

Frequently Asked Questions for the Wi-Fi Software Developer's Kit (SDK)

Application Note

v1.1

1 GETTING STARTED WITH THE SDK

1.1 How do I use the SDK with Embedded Visual C++?

To add the SDK to a new eVc project:

1. Use AppWizard to build an application (for example: Projects -> WCE MFC AppWizard(exe).
2. Include the header file in your source file (xxxDlg.cpp or elsewhere):

```
extern "C" {#include "sdc_sdk.h"}
```

3. Link to the libraries. (Project -> Settings)
4. On the 'Link' tab, in the 'Object/Library Modules' box, enter:

```
sdk.lib ws2.lib iphlpapi.lib
```

Make sure that you are using the appropriate sdk.lib file for your platform. Make sure that you have downloaded the latest files from our website

The sdk.lib static library is the Wi-Fi SDK. You can download the latest version on our website. Ws2.lib and iphlpapi.lib are both Microsoft libraries and are available in the standard Microsoft SDKs.

1.2 How do I use the SDK with .NET?

We currently do not provide a static library (LIB file) or dynamic link library (DLL file) for .NET. You can create a wrapper DLL for our SDK to use in your .NET applications.

2 ENCRYPTION, EAP TYPES, CERTIFICATES

2.1 How do I know if my EAP/WEP combinations are valid?

An Extensible Authentication Protocol (EAP) type may not only performs authentication but also generates a dynamic encryption key. When you use an EAP type, you must use an encryption type that uses dynamic keys. When you do not use an EAP type, then you must use an encryption type that uses static keys. Here are possible combinations:

The SDCCConfig is validated on calls to ModifyConfig or AddConfig. If you have an invalid combination of EAP types (SDCCConfig's eapType) and encryption types (SDCCConfig's wepType), and invalid config error will be returned.

Although CCKM-AES is listed in the encryption type enum in the header file, it is not currently supported. Setting an encryption type (wepType) of CCKM-AES will always return an invalid config error.

2.2 How do I set up a single static WEP key?

For a single static WEP key, use the SetWEPKey() SDK call.

A WEP key can be either 40 bit (10 hexadecimal characters) or 128 bit (26 hexadecimal characters) set this using the keyLength parameter.

There can only be one active transmit key at any time. If this key should be the transmit key, set the 5th parameter (txKey) to TRUE.

```
SDCCConfig config;
SDCERR sdcErr;
char configName[80];
DWORD configNumber = 0;
unsigned char theWepKey[13] =
{0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11}

//Get a valid config using GetConfig or another call...
memset(&config, 0, sizeof(SDCCConfig));
sdcErr = GetCurrentConfig(&configNumber, configName);
sdcErr = GetConfig(configName, &config);

//Set WEP type and EAP type
config.wepType = WEP_ON;
config.eapType = EAP_NONE;

//Set the WEP key info
sdcErr = SetWEPKey(&config, 1, WEPLen_128BIT, theWepKey, FALSE);

//Save the config by using ModifyConfig, AddConfig, etc.
sdcErr = ModifyConfig(configName, &config);
```

2.3 How do I set up multiple static WEP Keys?

To set multiple WEP keys, use the SetMultipleWEPKeys () SDK call.

A WEP key can be either 40 bit (10 hexadecimal characters) or 128 bit (26 hexadecimal characters) set this using the keyLength parameter.

Although 4 WEP keys can be stored, there can only be one active transmit key at any time. To set the transmit key, set the 2nd parameter (txKey) to TRUE.

```
SDCCConfig config;
SDCERR sdcErr;
SDCERR sdcErr;
DWORD configNumber = 0;

//Get a valid config using GetConfig or another call...
memset(&config, 0, sizeof(SDCCConfig));
sdcErr = GetCurrentConfig(&configNumber, configName);
sdcErr = GetConfig(configName, &config);

//Set WEP type and EAP type
config.wepType = WEP_ON;
config.eapType = EAP_NONE;

//Set the WEP key info
sdcErr = SetMultipleWEPKeys(&config, 3, WEPLen_40BIT, (unsigned char*) "1111111111",
WEPLen_NOT_SET,
(unsigned char*) "2222222222", WEPLen_40BIT, (unsigned char*) "3333333333",
WEPLen_128BIT,
```

```
(unsigned char*)"123456789012345678901234567");  
  
//Save the config by using ModifyConfig, AddConfig, etc.  
sdcErr = ModifyConfig(configName, &config);
```

2.4 How do I set up Pre-Shared Keys (PSKs)?

A PSK is a null terminated string that should have a length of either 8-63 ASCII characters or 64 hexadecimal characters.

