EWM Cellular Utility-Mobiware

User's Manual

Windows XP/7 Version 1.2

Advantech Co. Ltd.

No. 1, Alley 20, Lane 26,
Rueiguang Road, Neihu District,
Taipei 114, Taiwan, R. O. C.

www.advantech.com

Copyright Notice

Part No.
Version: 1.2

Printed in Taiwan 2013-10-07

Version History

Date	Version	Remark
2011-12-07	1.0	First release
2012-11-06	1.1	Modify the module support list.
2013-10-07	1.2	Modify the description

Table of Contents

Introduction

Cellular Mobiware software package is a Cellular management package contains utility and API to access Advantech Cellular Modules.

Benefits

- **Reduce risk**
  The Utility provide the site survey function to reduce risk of error and ensure what operator and wireless-mode is available on site. Also the download link/ Upload link speed measurement tool is able to give you crucial information of actual data rate throughput.

- **Faster Time to Market**
  The Utility is ready to run without modifications. System developers can use it to control the Cellular modules without knowing the controller specs. API and sample code ready for software developers to implement Cellular management mechanism into their applications.

- **Unify Interface**
  Cellular Mobiware provides most frequently used data by simple and unify API

System Requirements

Hardware

Following hardware are required to run Advantech EWM Cellular Mobiware Utility for Windows XP/7：

1. Processor

   Minimum Requirement： A 200 megahertz (MHz) processor, such as the Intel Pentium/Celeron family, AMD K6/Athlon/Duron family, or compatible processor.

2. RAM

   Minimum Requirement： RAM size is dependent on the running applications and using XPE features.

3. Cellular Module

   Advantech Embedded Cellular Module PCI Express Mini Card:

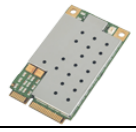
Photo	Model Name	Description
	EWM-C103FD	Europe, WCDMA module, 3G Dual-band UMTS 900, 2100 MHz Quad-band GPRS/ EDGE,
	EWM-C104FT	South + North America, WCDMA module, 3G Tri-band UMTS 850/ 1900/ 2100 MHz, Quad-band GPRS/ EDGE
	EWM-C106FT	7.2 Mbps HSUPA, WCDMA module, Tri-band WCDMA 850/ 1900/ 2100 MHz, Quad-band GSM/ GPRS/ EDGE
	EWM-C107HE	EVDO Cellular module, CDMA 800, 1900 MHz, Half-size Mini PCIe card
	EWM-C107FE	EVDO Cellular module, CDMA 800, 1900 MHz Full-size Mini PCIe card

Photo	Model Name	Description
	EWM-C109F60	3.75G UMTS/HSPA+, 6-band WCDMA, Quad-band GSM/GPRS/EDGE, voice function, w/ SIM Slot,
	EWM-C109F6G	3.75G UMTS/HSPA+, 6-band WCDMA, Quad-band GSM/GPRS/EDGE, GPS function, voice function, w/ SIM Slot

Software

- EWM Cellular Mobiware utility and API are required
- Dot Net Framework 2.0 or above required

