



SMS comfort

Datasheet SMS Comfort API

Algemeen

SMS comfort API (*Application Programming Interface*) is een middleware layer met een high level client interface, ondergebracht in de SmsComfortAPI.dll. Indien API geselecteerd, is het niet langer mogelijk de applicatie onder SMTP te gebruiken

SMS Comfort API als oplossing

Met SMS Comfort API als tussenstap bij de verbinding van uw eigen toepassingen met het mobiele netwerk, met alle SMS Comfort functionaliteiten.

Demo applicatie

Er is een demo applicatie geschreven in C# en in Delphi. Daarin wordt er een aparte class gemaakt die de DLL encapsuleert. De source code van deze class is inbegrepen en kan vrij gebruikt worden. Deze class behandelt ook de callbacks en zet ze om in gewone events, bruikbaar voor tal van applicaties.

De demo is een volledig werkende toepassing en is bedoeld als testmateriaal en als programmatie voorbeeld voor de API.

Procedures en callbacks

Alle voids en callbacks gebruiken de **stdcall calling convention**. Dit wil zeggen dat de parameters doorgegeven worden via de stack in volgorde van rechts naar links en dat de opgeroepen void de stack weer vrijmaakt.

Procedures

Alle procedures zijn asynchroon. Een resultaat van een procedure wordt meegedeeld aan de toepassing door een callback. Hierdoor kan de applicatie event driven werken, dus tegelijk nog andere taken uitvoeren.

void Connect(LPStr Addr, LPStr Port);

Maakt de TCP sessie met SmsComfort. **Addr** is ofwel het IP adres van de host ofwel de hostname. **Port** kan de poortnummer zijn of ook een poortnaam.

void Delete(DWord Ref);

Wist het ontvangen SMS-bericht met **Ref** uit de receiving queue.

void Disconnect(void);

Verbreekt de TCP sessie met SmsComfort.

void Finalize(void);

Finalizeert de DLL. Dit is de laatste uit te voeren functie en hierna kan geen andere functie meer uitgevoerd worden.

void Initialize(void);

Initialiseert de DLL. Deze moet als eerste functie uitgevoerd worden voor enige andere functie.

void Send(LPStr Dest, LPStr Text, DWord Ref);

Verzend een **Text** bericht naar **Dest**. De referentie **Ref** wordt terug meegegeven bij de callbacks. Er moet niet gewacht worden op de callback, de API zal de verschillende **Send** calls serialiseren.

void SetDebug(boolean Console);

Opent of sluit een console window met debug informatie. Kan nuttig zijn in het begin van de ontwikkeling.

void SetLoggedOn(void* OnLoggedOn);

void SetNotSentToNet(void* OnNotSentToNet);

void SetReceiveSms(void* OnReceiveSms);

void SetSentToNet(void* OnSentToNet);

void SetSessionClosed(void* OnSessionClosed);

void SetSessionConnected(void* OnSessionConnected);

void SetTransportError(void* OnTransportError)

De meegegeven pointers zijn het adres van de callback routine. Zie volgend hoofdstuk.

LPStr Version(LPStr Dest, DWord MaxLen);

Geeft de versie van de DLL. De return value wijst naar **Dest**. Er moet **MaxLen** plaats zijn + een terminating null.

Callbacks

Events worden uitgevoerd door middel van callbacks. Deze worden uitgevoerd in een andere thread context. De class die de DLL encapsuleert is verantwoordelijk voor de uitvoering ervan in de juiste thread context, indien nodig.

void OnLoggedOn(void);

De communicatie tussen SmsComfort en deze toepassing is in orde.

void OnNotSentToNet(DWord Ref);

Het bericht met referentie **Ref** kon niet verzonden worden naar het Sms netwerk. De **Ref** is eerder meegegeven met de functie **Send()**.

void OnReceiveSms(LPStr From, LPStr Data, DWord Ref);

Dit bericht is ontvangen van het SMS netwerk. Het telefoonnummer is afhankelijk van de gebruikte hardware doch is normaal in internationaal formaat. De referentie **Ref** wordt gebruikt om dit bericht te wissen uit de queue.

void OnSentToNet(DWord Ref);

Het bericht met referentie **Ref** is verzonden naar het Sms netwerk. De referentie **Ref** is eerder meegegeven met de functie **Send()**.

void OnSessionClosed(Word Error);

De TCP sessie met SmsComfort is verbroken. **Error** is de eventuele winsock error.

void OnSessionConnected(Word Error);

De TCP sessie met SmsComfort is tot stand gebracht. **Error** is de eventuele winsock error.

void OnTransportError(DWord Error, LPStr Description)

Dit is meer voor debug doeleinden daar sommige errors dubbel kunnen komen, zoals bijvoorbeeld een TCP connect die mislukt. (Hiervoor een event handler schrijven, zeker handig tijdens ontwikkeling en testen.) De **Error** kan een interne error zijn of een winsock error.

Installatie vereisten.

SMS comfort server moet deel uitmaken van uw netwerk en een fixed IP adres krijgen.
De server moet niet dedicated zijn voor SMS comfort, het kan samen met andere applicaties op een server draaien.

PC vereisten:

MS Vista Business / XP Pro / Win2k Pro of 2000 - 2003 Server OS
256MB RAM of meer
Seriele connectie RS232 (Comport(s) voor GSM modem)
USB port (licentie dongle)
100 MB vrije schijfruimte

GSM vereisten:

AS 5401/V2 SMS/RS232
AS5411/4D COM / LAN
Siemens TC 35 (i)
Siemens MC 35 (i)

Bel uw SMS Comfort leverancier voor mogelijke andere toestellen.

Software vereisten:

API connectie

SMS comfort (Standaard licentie 50 Users + uitbreiding API licentie)
Crypto-Box (USB Dongle)

Test tools

API Client Test Tool

GSM AS5411/D4-LAN

RealPort (virtual comports)
Digi Port Authority - Remote configuration and monitoring tool