To set a PSK, use the SetPSK() SDK call. In your application, make sure to check for errors after each SDK call.

```
SDCCConfig config;  
SDCERR sdcErr;  
char configName[80];  
DWORD configNumber = 0;  
char hexPSK[ ] = "012345678901234567890123456789012345678901234567890123456789abcd";  
  
//Get a valid config using GetConfig or another call...  
memset(&config, 0, sizeof(SDCCConfig));  
sdcErr = GetCurrentConfig(&configNumber, configName);  
sdcErr = GetConfig(configName, &config);  
  
//Set WEP type and EAP type  
config.wepType = WPA_PSK;  
config.eapType = EAP_NONE;  
  
//Set the PSK  
sdcErr = SetPSK(&config, hexPSK);  
  
//Save the config by using ModifyConfig, AddConfig, etc.  
sdcErr = ModifyConfig(configName, &config);
```

2.5 How do I set up LEAP credentials?

For LEAP, use the SetLEAPCred() SDK call.

```
SDCCConfig config;  
SDCERR sdcErr;  
char user[ ] = "myUserName";  
char pwd[ ] = "myPassWord";  
char configName[80];  
DWORD configNumber = 0;  
  
//Get a valid config using GetConfig or another call...  
memset(&config, 0, sizeof(SDCCConfig));  
sdcErr = GetCurrentConfig(&configNumber, configName);  
sdcErr = GetConfig(configName, &config);  
  
//Set WEP type and EAP type  
config.wepType = WPA_TKIP;  
config.eapType = EAP_LEAP;  
  
//Set the LEAP credentials  
sdcErr=SetLEAPCred(&config, user, pwd);  
  
//Save the config by using ModifyConfig, AddConfig, etc.  
sdcErr = ModifyConfig(configName, &config);
```

2.6 How do I set EAP-FAST credentials?

For EAP-FAST, use the SetEAPFASTCred SDK call.

```
SDCCConfig config;
SDCERR sdcErr;
char user[ ] = "myUserName";
char pwd[ ] = "myPassWord";
char pac1[ ] = "000pac000";
char pac2[ ] = "12345678901234567890123456789012345678901234567890";
char configName[80];
DWORD configNumber = 0;

//Get a valid config using GetConfig or another call...
memset(&config, 0, sizeof(SDCCConfig));
sdcErr = GetCurrentConfig(&configNumber, configName);
sdcErr = GetConfig(configName, &config);

//Set WEP type and EAP type
config.wepType = WPA_TKIP;
config.eapType = EAP_EAPFAST;

//Set the EAP-FAST credentials
sdcErr=SetEAPFASTCred(&config, user, pwd, pac1, pac2);

//Save the config by using ModifyConfig, AddConfig, etc.
sdcErr = ModifyConfig(configName, &config);
```

2.7 How do I set PEAP-MSCHAP credentials?

For PEAP-MSCHAP, use the SetPEAPMSCHAP SDK call.

```
SDCCConfig config;
SDCERR sdcErr;
char user[ ] = "myUsername";
char pwd[ ] = "myPassword";
char cert[ ] = "certfile.cer";
CERTLOCATION certLocation = CERT_FILE;
char configName[80];
DWORD configNumber;

//Get a valid config using GetConfig or another call
memset(&config, 0, sizeof(SDCCConfig));
sdcErr = GetCurrentConfig(&configNum, configName);
sdcErr = GetConfig(configName, &config);

//Set WEP type and EAP type
config.wepType = WPA2_AES;
config.eapType = EAP_PEAPMSCHAP;

//Set the PEAP-GTC credentials
sdcErr=SetPEAPMSCHAPCred(&config, user, pwd, certLocation, cert);

//Save the config by using ModifyConfig, AddConfig, etc.
sdcErr = ModifyConfig(configName, &config);
```

2.8 How do I set PEAP-GTC credentials?

For PEAP-GTC, use the SetPEAPGTC SDK call.

```
SDCCConfig config;
SDCERR sdcErr;
char user[ ] = "userName";
char pwd[ ] = "passWord";
char cert[ ] = "000pac000.cer";
CERTLOCATION certLocation = CERT_FILE;
char configName[80];
DWORD configNumber;

//Get a valid config using GetConfig or another call
memset(&config, 0, sizeof(SDCCConfig));
sdcErr = GetCurrentConfig(&configNum, configName);
sdcErr = GetConfig(configName, &config);

//Set WEP type and EAP type
config.wepType = WPA_TKIP;
config.eapType = EAP_PEAPGTC;

//Set the PEAP-GTC credentials
sdcErr=SetPEAPGTCred(&config, user, pwd, certLocation, cert);

//Save the config by using ModifyConfig, AddConfig, etc.
sdcErr = ModifyConfig(configName, &config);
```