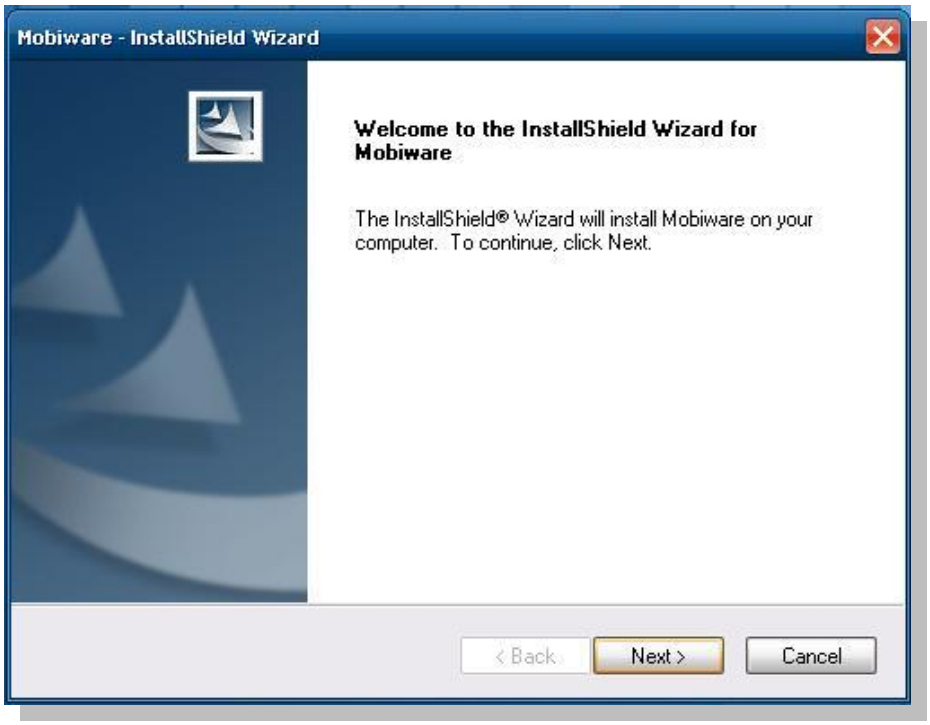
Environments

Operating Systems that EWM Cellular Mobiware Utility supports include:

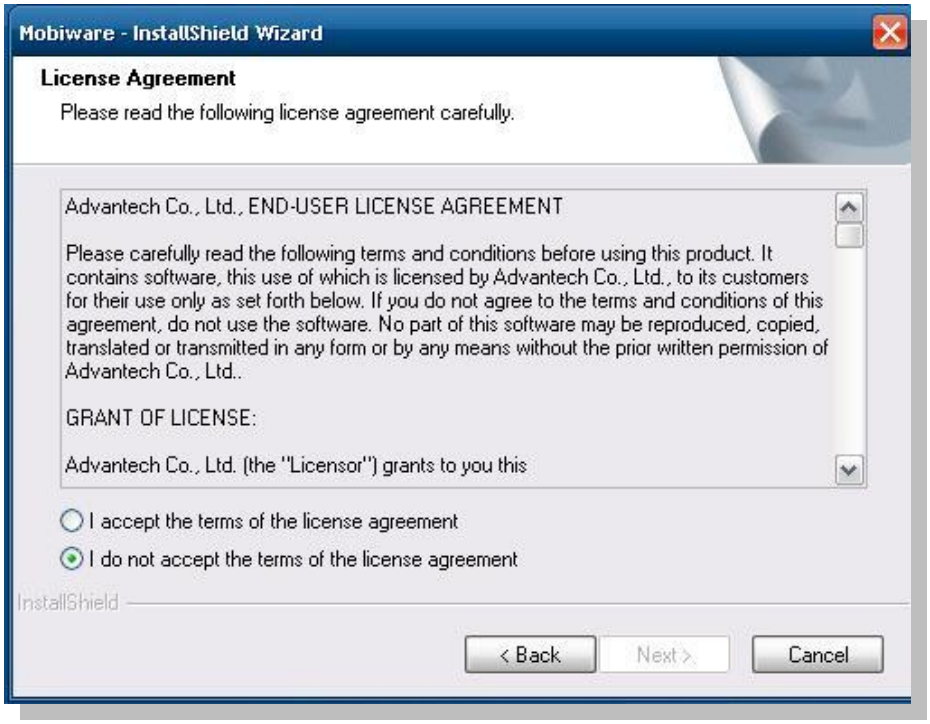
- Windows XP Embedded
- Windows XP Professional or Home Edition
- Windows 7 (32/ 64-bit)

