

Enterprise Mobile Solutions

openPush - Technical Whitepaper

Datum	Autor	Kommentar
14.03.2012	Sven Braun	



Inhalt

openPush - Technical Whitepaper	1
Was ist openPush?	3
openPush-Anwendungen auf mobilen Geräten.....	3
Beispielanwendung.....	4
Grundfunktionalität des openPush-Clients	5
Benutzeroberfläche.....	5
Anwendungslogik	5
Datenbank	5
Erweiterte Funktionalitäten.....	6
ShockwaveFlash	6
Outlook und Pocket-Outlook API	6
Barcode und RFID	6
GPS Unterstützung.....	7
Gerätesetup	7
COM / DCOM	7
TAPI – Telefonie.....	7
Synchronisation.....	8
Push-Sync	8
HTTP-Sync	8
Plattformen	9
Standardlösungen auf Basis openPush.....	10
Anhang	11
Übersicht Benutzercontrols	11
Formularelemente (Windows).....	11
Properties des Formelement	11
Übersicht XML Properties	12
Referenz db Interface (SQLite3).....	14
Referenz mo Interface (Applikation).....	16
Referenz obj Interface (Benutzercontrols).....	19

Was ist openPush?

openPush ist ein XML-basiertes Framework, mit dem sich ohne tiefergehende Programmierkenntnisse Datenbank-Anwendungen für mobile Geräte entwickeln lassen. Die Anwendungslogik wird mittels XML realisiert. Dadurch kann die sie ohne Anpassungen auf verschiedenen mobilen Geräte-Typen eingesetzt werden. Auf der Basis von openPush entwickelte Anwendungen können zentral verteilt und administriert werden. Sie sind kompatibel zu evtl. vorhandenen Sicherheitsrichtlinien für mobile Geräte.

Die Daten stehen in den Anwendungen auch ohne aktive Online-Verbindungen dem User zur Verfügung. Mittels der integrierten Synchronisations-Funktionalität kann die Anwendung so konfiguriert werden, dass bei Vorhandensein einer Online-Verbindung die Synchronisation mit zentralen Datenbeständen erfolgt.

Dank des geringen Konfigurationsaufwands weist openPush die Vorteile klassischer Standard-Software auf. Zugleich ist durch die hohen Flexibilität des Frameworks jeder gewünschter Individualisierungsgrad möglich. Da alle Anwendungen von mobil only auf openPush basieren, ist die technische Weiterentwicklung sicher gestellt. Stand September 2011 beträgt die Anzahl der bekannten Installationen ca. 7370.

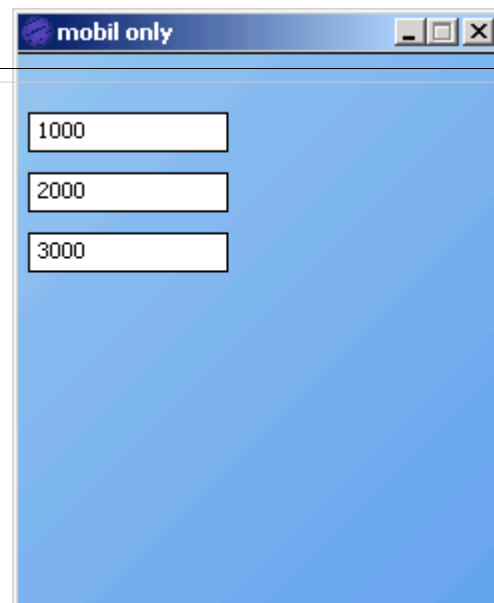
openPush-Anwendungen auf mobilen Geräten

Für die Ausführung der mit openPush entwickelten Anwendung ist eine plattformabhängige, ausführbare Datei (openPush-Client) notwendig. Diese wird durch einfaches Kopieren installiert, weitere Veränderungen am Gerät (z. B. Registry-Veränderungen bei Windows-Mobile-Geräten) sind nicht notwendig. Hierdurch wird es möglich die Software von Speicherkarten und USB Sticks problemlos betreiben zu können. Die Anwendungslogik befindet sich in einer separaten Datei moScript.xml. Diese ist für alle verwendeten Geräte-Typen identisch und befindet sich im gleichen Verzeichnis wie die ausführbare Datei. Die moScript.xml enthält das gesamte UserInterface und die Programmlogik. UserInterface und Programmlogik sind innerhalb dieser Datei getrennt.

