクトのハンドルを指定します。

戻り値

関数が正常に終了した場合 TRUE を返します。それ以外の場合は FALSE を返します。

4.11. 切断要求ハンドルのクローズ処理

SruCloseHandle

SruCloseHandle は、SruSendRequest で取得した切断要求オブジェクトのハンドルをクローズします。切断要求の送信を終了する場合は、SruCloseHandle 関数を使用して、切断要求オブジェクトをクローズしてください。

```
BOOL SruCloseHandle(
    HANDLE    hRequest      /* 切断要求オブジェクトのハンドル */
);
```

パラメータ	意味
hRequest	SruSendRequest 関数によって返された切断要求オブジェクトのハンドルを指定します。

戻り値

関数が正常に終了した場合 TRUE を返します。それ以外の場合は FALSE を返します。

XM

4.12. SRU_NOTIFY メッセージ

SRU_NOTIFY は、切断要求イベントが発生したことを通知するメッセージです。

パラメータ	意味
wParam	切断要求についての情報を受け取る SRUINFO 構造体のポインタです。
lParam	使用しません。常に 0 が格納されます。

解説

SRU_NOTIFY メッセージは、切断要求イベントが発生したことをプログラムに通知する Windows メッセージです。メッセージの通知先は、SruInitialize 関数で指定したウィンドウのハンドル、また、SruSendRequest 関数で指定したウィンドウのハンドルとなります。切断要求のメッセージ種類は、SRUINFO 構造体の wMessage メンバに格納されます。プログラムは、wMessage メンバの種類に応じて、切断要求 API を呼び出す必要があります。

SRUINFO 構造体の定義

```
typedef struct _SRUINFO {
    DWORD      dwSize;                /* SRUINFO 構造体のサイズ */
    WORD       wMessage;              /* 切断要求メッセージ */
    BYTE       bIpAddr[16];           /* 通信先 PC の IP アドレス */
    int        iIpAddrLen;            /* 通信先アドレスサイズ */
    char       szMyName[64];          /* 通信元の名前 */
    char       szDestName[64];        /* 通信相手の名前 */
    BYTE       bSrvIpAddr[16];        /* デバイスサーバの IP アドレス */
    int        iSrvIpAddrLen;         /* デバイスサーバアドレスサイズ */
    BYTE       bNodeAddr[6];          /* デバイスサーバの Ethernet アドレス */
    WORD       wVid;                  /* デバイスの VID */
    WORD       wPid;                  /* デバイスの PID */
    char       szDeviceName[64];       /* デバイス名 */
    char       szLocation[16];         /* デバイス接続位置情報 */
    DWORD      dwRemainingTime;        /* 自動切断実行までの残時間(秒) */
} SRUINFO, *LPSRUINFO;
```

パラメータ	意味
dwSize	SRUINFO 構造体のサイズです。
wMessage	切断要求のメッセージ種別を格納します。メッセージ種別については、「4.13 切断要求メッセージ」を参照してください。
bIpAddr	接続先 PC の IP アドレスです。
iIpAddrLen	接続先 PC IP アドレスの長さです。
szMyName	SruInitialize 関数で指定した名称が格納されます。
szDestName	切断要求の通信相手の名前が格納されます。
bSrvIpAddr	デバイスサーバの IP アドレスです。
iSrvIpAddrLen	デバイスサーバ IP アドレスの長さです。
bNodeAddr	デバイスサーバの Ethernet アドレスです。
wVid	USB デバイスの VID を格納します。
wPid	USB デバイスの PID を格納します。
szDeviceName	USB デバイスの名称です。
szLocation	接続されているデバイスの位置情報を示します。マルチファンクションデバイスの場合は、先頭エンドポイントの情報のみ格納します。
dwRemainingTime	自動切断機能を有効にして、切断要求を行った場合に、自動切断までの残時間を秒で格納します。

4.13. 切断要求メッセージ

メッセージ	意味
SRU_REQUEST	プログラムが切断要求を受信した場合に通知されます。プログラムは、直ちに要求された USB デバイスが自プログラムで接続中か確認し、戻り値を返す必要があります。 自プログラムで接続中の USB デバイスの場合は、SruSendReply 関数を呼び出してください。また、SRU_NOTIFY メッセージに対して TRUE を返してください。そうでない場合は、FALSE を返してください。
SRU_CANCEL	切断要求がキャンセルされた場合に通知されます。
SRU_ACCEPT	切断要求が許可された場合に通知されます。
SRU_REFUSE	切断要求が拒否された場合に通知されます。
SRU_DISCONNECT	USB デバイスの切断が完了した場合に通知されます。 切断要求を送信したプログラムは、このメッセージ受信後に対象 USB デバイスを接続することが可能です。
SRU_BUSY	他ユーザーと切断要求の通信中のため、要求を受信できない場合に通知されます。
SRU_REPLY	切断要求の応答があったことを通知します。
SRU_NOTFOUND	切断要求の受信側プログラムが、対象 USB デバイスを検出できない場合や切断に失敗した場合に通知されます。
SRU_REPLYEX	切断要求の応答があったことを通知します。このメッセージは、自動切断有効で、応答があったことを示します。
SRU_DISCONTIME	自動切断までの残り時間を通知します。 切断要求の受信者側プログラムは、dwRemainingTime の値が 0 になった場合、USB デバイスを切断する必要があります。 以下の順序で、API を呼び出してください。 SruAcceptRequest SxuotpDisconnectDeviceEx SruDeviceDisconnected
SRU_CONFIRM	切断要求の対象 USB デバイスが、自プログラムで接続中か確認する必要がある場合に通知されます。 このメッセージを受け取ったプログラムは、直ちに要求された USB デバイスが自プログラムで接続中か確認し、戻り値を返す必要があります。USB デバイスを接続中の場合は、SRU_NOTIFY メッセージに対して TRUE を返してください。そうでない場合は、FALSE を返してください。
SRU_TIMEOUT	SruSendRequest 関数で指定したタイムアウト値を経過しても、切断要求の応答がない場合、通知されるメッセージです。