Installation

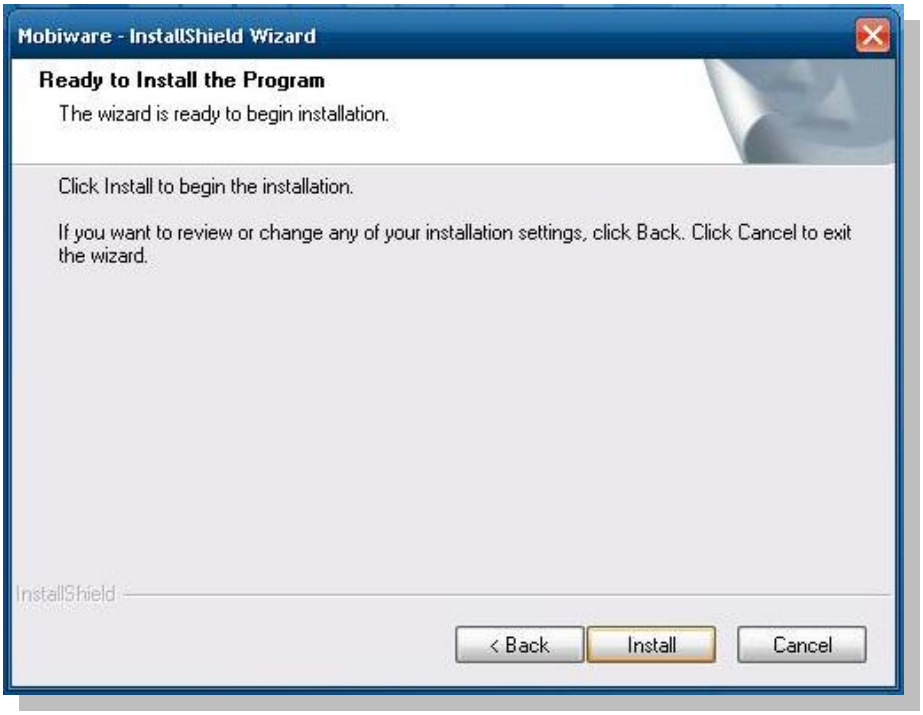
Run the **setup.exe**; and then click “Next”



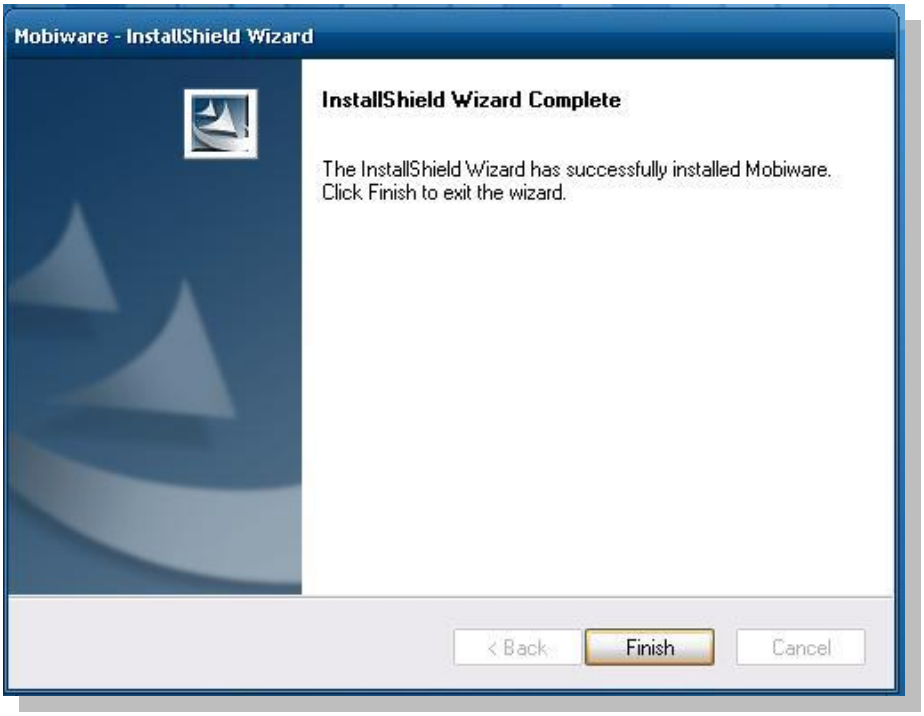
And accept the license and click “Next”



Click “Install”



Click “Finish” and finish the installation.



EWM Cellular Mobiware Utility

EWM Cellular Mobiware Utility is a utility to manage Connection and help to get critical information b using site survey, speed measurement, and so on.

Operating System	Location	Installation
Windows XP		
Windows 7		
Directory	**Contents**	
User Manual	Mobiware_User Manual_V1.3.pdf	
Library Files	ATPoor.dll & Dialup.dll – Dynamic link library	
API Demo program	ESS.exe – Demo program execution file	

Utility Overview

Setting

1. Internet Setting:

Mobiware will give default information as below,

"Profile": Advantech

"APN": Mobiware will get the default information from operator automatically.

Clicking "**New Profile**" & "**New APN**" is able to change Profile and APN, but incorrect APN will cause fail of connection.

And if the connection is needed to always connecting, please click "auto repeat dialup"

Figure-1

2. Lock SIM/ Unlock SIM/ Change PIN code:

Lock SIM card: click “Lock SIM Card (Requires PIN to use module)” and keyin the PIN code to lock SIM card.

