

2015/02

Matrix USB キー接続/取り外しユーティリティ作成 SDK

© RiBiG Inc. 2013 All Rights Reserved

横浜市港南区上大岡西 1-12-2-801

(有)リビグ

<http://www.ribig.co.jp/matrix>

matrix@ribig.co.jp

目次

この SDK について	3
MxConnect の起動方法	4
アプリケーションとして実行	4
サービスとして実行	4
プラグイン MxConnect が呼び出す プラグイン関数	5
----- Windows サービス/アプリケーション共通 -----	5
----- Windows サービスとして動作している場合のみ -----	5
----- アプリケーションとして動作している場合のみ -----	5
関数詳細	6
void OnMxEvent(BOOL bConnected);	6
int OnServiceStarted(void ** winMsg);	6
void OnServiceStopped();	6
LONG_PTR OnWinMsg(HWND hWnd, UINT message, WPARAM wParam, LPARAM	6
void OnSessionChange(DWORD sessionId, DWORD dwEventType);	8
int OnServiceRegistered();	8
int OnServiceUnregistered();	8
BOOL OnNotifyAdd(HWND hWnd);	9
BOOL OnNotifyDelete(HWND hWnd);	9
サンプルコード C/C++	10
A. 基本サンプル (mxplugin_basic)	10
B. 拡張サンプル(mxplugin)	10
COM プラグインサンプルコード (.Net C#)	11

この SDK について

Matrix USB キーを取り付けると特定の処理がされ、取り外した時にも別の処理がされる、というのはセキュリティキーの典型的な利用方法の 1 つです。しかし、このような利用法を実現するプログラムを開発するのは大変な作業です。この SDK は Matrix キーの接続時、取り外し時に特定の処理を行うプログラムを比較的容易に作成可能にするフレームワークです。

2 つのソフトウェアで構成されます。

1. メインプログラム MxConnect.exe
2. プラグイン Mxplugin.dll / COM サーバ (変更可能)

メインプログラム MxConnect は特定のイベントが発生すると、プラグイン Mxplugin.dll 内の対応する関数、または、プラグイン COM サーバのインターフェース関数を呼び出します。

プラグインは、関数をエクスポートする通常の DLL か、インターフェースを公開する COM DLL です。MxConnect が呼び出す関数、インターフェース関数のひな形が用意されていますので、処理コードを追加するだけで、Matrix キーの接続、取り外し時に希望する処理を行うことができます。

例えば、MxConnect は、Matrix USB キーが取り付けられたり、取り外された時、Mxplugin.dll 内の void WINAPI OnMxEvent(BOOL bConnected) 関数を呼び出します。この関数に処理コードを書くことで MxConnect の動作をカスタマイズできます。

メインプログラム MxConnect.exe は Windows サービス、もしくは、通常のアプリケーションとして実行できるプログラムです。Matrix USB キーが取り付けられたときに行う処理に高い権限が必要だったり、サーバ上でログインしないまま利用したいといったケースが多いため、Windows 起動時に自動実行される Windows サービスの一部としてコードが実行できる点は重要です。高い権限が不要であれば、MxConnect は通常のアプリケーションとして実行できます。

MxConnect, プラグインとも 32bit バージョン/64bit バージョンがあります。Mxplugin.dll は 32bit/64bit どちらにもビルドできます。

64bit の MxConnect は 64bit DLL とのみ動作します。32bit 版 MxConnect は 32bit DLL とのみ動作します。

MxConnect の起動方法

MxConnect.exe を実行するには必ず Mxplugin.dll を同じフォルダ（または、DLL の検索パス）に置いてあるか、COM バージョンのプラグインが登録済みでなければなりません。

アプリケーションとして実行

app という引数をつけて起動します。

```
> mxconnect app
```

サービスとして起動する場合、何も設定をしなれば高い権限の system アカウントで動作しますので、プラグインではあらゆる処理が制限なく可能です。アプリケーションとして実行する場合、起動したユーザの権限で動作します。対象とする処理がそのユーザの権限で行えなければ、プラグインの処理は失敗します。

MxConnect をアプリケーションとして実行すると、Mxconnect が問題なくプラグイン関数を呼び出しながら実行することを確認できます。異常が発生すると MxConnect が異常終了するなどして、すぐに問題があることが分かります。UI がないサービスとして実行する前に、アプリケーションとして実行して問題のないことを確認してみてください。

サービスとして実行

サービスは Windows が開始（起動）します。Windows が開始できるように事前に登録しなければなりません。一度登録して自動開始されるようになっていると、Windows が起動すると自動でサービスは開始（起動）するようになります。MxConnect をサービスとして登録するには、MxConnect.exe を管理者として実行してください。サービスとして登録済みの場合、サービスが解除されます。管理者として実行しなければサービス登録や解除は行われません（一般ユーザとして実行してもメッセージが表示されますが登録、解除処理は正常に行われません）