Beispielanwendung

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <form id="myform">
    <edit id="field1" x="5" y="30" h="20" w="100" onclick="rechne"/>
    <edit id="field2" x="5" y="60" h="20" w="100" onclick="rechne"/>
    <edit id="field3" x="5" y="90" h="20" w="100"/>
  </form>

  <script lang="JScript">
    <![CDATA[
      function init() {
        mo.LoadForm("myform");
      }
      function uninit() {
      }
      function rechne() {
        field3.text = eval(field1.text + " + " + field2.text);
      }
    ]]>
  </script>
</root>
```



Tipp

Da moScript.xml und die Daten plattformunabhängig sind, können Sie beispielsweise die Anwendung im Smartphone über eine eingesteckte SD Karte betreiben. Wenn Sie die SD Karte nun in einem Notebook mit Cardreader einlegen, können Sie auch auf dem Notebook problemlos weiterarbeiten. Hierfür sind keine Administrationsrechte notwendig.

Die mit der Anwendung erstellten Daten werden im gleichen Verzeichnis abgelegt. Auch dieses Dateiformat ist plattformunabhängig. Somit kann auch bei Wechsel des Geräts (z.B. von Smartphone zu Notebook) mit dem gleichen Datenstand weitergearbeitet werden.

Ein Supportforum befindet sich unter <http://forum.mobilonly.de/viewforum.php?f=4>

Weitere XML Programmbeispiele können unter <http://www.openpush.de/index/examples/> abgerufen werden.

Grundfunktionalität des openPush-Clients

Der openPush-Client ist die Schnittstelle zwischen dem geräteabhängigen Betriebssystem und der einheitlichen Anwendungs-Logik. Er setzt die in der moScript.xml gespeicherte Benutzeroberfläche inkl. der Anwendungslogik gerätespezifisch um.

Benutzeroberfläche

Für die Gestaltung der Benutzeroberfläche stehen 14 Standard-Elemente zur Verfügung. Hierzu gehören das klassische Dialogfeld, Auswahlfelder und Verzeichnisansichten. Für Geräte mit Touchscreen kann das Unterschriftsfeld genutzt werden. Eigene Elemente lassen sich mittels des Frameworks definieren. Die Steuerung der Elemente erfolgt über definierte Eigenschaften. Auch diese können mittels Klassenableitung erweitert werden.

Anwendungslogik

Die Anwendungslogik wird mittels Script realisiert. Da auf allen mobilen Geräten ein ECMA-Scripting-Host (JavaScript) standardmäßig vorhanden ist, empfiehlt sich die Realisierung der Anwendungslogik mit JavaScript. Somit wird eine größtmögliche Plattformunabhängigkeit erreicht.

In Szenarien, in denen ausschließlich Windows-Mobile-Geräte und/oder Windows-Notebooks zum Einsatz kommen, kann die Anwendungslogik alternativ in VB Script realisiert werden.

Datenbank

Die Daten können auf dem mobilen Gerät entweder in XML-Dateien oder mit einer Datenbank-Engine gespeichert werden. Für die größtmögliche Plattformunabhängigkeit ist im openPush-Client die Sqlite3-Engine fest integriert. Die kompakte Datenhaltung und hohe Geschwindigkeit ermöglichen auch auf Smartphones mit langsamer CPU einen schnellen Zugriff auch auf große Datenmengen.

Analog zu den Überlegungen bei der Anwendungslogik können auch alternative Datenbank-Engines eingesetzt werden, z. B.

- IBM DB2 Everyplace