Unlock SIM card: click “UnLock SIM Card (Requires PIN to use module)” and keyin the PIN code to unlock SIM card.

Change PIN code: click “Change Pin” and keyin “old PIN code”, and “new PIN code”, also keyin one more time “new PIN code” for confirmation.

Figure-2

Figure-3

Figure-4

3. Wireless Mode:

Wireless Mode setting: click available wireless mode to choose wireless mode.

Figure-5

4. About Module: provide the information of module.

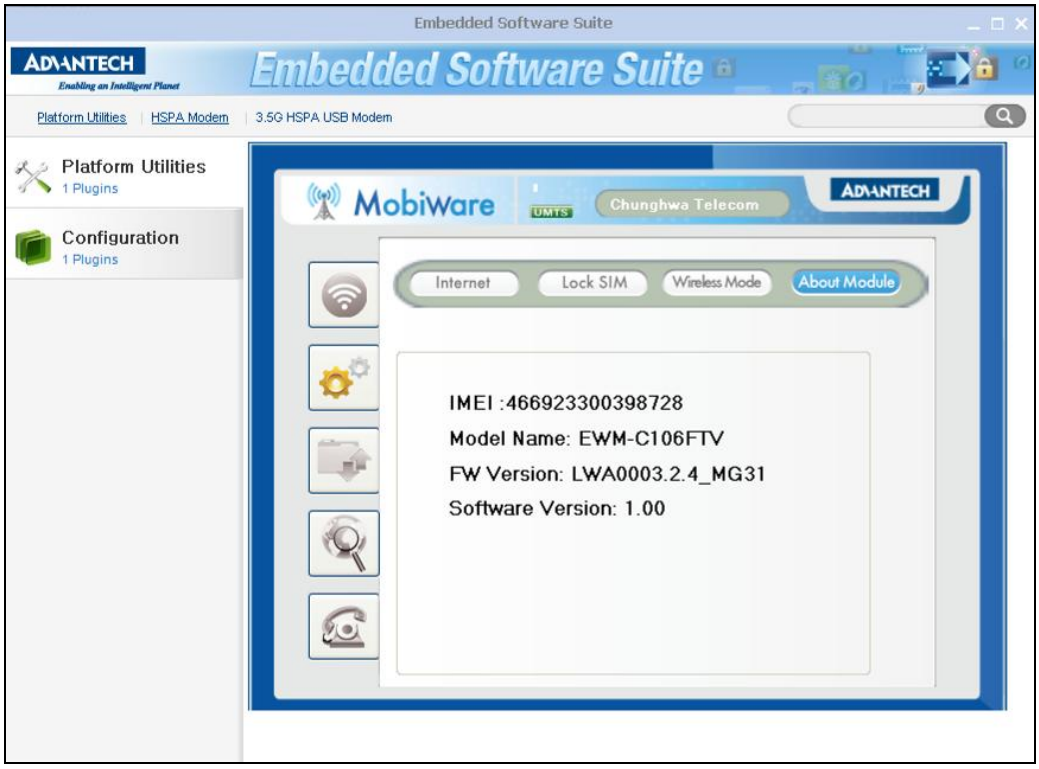


Figure-6

Connect/Disconnect Mobile Network

Click “Connect” button to make connection and click “Disconnect” button to disconnect the connection.