既定ではサービスは高い権限をもった system アカウントで実行されます（そのため登録、解除には管理者権限が必要になっています）。信頼できるプログラムのみサービス登録するようにしてください。

1. MxConnect を引数を与えずに管理者として実行。まだサービスとして登録されていなければサービスとして登録（Windows が起動したら自動的に起動されるような設定になります）。さらに、登録したサービスを開始します。
2. MxConnect が既にサービスとして登録されていれば、サービスを停止してから登録を解除します。

プラグイン MxConnect が呼び出す プラグイン関数

MxConnect は特定のイベントが発生する毎に以下 9 個のプラグイン関数を呼び出します。その内 3 個は MxConnect が Windows サービスとして動作している時のみに呼び出されます。また、2 つはアプリケーションとして動作している時のみ呼び出されます。どちらの場合でも呼び出される関数は 4 個です。

----- Windows サービス/アプリケーション共通 -----

```
void          OnMxEvent(BOOL);
int           OnServiceStarted(void ** winMsg);
void          OnServiceStopped();
LONG_PTR      OnWinMsg(HWND hWnd, UINT message,
                    WPARAM wParam, LPARAM lParam);
```

----- Windows サービスとして動作している場合のみ -----

```
void  OnSessionChange(DWORD sessionId, DWORD dwEventType);
int   OnServiceRegistered();
int   OnServiceUnregistered();
```

----- アプリケーションとして動作している場合のみ -----

```
BOOL  OnNotifyAdd(HWND hWnd);
BOOL  OnNotifyDelete(HWND hWnd);
```

関数詳細

----- Windows サービス/アプリケーション共通 -----

void OnMxEvent(BOOL bConnected);

Matrix キーが接続されたり、取り外されたときに呼び出されます。接続された時には bConnect が TRUE、取り外されたときには FALSE で呼び出されます。

int OnServiceStarted(void ** winMsg);

MxConnect.exe が起動したときに呼び出されます。Windows サービスではサービスが Windows によって起動されたとき、アプリケーションでは実行ファイルが起動されたときに呼び出されます。

出力専用の引数 winMsg は、プラグインの OnWinMsg で受け取りたい Windows メッセージを指定します。MxConnect が指定されたメッセージを受け取ると、そのメッセージをプラグインの OnWinMsg 転送します。

メッセージが必要なければ *winMsg を NULL にします。必要であれば、UINT 要素を持つ配列のメモリを割り当て、各要素に転送を要求する Windows メッセージを設定します。最後の要素は必ず 0 に設定します。

```
// allocate UINT array
*winMsg = LocalAlloc(LPTR, sizeof(UINT)*16); //new UINT[16];
// set WinMsg routed to OnWinMsg
((UINT*)(*winMsg))[0] = WM_CREATE;
((UINT*)(*winMsg))[1] = MYWM_NOTIFYICON;
((UINT*)(*winMsg))[2] = WM_COMMAND;
// the last element must be 0
((UINT*)(*winMsg))[3] = 0;
```

WM_CREATE 処理で、FALSE を返すと MxConnect は終了します。TRUE で処理が続行します。

void OnServiceStopped();

MxConnect.exe が終了するときに呼び出されます。Windows サービスではサービスが Windows によって停止したとき、アプリケーションではプログラムが終了したときに呼び出されます。

LONG_PTR OnWinMsg(HWND hWnd, UINT message, WPARAM wParam,

LPARAM lParam);

int OnServiceStarted(void ** winMsg) の *winMsg で要求した Windows メッセージがこの関数に送られてきます（ 要求したメッセージ以外は送られてきません ）。WndProc と同じようにコーディングしてください。

----- Windows サービスとして動作している場合のみ -----

void OnSessionChange(DWORD sessionID, DWORD dwEventType);

この関数は Windows サービスがセッション状態の変化を受け取った時に呼び出します。dwEventType に状態が、sessionID には状態が変化したセッションの ID がセットされます。

int OnServiceRegistered();

MxConnect がサービスとして登録されたときに呼び出されます。Windows サービスは最初に登録しなければなりません。登録後、開始（起動）可能になります。一度登録すると、登録解除まで登録された状態のままになります。

int OnServiceUnregistered();

MxConnect がサービスから登録解除されたときに呼び出されます。

----- アプリケーションして動作している場合のみ -----

BOOL OnNotifyAdd(HWND hWnd);

MxConnect がアプリケーションとして実行中、アプリケーションアイコンをタスクトレに表示するときに呼び出されます。hWnd には MxConnect のメインウィンドウのハンドルがセットされます。ここで処理しなければ必ず FALSE を返します。FALSE を返すと MxConnect で既定アイコンが表示されます。処理をする場合、必ず TRUE を返します。

BOOL OnNotifyDelete(HWND hWnd);

MxConnect がアプリケーションとして実行中、アプリケーションアイコンをタスクトレから削除するときに呼び出されます。hWnd には MxConnect のメインウィンドウのハンドルがセットされます。OnNotifyAdd と同じ BOOL 値を返します。