- Oracle Lite
- MS SQL Server
- Sybase IAnywhere Ultralite

Diese Engines müssen unabhängig von der openPush-Anwendung installiert sein und für alle gewünschten Geräte-Typen verfügbar sein..

Erweiterte Funktionalitäten

Über die Anwendungslogik hinaus stellt der openPush-Client zusätzliche Funktionalitäten zur Verfügung. Diese müssen nicht separat auf den Geräten installiert sein. Sie sind integrierter Bestandteil des openPush-Clients.

ShockwaveFlash

In multimediale Anwendungen können Flash-Komponenten eingebunden werden. Die Steuerung ist über Script möglich.

Outlook und Pocket-Outlook API

Für Windows-Mobile-Geräte ist ein XML-Interface mit Zugriff auf die Outlook-Datenbank des jeweiligen Geräts integriert. Es können Kontakte, Kalendereinträge und Aufgaben gelesen und geschrieben werden. Für Outlook (Desktop) steht eine COM-Schnittstelle zur Verfügung.

Barcode und RFID

Die Unterstützung von Barcodescannern und RFID-Readern erfolgt über ein integriertes API. Es können z. B. Geräte mit eingebautem Scanner, externe Bluetooth Scanner und Smartphones mit eingebauter Kamera genutzt werden.

Baracoda Pencil	Bluetooth Buildin Driver. Keine zusätzlichen Treiber erforderlich.	1D Barcodes (EAN, UPC, Code39, Code128, u.a.)
Baracoda D-Fly	Bluetooth Buildin Driver. Keine zusätzlichen Treiber erforderlich.	1D Barcodes (EAN, UPC, Code39, Code128)
Baracoda ID Blue	Bluetooth RFID Stift	RFID Tags lesen und schreiben
Baracoda Roadrunner	Bluetooth Buildin Driver. Keine zusätzlichen Treiber erforderlich.	1D Barcodes (EAN, UPC, Code39, Code128)
Datamatrix / Semacode	Buildin bei Smartphone mit eingebauter Kamera	2D Barcodes
QR Code	Bei Smartphone mit eingebauter Kamera muss zusätzlich Quickmark installiert werden	2D QR-Codes

GPS Unterstützung

Für Telematiklösungen besteht die Möglichkeit, auf verfügbare GPS Empfänger zuzugreifen.

Hierbei werden alle Geräte ab Windows Mobile 5 unterstützt. Bei Smartphones besteht optional auch die Möglichkeit Mobilfunkzellen auszulesen. Dadurch wird eine Ortung auch ohne integriertes GPS Modul möglich.

Gerätesetup

Um den administrativen Aufwand für mobile Endgeräte zu minimieren besteht zusätzlich die Möglichkeit, über Wap Provisionierung komplette Gerätesetups durchzuführen.

Folgende Features stehen zur Verfügung

- Einrichtung von SSL Zertifikaten auf mobilen Geräten
- Setup von Programmverknüpfungen
- automatisierte Installation weiterer Softwarekomponenten
- Definition von Sicherheitsrichtlinien (z. B. Zwangspasswort)
- Autostartfunktion von Speicherkarten aktivieren/deaktivieren
- Setup von DFÜ Verbindungen für Mobilfunkprovider (GPRS,HSCSD,UMTS,EDGE usw.)

COM / DCOM

Es ist möglich, weitere Komponenten z. B. Navigationslösungen mit ActiveX einzubinden und zu steuern. Dadurch werden auch Anbindungen an Officepakete und andere Desktop-Anwendungen möglich.

TAPI – Telefonie

Bei Smartphones ist die Telefonieunterstützung implementiert. Es können Rufnummern durch die Anwendung gewählt werden.

Synchronisation