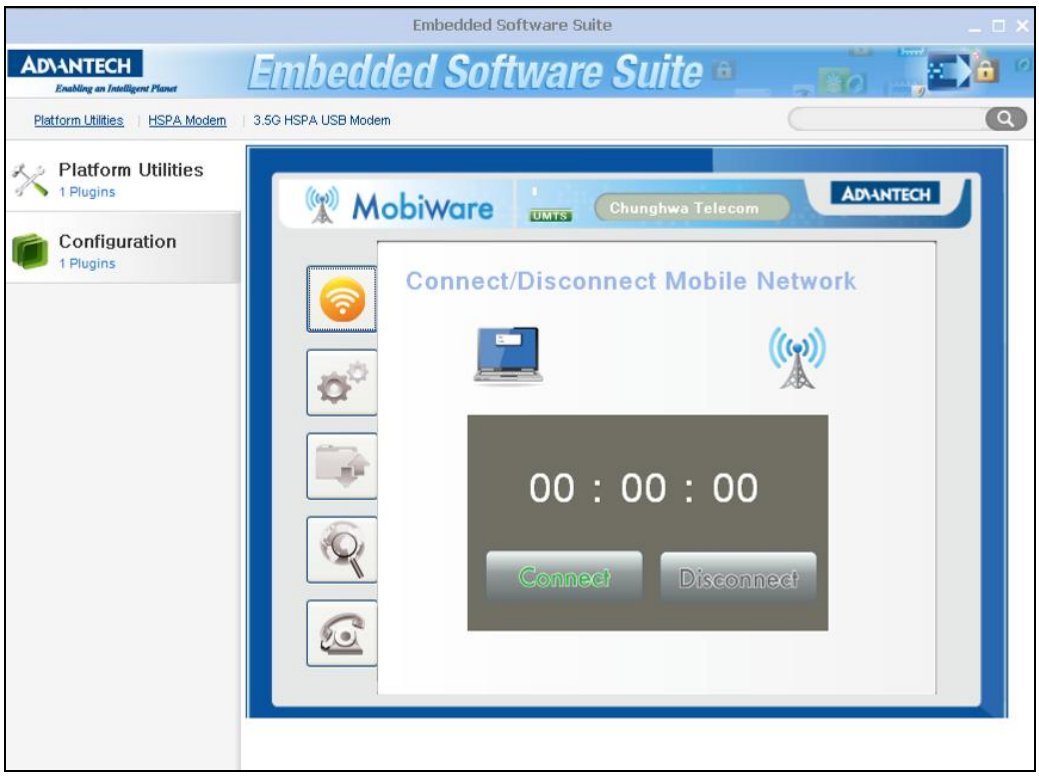


Figure-7

Download/Upload Speed Measurement

1. Set the FTP server information
   - FTP Server IP
   - User ID
   - User Password
   - FTP Root
2. Set the number of test cycles, 1~9
3. Click “Start” button to start testing.

Figure-8

Site survey

Click “Scan Surrounding base station” button to do site survey.

The site survey result is going to show,

1. Status: “Currently Used”, “Available Also”, “Un-Subscribed”.
2. Operator Name
3. Wireless Mode: “WCDMA”, “GSM”…

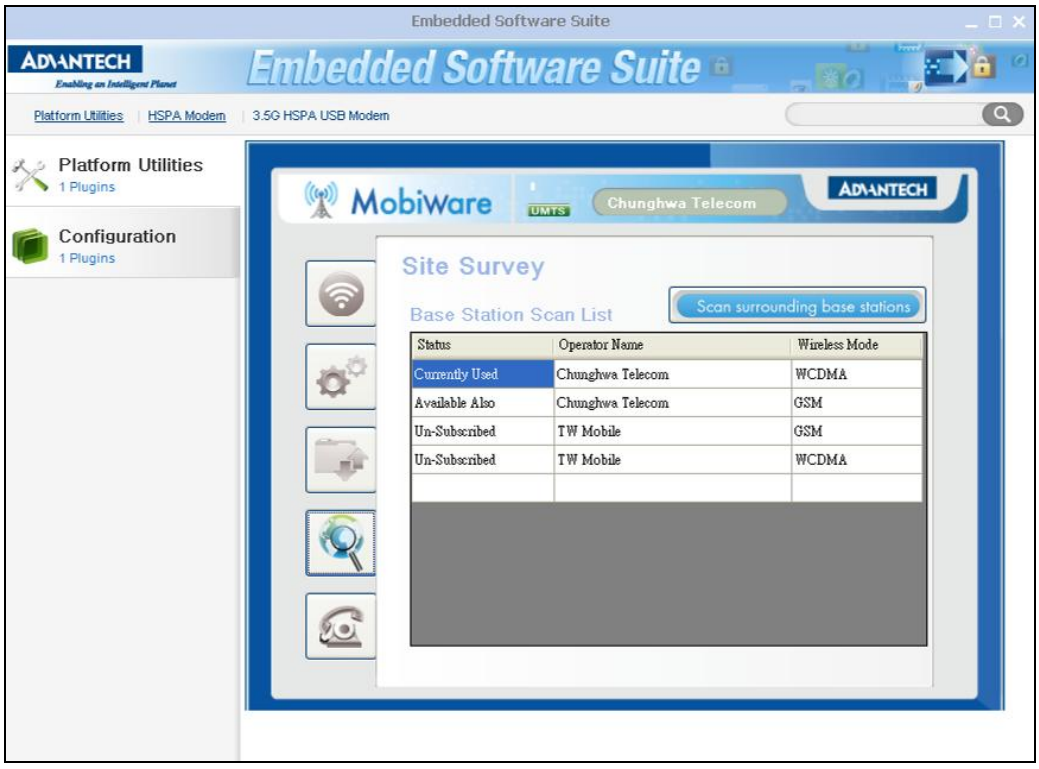


Figure-9

Voice Connection (Only for the cellular module support voice function)

Click the number key for dialing number and click “call” button to call out. And click “End Call” button to hang up the call.