XJ

5. デバイスドライバ API

5.1. インスタンス ID 取得処理

SxuptpGetInstanceId

SxuptpGetInstanceId は、SXUPTP Driver 経由で認識している USB デバイスドライバのインスタンス ID を取得します。

対象 USB デバイスが接続中でない場合は、関数は失敗します。

※SxuptpGetInstanceId は、Windows 2000 には対応していません。

DWORD SxuptpGetInstanceId(

 DWORD dwIp, /* デバイスサーバの IP アドレス */

 LPCSTR lpszLocation, /* USB デバイスのロケーション */

 LPSXDEVINST *lpDevInst /* インスタンス ID 情報 */

);

パラメータ	意味
dwIp	インスタンス ID を取得する USB デバイスを接続しているデバイスサーバの IP アドレスを指定します。
lpszLocation	インスタンス ID を取得する USB デバイスのロケーション情報を指定します。
*lpDevInst	SXDEVINST 構造体のポインタへのポインタを指定します。 *lpDevInst に USB デバイスドライバのインスタンス ID が保存されます。この関数を呼び出したアプリケーションは、SxuptpFreeInstanceId を呼び出して、メモリを解放する必要があります。

戻り値

関数が正常に終了した場合は、取得したインスタンス ID の個数が返ります。通常の USB デバイスの場合、インスタンス ID は 1 つになります。複合デバイスの場合、デバイスが持つ機能分インスタンス ID が返ります。

備考

SXDEVINST 構造体

```
typedef struct _SXDEVINST {
```

```
    LPSTR          lpszInstanceId; /* インスタンス ID */
```

```
} SXDEVINST, *LPSXDEVINST;
```

5.2. インスタンス ID 取得処理

SxuotpGetInstanceIdV6

SxuotpGetInstanceIdV6 は、SXUPTP Driver 経由で認識している USB デバイスドライバのインスタンス ID を取得します。

対象 USB デバイスが接続中でない場合は、関数は失敗します。

※SxuotpGetInstanceId は、Windows 2000 には対応していません。

DWORD SxuotpGetInstanceId(

LPBYTE lpbIpAddr, /* デバイスサーバの IP アドレス */
int iIpAddrLen, /* IP アドレス長 */
LPCSTR lpszLocation, /* USB デバイスのロケーション */
LPSXDEVINST *lpDevInst /* インスタンス ID 情報 */

);

パラメータ	意味
lpbIpAddr	デバイスサーバの IP アドレスを指定します。 IPv4 アドレス、IPv6 アドレスを指定することができます。
iIpAddrLen	指定した IP アドレスの長さを指定します。
lpszLocation	インスタンス ID を取得する USB デバイスのロケーション情報を指定します。
*lpDevInst	SXDEVINST 構造体のポインタへのポインタを指定します。 *lpDevInst に USB デバイスドライバのインスタンス ID が保存されます。この関数を呼び出したアプリケーションは、SxuotpFreeInstanceId を呼び出して、メモリを解放する必要があります。

戻り値

関数が正常に終了した場合は、取得したインスタンス ID の個数が返ります。通常の USB デバイスの場合、インスタンス ID は 1 つになります。複合デバイスの場合、デバイスが持つ機能分インスタンス ID が返ります。

備考

SXDEVINST 構造体

```
typedef struct _SXDEVINST {
    LPSTR lpszInstanceId; /* インスタンス ID */
} SXDEVINST, *LPSXDEVINST;
```

XJ

5.3. インスタンス ID 解放処理

SxuptpFreeInstanceId

SxuptpFreeInstanceId は、SxuptpGetInstanceId 関数で取得した SXDEVINST 構造体のメモリを解放します。

※SxuptpGetInstanceId は、Windows 2000 には対応していません。

```
BOOL SxuptpFreeInstanceId(
    LPSXDEVINST lpDevInst    /* インスタンス ID 情報 */
);
```

パラメータ	意味
*lpDevInst	SxuptpGetInstanceId 関数で取得した SXDEVINST 構造体のポインタを指定します。

戻り値

関数が正常に終了した場合は TRUE を返します。それ以外の場合は FALSE を返します。