Die Anwendungen können ihre Daten mit einem Server synchronisieren. Serverseitig wird eine Apache/PHP-Umgebung auf Linux- oder Windows Basis vorausgesetzt. Für kleinere Installationen können klassische RootServer Pakete der Internetprovider genutzt werden. Auf dem Server wird ein webbasiertes Administrationstool installiert. Dies beinhaltet Benutzerverwaltung, Syncmonitor mit Übersicht des Datenverkehrs, Versionsverwaltung der XML Files, Sessionverwaltung und Applikationsdownload.

Die Daten können über SSL beim Datenabgleich verschlüsselt werden. Der Datenaustausch kann auf zwei Wegen erfolgen:

Push-Sync

Das Endgerät versucht bei verfügbarer Netzwerkverbindung über Mobilfunk, WLAN oder Cradle eine TCP-Socketverbindung zu einem zentralen Server herzustellen. Über MessageQueues innerhalb der Serverlogik werden Daten synchronisiert. Der Push-Sync eignet sich für zeitkritische Informationen.

HTTP-Sync

Diese Synchronisation wird entweder zeitgesteuert oder durch eine Benutzereingabe ausgelöst. Hierbei werden verfügbare Netzwerkverbindungen aufgebaut und der Abgleich über das HTTP oder HTTPS Protokoll realisiert. Die Daten werden durch maximale Kompression übertragen. Diese Form der Synchronisation eignet sich besonders für große Datenmengen (z. B. Stammdaten) auf mobilen Geräten ohne permanente Netzwerkanbindung.

Plattformen

Derzeit werden folgende mobile Plattformen unterstützt

Binary	Plattform
moStart_win32.exe	Windows 2000 Windows 2000 Server Windows XP Windows XP Tablet Edition Windows Vista Windows Server 2003
moStart_wm5.exe	Windows Mobile 5 Windows Mobile 6.x
moStart_ppc2003.exe	Pocket PC 2003 Pocket PC 2003 SE Windows Mobile 5 (ohne HTML Controls)
openpush.jar openpush.jad	Nokia Serie 40 Nokia Serie 60 Nokia Serie 80 Motorola mit MIDP2 Samsung mit MIDP2 Sony Ericson mit MIDP2
openpush.cod openpush.alx	Blackberry RIM mit MIDP2
moStart.prc	PalmOS ab Version 4.0

Standardlösungen auf Basis openPush

Lösung	Entwicklung / Vertrieb	Beschreibung
CIP Mobile	Franz Drexel GmbH Mering	Mobile CRM Lösung für die Textilbranche
CamPush	Camdata GmbH Mönchengladbach	Mobile Pushlösung für Email, Adressen, Termine mit webbasierender Groupware
WinTrade	Togrund GmbH	Mobile CRM Lösung mit Standardschnittstelle zum ERP Paket Mesonic
WinLager	Togrund GmbH	Lagerkommisionierung für Warenein- und Ausgang zum ERP Paket Mesonic
moService	Pircher GmbH Austria	Mobiles Servicepaket für Kabelnetzbetreiber und Anbindung an Navision PINET
eCan	Mercoline GmbH	SAP CRM Aussendienstanbindung für Logistik, CRM, Rackjobbing
moCare	mobil only Berlin	Anbindung von mobilen Pflegedienstkräften mit Pflegedokumentation
molInventar	mobil only Berlin	Inventarisierungslösung mit PDA
PocketTrust	mobil only Berlin	Security Lösung mit Softwareverteilung
moTrade	mobil only Berlin	mobile CRM Lösung
moTravel	DATEV eG Nürnberg	elektronisches Fahrtenbuch für Windows Mobile und PalmOS
moLogistik	mobil only Berlin	Logistiklösung für Kurierdienstleister und Speditionen
moSurvey	mobil only Berlin	mobiles Umfragetool für CallCenter mit Handy oder Smartphone

Anhang

Übersicht Benutzercontrols

Formularelemente (Windows)

Element	Description
FORM	Dialog
PAGE	Pagegroup in Dialog