Figure-10

EWM Cellular Mobiware API

Programming Overview

Export functions

Dll functions

ATPoor.dll -

- EWF_HSPA_Init
- EWF_HSPA_SetAPN
- EWF_HSPA_GetAPN
- EWF_HSPA_GetDialUpPhoneNumber
- EWF_HSPA_GetDialUpName
- EWF_HSPA_GetDialUpPassWord
- EWF_HSPA_SetGenerationMode
- EWF_HSPA_QueryGenerationMode
- EWF_HSPA_QueryModule
- EWF_HSPA_QueryFWVersion
- EWF_HSPA_QueryIMSI
- EWF_HSPA_Call
- EWF_HSPA_Call_Status
- EWF_HSPA_Call_HangUp
- EWF_HSPA_SetDisplayCallInNumber
- EWF_HSPA_GetCallInNumber
- EWF_HSPA_GetOperatorName
- EWF_HSPA_QueryCurrentMode
- EWF_HSPA_GetSignalQuality
- EWF_HSPA_QueryField
- EWF_HSPA_GetFieldData
- EWF_HSPA_GetPinStatus
- EWF_HSPA_UnLockPin
- EWF_HSPA_EnablePin
- EWF_HSPA_DisablePin

- EWF_HSPA_QueryPinStatus
- EWF_HSPA_ChangePinCode
- EWF_HSPA_GetPinremaining

Dialup.dll –

- EWF_HSPA_DialUpConnect
- EWF_HSPA_DialUpDisconnect
- EWF_HSPA_CreateGPRSEntry
- EWF_HSPA_DialUpStatus

HSPA API Programmer's Documentation

EWF_HSPA_Init

Initialize the HSPA setting.

```
int EWF_HSPA_Init();
```

Parameters

Return Value

Reference the Error Code.

Remarks

An application must call this function before calling any other HSPA functions.

EWF_HSPA_SetAPN

Set the APN information for operator request.

```
int EWF_HSPA_SetAPN(char *APN_String);
```

Parameters

```
APN_String
  [IN]   Assign the string to set the APN.
```

Return Value

See the Error Code.

Remarks

Access Point Name (APN) is a computer protocol that typically allows a user's computer to access the internet using the mobile phone network. It will decide which network do you used.

EWF_HSPA_GetAPN

Get the APN string from operator information.

```
int EWF_HSPA_GetAPN(char *APN_String,unsigned int size);
```

Parameters

APN_String

[OUT] Retrieve the APN string.

APN_String

[IN] Assign the APN string size, if the size less than APN string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_GetDialUpPhoneNumber

Get the Operator dialup phone number information.

```
int   EWF_HSPA_GetDialUpPhoneNumber(char   *PhoneNumber,
unsigned int Size);
```

Parameters

PhoneNumber

[Out] Retrieve the phone number from operator default.

Size

[IN] Assign the size of phone number string, if the size less than string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_GetDialUpName

Get the Dial up information(User name) from operator information.

```
int EWF_HSPA_GetDialUpName(char *Username, unsigned int size);
```

Parameters

Username

[Out] Retrieve the user name from operator default.

Size

[IN] Assign the size of user name string, if the size less than string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_GetDialUpPassWord

Get the Dial up information (Password) from operator default.

```
int EWF_HSPA_GetDialUpPassWord(char *PassWord, unsigned int size);
```

Parameters

PassWord

[Out] Retrieve the user name from operator default.

Size

[IN] Assign the size of user name string, if the size less than string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_SetGenerationMode

Set the network mode.

```
void EWF_HSPA_SetGenerationMode(int mode);
```

Parameters

```
mode
  [IN]   Set the Generation mode.
  1: GSM only.
  2: WCDMA only.
  Other: GSM/WCDMA Auto select.
```

Return Value

See the Error Code.