サンプルコード C/C++

VS2008 のソリューション mxplugin に 2 つのプロジェクト mxplugin_basic と mxplugin が収められています。ビルドするとどちらも同じフォルダに同じ DLL 名でプログラムが出来上がるような設定になっています。32bit/64bit どちらにでもビルド可能です。出力フォルダには正しいバージョンの MxConnect.exe が置いてあります。Win7/Win8 に対応します。

A. 基本サンプル (mxplugin_basic)

コードを void OnMxEvent(BOOL bConnected) に追加して

1. キーが接続されると共有フォルダに対するネットワーク経由のアクセスを禁止
2. 取り外すとネットワークアクセスを有効化

という処理を実現します。

フォルダのセキュリティ設定の変更を行うためには高い権限でプログラムが動作していなければなりません。Windows サービスは system 権限で動作するため、このような処理に向いています。Windows サービスをゼロから作成してこの処理を実現するのはかなり面倒ですが、このフレームワークを使えば希望する処理コードのみの追加で Windows サービス実行される処理プログラムを作成できます。

B. 拡張サンプル(mxplugin)

もう 1 つのサンプルでは、WM_CREATE、MYWM_NOTIFYICON、WM_COMMAND をプラグインで受け取ることで、MxConnect のタスクトレイのアイコン、及び、アイコンを右クリックしたときのメニュー表示、メニュー処理を mxplugin.dll でカスタマイズしています。

また、Windows サービスとして実行した場合は、ユーザがログインしたら NotePad を自動起動させます。Vista 以降の OS では Windows サービス自身がユーザインターフェースを持つことはできません。常に隠れて実行します。Windows サービスはユーザのセッションで実行するプログラムで UI を提供しなければなりません（その UI プログラムと Windows サービスは何らかの通信を行いデータのやり取りをします）。ログイン時にプログラムを起動する手法は、サービスの UI プログラムを起動するときに便利です。ユーザセッションにメッセージを表示するコードをロック解除時に追加しています。

COM プラグインサンプルコード (.Net C#)

VS2008 のソリューション mxplugintest に 1 つのプロジェクト mxplugintest が収められています。 .NET によるプラグインは COM サーバになります。 指定 GUID が設定されたインターフェースには C++ のプラグインと同じ関数があります。

```
[InterfaceType(ComInterfaceType.InterfaceIsIUnknown)]
[Guid("3A08E57A-6821-3E5E-A5B2-305616A6F218")]
public interface IMxConnect
{
    void      OnMxEvent(Boolean bConnected);
    unsafe Boolean OnServiceStarted(void** ppWinMsg);
    void      OnServiceStopped();
    Int64      OnWinMsg(IntPtr hWnd, UInt32 message,
                      UInt64 wParam, UInt64 lParam);
    void      OnSessionChange(UInt32 sessionID, UInt32 eventType);
    void      OnServiceRegistered();
    void      OnServiceUnregistered();
    Boolean OnNotifyAdd(IntPtr hWnd);
    Boolean OnNotifyDelete(IntPtr hWnd);
}
```

このインターフェースを実装するクラスを指定 GUID で作成します。

```
[Guid("64939A9B-9995-3B36-B233-9889E5F3AC74")]
public class Plugin : IMxConnect
{
    ...
}
```

MxConnect は DLL をエクスポートする Mxplugin.DLL を見つけると優先的に利用しますが、見つからなければこの指定 GUID クラスの指定 GUID インターフェースを探します。 MxConnect が COM プラグインのインタフェースを利用できるようにするには、COM プラグイン DLL を 以下の通りに RegAsm で登録します。

```
>regasm mxplugintest.dll /tlb:mxplugintest.tlb /codebase
```

備考：

1. regasm.exe へのパスが通っていない場合、regasm.exe を検索してから実行してください。
2. 登録には管理者権限が必要です。
3. 64bit 版 MxConnect から呼び出せるようにするには 64bit 環境で RegAsm を実行して DLL の登録を行ってください。32bit 版 は 32bit 環境で RegAsm 登録を行ってください。
4. 開発者はプロジェクトの“COM 相互運用機能の登録”を有効にすることでビルド時に自動登録されるように設定可能です。ただし、自動登録するには Visual Studio を管理者として実行しなければなりません。また、Visual Studio のバージョンによっては 64bit 環境に正しく自動登録されないのをご注意ください。

関数をエクスポートする C++のプラグインは固定ファイル名 mxplugin.dll でなければなりません。しかし、.NET の COM プラグインではファイル名は重要ではありません。重要なのは GUID（インターフェースと実装クラスのもの）とインターフェースのメモリレイアウトです。サンプルのインターフェース / クラスの GUI、及び、IMxConnect というインターフェースを変更しなければ、他の箇所を変更したとしても MxConnect からインターフェースの関数は呼び出されるはずです。

public interface IMxConnect の各関数の詳細は“MxConnect が呼び出す Mxplugin 関数”章をご参照ください