EDIT	Editcontrol
BUTTON	Button ; ImageButton ; Checkbox ; Radiobutton
GRID	GridControl, Checkboxlist
TREE	Tree Control
COMBO	ComboBox
SIGN	Unterschriftsfeld
STATIC	Icon/Text/Button
DATE	Date Picker
TIME	Date/Time Control
PGBAR	Progress Bar Control
SLIDER	ScollBar Control
WEBVIEW	HTML Control

Properties des Formelement

Property	Description
id	unique ID (set as global var in JScript)
onload	function execute after load form
imgbg	background Image
colg1	color gradient start color
colg2	color gradient stop color
h	resize height on X86 Version
w	resize weight on X86 Version

Übersicht XML Properties

'F' müssen definiert sein

'O' können optional angegeben werden

'-' wird nicht unterstützt

property	Beschreibung	EDIT	BUTTON	GRID	PAGE	STATIC	DATE	TIME	FORM
id	unique ID (set as global var in JScript)	F	F	F	F	F	F	F	F
label	Label for Buttons;Defaulttext for Edits	F	F	-	-	-	-	-	-
group	1 to start new Group (Radiobuttons)	-	-	-	-	-	-	-	-
x	X Position im Formular	F	F	F	O	F	F	F	-
y	Y Position im Formular	F	F	F	O	F	F	F	-
h	Height in Pixel	F	F	F	O	F	F	F	O
w	Weight in Pixel	F	F	F	O	F	F	F	O
lx	X Position Landscapemode	O	O	O	-	O	O	O	-
ly	Y Position Landscapemode	O	O	O	-	O	O	O	-
lh	Height Landscapemode	O	O	O	-	O	O	O	-
lw	Weight Landscapemode	O	O	O	-	O	O	O	-
label	Control Label (default id)	-	O	-	-	O	-	-	-
onclick	EventHandle JScript	O	O	O	O	O	-	-	-
onchange	EventHandle JScript	O	-	-	-	-	-	-	-
onfocus	EventHandle JScript	O	-	-	-	-	-	-	-
onblur	EventHandle JScript	O	-	-	-	-	-	-	-
onselect	EventHandle JScript	-	O	-	-	-	-	-	-
onhold	EventHandle JScript Tap&Hold	-	O	-	-	-	-	-	-
onload	EventHandle load a dialog	-	-	-	-	-	-	-	O
imgsrc	Datei BMP 265 Farben Background Image	O	-	-	-	O	-	-	O
bkcol	Color Background #RRGGBB	O	O	O	-	O	O	O	-
txcol	Color Text #RRGGBB	O	O	O	O	O	O	O	-
border	border 0 or 1	O	O	O	-	O	O	O	-
readonly	readonly 0 or 1	O	O	O	-	O	O	O	-
disabled	disabled 0 or 1	O	O	O	O	O	O	O	-
number	number 0 or 1	O	-	-	-	-	-	-	-
password	password 0 or 1	O	-	-	-	-	-	-	-
multiline	multiline 0 or 1	O	-	-	-	-	-	-	-
checkbox	Button als Checkbox	-	O	O	-	-	-	-	-
radio	Button als Radiobutton	-	O	-	-	-	-	-	-
vscroll	Scrollbar vertical	O	-	O	-	-	-	-	-
hscroll	Scrollbar Horizontal	O	-	O	-	-	-	-	-
format	Datumsformat	-	-	-	-	-	F	F	-

font	fontname default TAHOMA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-
fontsize	fontsize default 13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-
fontweight	fontweight default 500	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-
italic	font italic style 0 1 default 0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-
underline	font underline style 0 1 default 0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-

Referenz db Interface (SQLite3)

function	description	parameters	Result	Sample
Open	Open Database File and Create File if not exist	BSTR Filename	VT_I4 dbid or -1 on Error	<code>var dbid = db.Open("\\testdb.db");</code>
Close	Close Database File	VT_I4 dbid	VT_B OOL = false on Error	<code>db.Close(dbid);</code>
Drop	Drop Database File	BSTR Filename	VT_B OOL = false on Error	<code>db.Drop("\\testdb.db");</code>
