

## ネットワークドングル機能 - MxNet1

### 概要

Matrix プロテクションシステムには、ファイルベースのネットワーク機能 Mxnet が付属しますが、MxNet1 は TCP/IP ベースのネットワーク機能を提供します。

#### 1. サーバ

Windows/Mac/Linux 対応（それぞれの OS 対応プログラムの提供）

Mac/Linux はコマンドラインプログラムです。

#### 2. クライアント API

Windows/Mac/Linux 対応

API ファイルは、Windows 版は DLL とスタティックライブラリ、Mac 版は dylib、Linux 版は共有ライブラリ(so)

#### 3. サーバ/クライアント間通信

TCP/IP ベースですので、LAN/インターネット経由でクライアントはリモートドングルを利用できます。クライアント・サーバー間通信は OpenSSL を利用して暗号化されています。サーバを実行している OS に関係なく、クライアントはサーバを利用できます。

## 機能

- a. LTP ドングルはサポートしません。USB キーのみ対応します。
- b. クライアントは MxNet1 API 経由で MxNet1 サーバに接続されたりリモートドングル、ローカルドングルのどちらにもアクセスできます。
- c. MxNet1 API は Matrix API とほぼ同じ API を提供します
- d. MxNet1 API は書き込み API、暗号化 API をサポートします。
- e. Matrix API を使ったソースコードは、ほぼそのままリモートドングルを利用できます  
(MxNet1 API は Matrix API に "r" を先頭につけた名前になります。Matrix API 関数のすべてに r を付け加える作業が必要です)

### MxNet1 API 例 :

```
short Init_MatrixAPI() → short rInit_MatrixAPI(void);  
shortl Release_MatrixAPI(void) → shortl rRelease_MatrixAPI(void);  
short Dongle_Count(_mxINT16) → short rDongle_Count(_mxINT16);
```

### コード例 :

```
// MxNet1サーバにログインします。  
result=rInit_MatrixAPI();  
if( result < 0 )  
{  
    printf( "Init_MatrixAPI error %d¥n", result);  
    return -1;  
}  
  
// Get Serial Number  
long ser = rDongle_ReadSerNr(UserCode, dNr,85);
```

```

if ( ser < 0 )
    printf( "SerNo Error %ld\n", ser);
else
    printf( "SerNo: %ld\n", ser);

// Read & Write
result = rDongle_ReadDataEx( UserCode,data, 7, 5, dNr, 85 );
if ( result < 0 )
{
    printf("data read error %d\n", result );
    rRelease_MatrixAPI();
    exit(-1);
}
else
{
    printf( "data read Count=%d\n", result);

    data[0]++;
    data[1]++;
    data[2]++;
    data[3]++;
    data[4]++;
    result = rDongle_WriteDataEx(UserCode,data, 7, 5, dNr, 85);
    if( result < 0 )
        printf("data write error %d\n", result );
    else
        printf( "data write Count=%d\n", result);

}

// Encrypt
data[0]=12345;
data[1]=56789;
result = rDongle_EncryptData(UserCode,data, dNr,85);
if( result > 0 )
    printf( "12345 56789 ->data encrypted %u %u\n", data[0], data[1]);

```

```
else
    printf("encrypt error %d¥n", result );

// Decrypt to see if the original data are returned
result = rDongle_DecryptData(UserCode,data, dNr, 85);
if( result > 0 )
    printf( "data decrypted -> %u %u¥n", data[0], data[1]);
else
    printf("encrypt error %d¥n", result );

// MxNet1サーバからログアウトします。
rRelease_MatrixAPI();
```

-----  
Matrix API関数の先頭に "r" を付いたものが MxNet1 API です。

## リモート dongle / ローカル dongle 切替

設定ファイル(.INI)でリモート、ローカルどちらの dongle にアクセスするかを指定します。  
(Linux/Mac では設定ファイル名は api.cfg )

設定ファイル

=====

[Options]

#リモート = 1

#ローカル = 0

Remote=1

#IP of the computer running MxNet1

#既定

#ip=127.0.0.1

#Port MxNet1 is using

#既定

#port=12300

#Response Timeout in millisec.( how long to wait for MxNet1 response )

#timeout=3000

-----  
Remote=1 とすると MxNet1 サーバと通信します。Remote=0 とすると Matrix API を  
使ってローカル dongle にアクセスします。

Ip / Port / Timeout は Remote=1 の場合のみ有効

## MxNet との相違

1. リモートdongleへの書き込み API( Dongle\_WriteData, Dongle\_WriteDataEx)、リモートdongleを使った暗号化/復号化 API( Dongle\_EncryptData, Dongle\_DecryptData)をサポートします。
2. LPT dongleをサポートしません
3. MxNet ではクライアントは実行している IP アドレスで認識されます ( 複数のクライアントが同じコンピュータで実行している場合、区別できません)。MxNet1 ではクライアントプログラム毎に認識します。
4. MxNet1 は Windows 以外に Linux/Mac OS に対応します。
5. MxNet では SetConfig\_MatrixNet API でネットワーク機能を有効してから Net 対応 API を利用します。MxNet1 では設定ファイルの Remote に 1 を設定することでリモートdongleを利用できます。 同じコードでローカルドongleにも、リモートdongleにもアクセスできます。
6. MxNet1 は TCP/IP ベースなので、MxNet のサーバーファイル指定は不要です。LAN だけでなくインターネット経由でリモートdongleを利用できます
7. Windows 7 の XP モードで動いているクライアントは、ホスト OS である Windows 7 側で動いている MxNet1 サーバに接続することでローカルドongleを快適に利用できます。
8. MxCrypt 実行ファイル暗号化ユーティリティはリモート API に対応していません。リモート API を利用するプログラムの実行ファイルを暗号化するには別途暗号化ユーティリティが必要です ( 標準添付の MxNet API を利用する場合も同様 )