2.9 How do I set EAP-TLS credentials?

For EAP-TLS, use the SetEAPTLS SDK call.

```
SDCCConfig config;
SDCERR sdcErr;
BYTE* userCert = new BYTE[20];
BYTE* caCert = new BYTE[20];
char configName[80];
DWORD configNumber = 0;
CERTLOCATION certLocation = CERT_IN_STORE;

//Get a valid config using GetConfig or another call...
memset(&config, 0, sizeof(SDCCConfig));
sdcErr = GetCurrentConfig(&configNumber, configName);
sdcErr = GetConfig(configName, &config);

GetCerts(TRUE, 2, userCert);
GetCerts(FALSE, 1, caCert);

//Set WEP type and EAP type
config.wepType = WPA_TKIP;
config.eapType = EAP-EAPTLS;

//Set the EAP-TLS credentials
sdcErr=SetEAPTLSCred(&config, "user", (char*)userCert, certLocation, (char*)caCert);

//Save the config by using ModifyConfig, AddConfig, etc.
sdcErr = ModifyConfig(configName, &config);
```

2.10 How do I set EAP-FAST provisioning or automatic PAC provisioning in the SDK?

Changing the provisioning settings in the SDK works in the same way as the SCU. If PAC information is specified in the userName and userPassword variables in the SetEAPFASTCred function, manual PAC provisioning is used. If the PAC information is blank (NULL strings), automatic PAC provisioning is used.

PAC files should be placed in the Summit -> Certs -> directory or in the directory specified in the global setting `certsPath`.

2.11 Where is CA certificate information stored?

All certificate files should be stored in the -> Certs -> subdirectory inside the Summit Client Utility (SCU) directory.

Pass the certificate filename with extension into the appropriate function, like `SetPEAPGTC`.

3 STATUS

3.1 How can I tell when an AP/network/SSID is available?

To tell when you have entered an area where the SSID is available, use our SDK to poll the status. Once the AP/SSID is available, the status will change from 'not associated' to 'associated.' Check the ***cardState*** member of the ***CF10G_STATUS*** structure returned by the ***GetCurrentStatus*** function:

```
SDCERR GetCurrentStatus(CF10G_STATUS *status);
typedef enum _CARDSTATE
{
    CARDSTATE_NOT_INSERTED = 0,
    CARDSTATE_NOT_ASSOCIATED,
    CARDSTATE_ASSOCIATED,
    CARDSTATE_AUTHENTICATED,
    CARDSTATE_FCCTEST,
    CARDSTATE_NOT_SDC
}
CARDSTATE;
```

When the status is ***CARDSTATE_ASSOCIATED*** or ***CARDSTATE_AUTHENTICATED***, the network is available.

3.2 How can I convert the Received Signal Strength Indication (RSSI) to a percentage?

As for a formula to convert RSSI to a percentage, we have not done enough measurements at this time other than -20 = 100% and -95 = 0%.

3.3 How do I calculate Signal Quality and integrate it into my application?

For versions 1.3.0 and above:

Determining signal quality requires three values in ***CF10G STATUS***:

```
DWORD DTIM (range 1-100; no associated unit)
DWORD beaconPeriod (range 20-4000 Kusec or roughly 20-4000 msec)
DWORD beaconsReceived
```

The sdk/driver will fill in these values each time ***UpdateStatus*** is called.

How your application can determine Signal Quality:

- In CAM ***powerSave*** mode, the app should get beacons every ***beaconPeriod***
- In PSP ***powerSave*** mode, the app should get beacons every ***(beaconPeriod * DTIM)***

Notes and Usage Tips:

- In PSP, you can get more beacons than expected if radio is transmitting data, so always round down to 100% signal quality. You will see this especially in the first 20 seconds or so.

- When the driver roams or connects for the first time, **beaconsReceived** will be reset to zero.
- If (**beaconPeriod * DTIM**) > sampling interval, then you should only display signal quality if there is enough data to make a decision.

The SCU updates every 1500 ms. It keeps track of the last four readings and averages them to display Signal Quality.

4 GENERAL

4.1 How do I disable/enable the radio?

You can use the enable/disable functions in the SDK:

```
SDCERR RadioDisable();
SDCERR RadioEnable();
```

4.2 Can I turn off power to the radio with the SDK?

EnableRadio and DisableRadio do not turn off the power to the radio. The radio goes into a 'sleep' mode that minimizes power consumption.