Exec	Exec SQL Query on DB	VT_I4 dbid, BSTR sql_query	VT_B OOL = false on Error	<code>db.Exec(dbid,"CREATE TABLE test (a1 int, a2 varchar(20))");</code>
Query	SQL Query on DB	VT_I4 dbid, BSTR sql_query	VT_I4 Result set or -1 on Error	<code>var rs = db.Query(dbid,"CREATE TABLE test (a1 int, a2 varchar(20))");</code>
QueryToSync	SQL Query on DB and Add Datarows to Sync	VT_I4 dbid, BSTR sql_query	VT_B OOL = true or false	<code>db.QueryToSync(dbid,"SELECT * FROM motrade_debitor LIMIT 100");</code>
Fetch	Fetch next Row in Resultset from Query	VT_I4 resultset VT_I4 Position 1,2,3... -1 Forward (default) -2 Backward -3 First Row -4 Last Row	VT_B OOL = true or false on EOF	<code>while(db.Fetch(rs)) { }</code>
InsertID	Get Last Insert ID from dbid	VT_I4 dbid	VT_I4	<code>var myid=db.InsertID(dbid);</code>
GetLastError	GetLastError from DB	VT_I4 dbid	BSTR msg	<code>mo.alert(db.GetLastError(dbid));</code>
FreeResult	Free Resultset	VT_I4 recordset	VT_B OOL true	<code>db.FreeResult(rs);</code>
NumRows	Get Number of Rows from Resultset	VT_I4 recordset	VT_I4 count	<code>db.NumRows(rs);</code>
NumCols	Get Number of Columns from Resultset	VT_I4 recordset	VT_I4 count	<code>db.NumCols(rs);</code>

GetColName	Get Columnname from Resultset	VT_I4 recordset, VT_I4 colid beginning with 0	BSTR String	db.GetColName(rs,0);
GetValue	GetValue from Resultset	VT_I4 recordset, BSTR columnname or VT_I4 colnumber beginning with 0	BSTR String	db.GetValue(rs,'a1');

Referenz mo Interface (Applikation)

Terminate	Call JScript Uninit() and close Application	-	-	mo.Terminate();
ShowKeyboard();	enable/disable keyboard	VT_BOOL	-	mo.ShowKeyboard(false);
PopupReset	Clear Popupmenu and Create new	-	-	mo.PopupReset();
PopupAdd	Add Entry to PopupmenuList (id is later the Result off PopupShow and must numeric)	VT_I4 id, BSTR itemtext	VT_BOOL true	mo.PopupAdd(815,"Hello 0815");
PopupShow	Show Popup Menu at Cords x,y AND Return id from PopupAdd	VT_I4 x, VT_I4 y	VT_BOOL true	mo.PopupShow(30,40);
PlaySound	Play Wave Soundfile	BSTR soundfile	VT_BOOL true or false	mo.PlaySound("\\windows\\alar m1.wav");
RegGet	Get Key from Registry with default if not exist	BSTR key, BSTR default	BSTR value	mo.RegGet("Port", "443");
RegSet	Set Registry Value	BSTR key, BSTR value	VT_BOOL true or false	mo.RegSet("Port", "443");
.title	Member	BSTR WindowTitle	R/W	mo.title="Setup";
ShowDoneButton	enable/disable Dialog OK Button	VT_BOOL	-	mo.ShowDoneButton(false);
LoadForm	Load new Form by id Property (its casesensitiv !!!)	BSTR formid	only VT_BOOL true	mo.LoadForm("Deb");
MessageBox	Show HTML Notification Box	BSTR html_content, BSTR title, VT_I4 timeout	only VT_BOOL true	mo.MessageBox("hier den html text", "html_answer", 10);
QuestionYesNo	Show Dialog with YES or NO	BSTR title, BSTR str	only VT_BOOL true if pressed YES	if(mo.QuestionYesNo("Sicherheitsabfrage", "wirklich loeschen ???")) { ... }
GetSystem()	Get System Info	0=Freier Speicher in Bytes 1=Freier Speicher in Prozent 2=OS Major Version 3=OS	BSTR infotext	xyz=GetSystem(2);

		Minor Version 4=OS Build 5=OS CSDVersion 6=Appl Build Timestamp		