Remarks

EWF_HSPA_QueryGenerationMode

Query the network mode.

```
int EWF_HSPA_QueryGenerationMode();
```

Parameters

N/A

Return Value

Mode:

1: GSM only.

2: WCDMA only.

Other: GSM/WCDMA Auto select.

Remarks

EWF_HSPA_QueryModule

Query the module type.

```
int EWF_HSPA_QueryModule();
```

Parameters

Return Value

```
Mode:
  1: Model Name: "EWM-C104"
  2: Model Name: "EWM-C106FTV"
  3: Model Name: "EWM-C107"
```

Remarks

EWF_HSPA_QueryFWVersion

Query the firmware version of module.

```
int EWF_HSPA_QueryFWEversion(char *FWVersion, unsigned int
size);
```

Parameters

FWVersion

[Out] Retrieve the firmware version string.

Size

[IN] Assign the size of Firmware version string, if the size less than string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_QueryIMSI

Query the IMSI code of module.

```
int EWF_HSPA_QueryIMSI(char *IMSI, unsigned int size);
```

Parameters

IMSI

[Out] Retrieve the IMSI string of module.

Size

[IN] Assign the size of IMSI string, if the size less than string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_Call

Assign the phone number for calling telephone.

```
int EWF_HSPA_Call(byte* PhoneNumber, unsigned int size);
```

Parameters

PhoneNumber

[IN] Assign the phone number.

Size

[IN] Assign the size of phone number string, if the size less than string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_Call_Status

Retrieve the calling status.

```
int EWF_HSPA_Call_Status();
```

Parameters

Return Value

1: Ring.

2: Connecting.

3: Hang up.

Remarks

EWF_HSPA_Call_HangUp

Hang up the call.

```
void EWF_HSPA_Call_HangUp();
```

Parameters

N/A

Return Value

N/A

Remarks

EWF_HSPA_SetDisplayCallInNumber

Option for showing the call-in number.

```
int EWF_HSPA_SetDisplayCallInNumber();
```

Parameters

N/A

Return Value

N/A

Remarks

After turn on the option, module will report the call-in number. If turn off the option, library don't report the call-in phone number.

EWF_HSPA_GetCallInNumber

Retrieve the call in phone number.

```
int EWF_HSPA_GetCallInNumber(char *CallNumber, unsigned int size);
```

Parameters

CallNumber

[OUT] Retrieve the call-in number.

Size

[IN] Assign the size of Call in number string, if the size less than string size, it will return Error_SizeError.

Return Value

See the Error Code.

Remarks

EWF_HSPA_GetOperatorName

Query the operator name

```
int EWF_HSPA_GetOperatorName(byte* Name, int size);
```

Parameters

```
Name
  [OUT]  Retrieve the Operator name.
Size
  [IN]   Assign the size of Name string array.
```

Return Value

See the Error Code.

Remarks

EWF_HSPA_QueryCurrentMode

Query the network mode.

```
int EWF_HSPA_QueryCurrentmode();
```

Parameters

N/A

Return Value

Retrieve the network mode.

0: None.

2: GPRS Mode.

3: EDGE Mode.

4: UMTS Mode.

5: HSDPA Mode.

6: HSUPA Mode.

Remarks

EWF_HSPA_GetSignalQuality

Retrieve the signal quality.

```
int EWF_HSPA_GetSignalQuality();
```

Parameters

N/A

Return Value

Retrieve the signal quality. The value will be 0 – 31 & 99(disconnect).

Remarks

EWF_HSPA_QueryField

Query the IMSI code of module.

```
int EWF_HSPA_QueryField();
```

Parameters

N/A

Return Value

Retrieve the base station count number.

Remarks

EWF_HSPA_GetFieldData

Query the base station and retrieve the information.

```
Void EWF_HSPA_GetFieldData(int Count, pFieldData pFiled);
```

Parameters

Count

[IN] Assign the Count number for getting field data.

pFiled

[OUT] Retrieve the filed data. See reference.

Return Value

Remarks

EWF_HSPA_GetPinStatus

Query the pin status.

```
int EWF_HSPA_GetPinStatus();
```

Parameters

N/A

Return Value

0: SIM Ready.

1: PIN1 Lock.

2: PIN2 Lock.

3: PUK1 Lock.

4: PUK2 Lock.

5: SIM busy.

Remarks

If the PIN status is not “Ready”, all functions which connect to base station will fail.