The OS keeps power to the slot. The only way to truly power the radio off is through the OS and controlling the PCMCIA slot.

4.3 5.3 How do I obtain the list of available access points?

From v2.2.3, you can use our QueryOID and SetOID functions with the BSSID_LIST_SCAN and BSSID_LIST OIDs. These return the list of available access points. Please refer to the Microsoft documentation for more information on how to use the links here and here. A code snippet showing how to use our QueryOID and SetOID functions to obtain the list of access points follows.

From v1.3.0, in the SCU we support a 'Scan' feature that displays a list of APs and their RSSI values. Please see Section 3.3 of the User's Guide for more information on this feature.

```
UCHAR QueryBuffer[sizeof(NDISUIO_QUERY_OID)+sizeof(NDIS_802_11_BSSID_LIST_EX)*100];
LONG retval=0;
CString str;

//Tell driver to scan
retval = SetOID(OID_802_11_BSSID_LIST_SCAN, QueryBuffer,
  sizeof(NDISUIO_QUERY_OID)+sizeof(NDIS_802_11_BSSID_LIST_EX)*100);

if ( retval )
{
  //Ask for list of BSSIDS
  retval = QueryOID(OID_802_11_BSSID_LIST, QueryBuffer,
    sizeof(NDISUIO_QUERY_OID)+sizeof(NDIS_802_11_BSSID_LIST_EX)*100);

  if ( retval )
  {
    NDISUIO_QUERY_OID* queryOID = (PNDISUIO_QUERY_OID) &QueryBuffer[0];
    PNDIS_802_11_BSSID_LIST_EX buf = (PNDIS_802_11_BSSID_LIST_EX) &queryOID->Data;

    int nIndex = 0;
    ULONG nNumItems = buf->NumberOfItems;

    //Iterate through list and get ssid and rssi values
```

```
PNDIS_WLAN_BSSID_EX pBssid = &buf->Bssid[0];
for ( ULONG i = 0; i < nNumItems; i++ )
{
str = pBssid->Ssid.Ssid;
writeToTextBox(str);

str.Format(L"%d", pBssid->Rssi);
writeToTextBox(str);

pBssid = (PNDIS_WLAN_BSSID_EX)((LPBYTE)pBssid + pBssid->Length);
}
}
```

4.4 Are ad hoc networks supported by the SDK?

From version 2.0, ad hoc networks are supported in the Wi-Fi SDK. To use ad hoc networks, set the radioMode of the desired profile (SDConfig) to RADIOMODE_ADHOC. With ad hoc networks, you may use a WEP key or unsecured network (no encryption and no EAP types).

```
SDConfig myConfig;

SDCERR err;

memset(&myConfig, 0, sizeof(SDConfig));

//configure ad hoc profile
strcpy(myConfig.configName, "myAdHocConfig");
strcpy(myConfig.SSID, "myAdHocSSID");
myConfig.radioMode = RADIOMODE_ADHOC;

//add profile and activate it
err = AddConfig(&myConfig);
if (err == SDCERR_SUCCESS)
err = ActivateConfig("myAdHocConfig");
```

4.5 Is Automatic Profile Selection supported by the SDK?

As of v1.3.9, we currently do not support roaming between profiles with different SSIDs. An auto-profile selection feature is planned for a future software release.

4.6 What is the maximum number of APs (AP saturation) that can be included in a single scan?

We support up to 64 APs in a scan. AP saturation is the maximum number of simultaneous connections.

4.7 Can I update radio firmware through the SDK?

As of 1.3.9, we do not support this.

4.8 Is an admin password necessary when using the SDK?

You do not need to use or set the admin password when using the SDK. This is for the SCU only.

4.9 How can I determine if the radio is installed and the driver is loaded through the SDK?

The best way is to check the status in GetCurrentStatus. If it returns CARDSTATE_NOT_INSERTED, the driver is not loaded or the card is not currently inserted.

You could also look in the HKLM\Drivers\Active area and search for the SDC-CF10G, but it is probably easier to use GetCurrentStatus.

4.10 How do I retrieve the firmware version number?

The firmware is built into our driver. You should use our driver version instead (CF10G_STATUS -> driver Version).

5 REVISION HISTORY

Version	Date	Notes	Approver
1.0	15 May 2015	Initial Release	Jonathan Kaye
1.1	10 Nov 2016	Updated to new template, removed Global/SROM section.	N. Zach Hogya