EWF_HSPA_UnLockPin

Unlock the pin code.

```
BOOL EWF_HSPA_UnLockPin(byte* PinCode);
```

Parameters

PinCode

[IN] Assign the pin code for enable lock.

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

EWF_HSPA_EnablePin

Enable the Pin Lock.

```
BOOL EWF_HSPA_EnablePin(byte* PinCode);
```

Parameters

PinCode

[IN] Assign the pin code for enable lock.

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

EWF_HSPA_DisablePin

Disable the pin code checking for reboot.

```
BOOL EWF_HSPA_DisablePin(byte* PinCode);
```

Parameters

PinCode

[IN] Assign the pin code for disable pin checking.

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

EWF_HSPA_QueryPinStatus

Query the pin status.

```
BOOL EWF_HSPA_QueryPinStatus();
```

Parameters

N/A

Return Value

TRUE (1) indicates Lock; FALSE (0) indicates UnLock.

Remarks

EWF_HSPA_ChangePinCode

Change the pin1 code setting.

```
int EWF_HSPA_QueryPinStatus(byte* Old_pin, byte* New_pin,
byte* Confirm_Pin);
```

Parameters

```
Old_pin
  [IN]   Assign the original pin code.
New_pin
  [IN]   Assign the new pin code for changing.
Confirm_pin
  [IN]   Assign the pin code again for corfirm.
```

Return Value

See the Error Code.

Remarks

EWF_HSPA_GetPinRemaining

Query the pin code remaining.

```
BOOL EWF_HSPA_GetPinRemaining(byte* pin1, byte* pin2,
byte* puk1, byte* puk2);
```

Parameters

```
Pin1
  [OUT] Retrieve the times of pin1 remaining.
Pin2
  [OUT] Retrieve the times of pin2 remaining.
Puk1
  [OUT] Retrieve the times of puk1 remaining.
Puk2
  [OUT] Retrieve the times of puk2 remaining.
```

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

EWF_HSPA_DialUpConnect

Dial up for connecting the GPRS network.

```
bool EWF_HSPA_DialUpConnect(string entry_name, string phone_number, string user_name);
```

Parameters

entry_name

[IN] Assign the GPRS entry name for connecting to network.

Phone_number

[IN] Assign the phone number for GPRS entry.

user_name

[IN] Assign the user name for GPRS entry.

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

EWF_HSPA_DialUpDisConnect

Disconnect the GPRS network.

```
bool EWF_HSPA_DialUpDisConnect();
```

Parameters

N/A

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

Before call this function, we need to call the EWF_HSPA_DialUpConnect first. It will create a handle for library, after calling EWF_HSPA_DialupDisConnect, the library will hang up this handle for disconnect.

EWF_HSPA_CreateGPRSEntry

Create a dial up entry.

```
BOOL EWF_HSPA_CreateGPRSEntry(string entry_name, string phone_number);
```

Parameters

```
Entry_name
  [IN]   Assign the GPRS Entry Name.
Phone_number
  [IN]   Assign the phone number for dial up to GRPS network.
```

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

EWF_HSPA_DialUpStatus

Query the Dial up status.

```
BOOL EWF_HSPA_DialUpStatus(DWORD* RX, DWORD* TX, DWORD* TimeDuration);
```

Parameters

```
RX
  [OUT]  Retrieve the receive transfer rate.
TX
  [OUT]  Retrieve the transmit transfer rate.
TimeDuration
  [OUT]  Retrieve all the Dialup duration.
```

Return Value

TRUE (1) indicates success; FALSE (0) indicates failure.

Remarks

About

Advantech's Embedded Core Service (Emb'Core) is an open business model that provides integrated boards, modules and software services to speed up application development. The key proficiencies of Embedded Core Services are software, firmware, module, and board integration capability that offer customized solutions.
For further information please email: EmbCore@advantech.com

About Advantech
Founded in 1983, Advantech is a global leading ePlatform services provider of web-based technology, computing platforms and customization services to empower innovations in the connected eWorld. Advantech cooperates closely with partners to help provide complete solutions for a wide array of applications in various industries. Advantech delivers more than a thousand products and platform solutions in 5 main categories: Industrial & Network Computing, Embedded Computing, Applied Computing, eVideo Solutions, and eAutomation. With more than 2,700 talented people, Advantech operates an extensive support, sales and marketing network in 18 countries and 36 major cities. Advantech delivers efficient time-to-market services to all worldwide customers. (Corporate Website: www.advantech.com).