ScannerRead()	Read current Barcode, you must define a callback function on_barcode() to handle barcode Events	-	BSTR barcode	function on_barcode() { pos_suchean.text=mo.ScannerRead(); pos_such_ean(); }
ScannerStart()	Start BaracodaManager an connect to Device	BSTR Baracoda BT ID	only VT_BOOL true	mo.ScannerStart(mo.RegGet("btadr", "00:00:00:00:00:00"));
ScannerStop()	Stop the Scanner	-	-	mo.ScannerStop();
DirectorySnapshot()	Take a Filesystem Snapshot of all existing files	VT_BOOL	BSTR Directory	mo.DirectorySnapshot("\\My Documents");
GetChangeFile()	Return new filename since last DirectorySnapshot or empty string if not found	BSTR Filename	BSTR Directory	mo.GetChangeFile("\\My Documents");
vibrate()	Play vibration motor x times (only WM5)	VT_I4 count	-	mo.vibrate(2);
DialUp()	Bring up GPRS Connection (true) or Hangup (false)	-	VT_BOOL	mo.DialUp(true);
SyncInit()	Initilize SyncEngine the Dialog must have Elements (sync_inf1, sync_inf2, sync_inf3, sync_bar1, sync_bar2, sync_bar3)	-	-	mo.SyncInit();
Sync()	Sync Data	-	-	mo.Sync();
SyncSetData()	Set Column Data	-	BSTR key, BSTR val	mo.SyncSetData("col1", "anydata");
SyncAddRow()	Add all current Columns to Row and Empty last Data	-	BSTR key (len==1)	mo.SyncAddRow("W");
SyncAddFile()	Add Files to PostData	-	BSTR filename	mo.SyncAddFile("\\Storage Card\\icon-sync.bmp");
SockOpen()	Open TCP Socket (Receive over function function)	BSTR server, BSTR port	VT_BOOL	mo.SockOpen("www.google.de", "80");

	on_socket() {mo.alert(mo.Get System(8));)			
SockSend()	Send over TCP Socket	VT_BOOL	mo.SockSend(" + tx.text+ ");	BSTR data
SockClose()	Close current TCP Socket	VT_BOOL	mo.SockClose();	-
FileMkdir()	Create Directory	BSTR dirname	VT_BOOL	mo.FileMkdir("\\tester");
FileDelete()	Remove File	BSTR filename	VT_BOOL	mo.FileDelete("\\tester.txt");
FileCopy()	Copy File	BSTR src,BSTR dst	VT_BOOL	mo.FileCopy("\\tester.txt", "\\tester2.txt");
FileMove()	Rename File	BSTR src,BSTR dst	VT_BOOL	mo.FileMove("\\tester.txt", "\\tester2.txt");
FileOpen()	Open File	BSTR name,BSTR R mode ("TRW")	VT_I4 handle	mo.FileOpen("\\tester.txt", "TW"); // Truncate Write
FileClose()	Close File	VT_I4 handle	VT_BOOL	mo.FileClose(handle); //
FileSeek()	Seek PositionFile	VT_I4 handle , VT_I4 position from FILE_BEGIN	VT_BOOL	mo.FileSeek(handle,100);
FileSize()	get FileSize	VT_I4 handle	VT_I4 size	mo.alert(mo.FileSize(handle));
FileWrite()	FileWrite	VT_I4 handle,BSTR data	VT_I4 size	mo.FileWrite(handle,"some Text");
FileRead()	FileRead	VT_I4 handle,VT_I4 len	VT_BSTR data	
FileSelect()	FileSelect Dialog	VT_BSTR title, VT_BSTR file, VT_BSTR filter	VT_BSTR filename	

Referenz obj Interface (Benutzercontrols)

AddProperty()	Set XML Property in XML Tree with key= Value u can set WindowText	BSTR key, BSTR val	-	field1.AddProperty("border","1");
GetProperty()	Get Property from Control (styles,colors,value,etc)	BSTR key	BSTR val	mo.alert(field1.GetProperty("x"));
AddString()	Insert Item to COMBO,GRID data is the value-property on selecting. Insert on Grid is top position	VT_I4 data, BSTR itemte xt1 , BSTR itemte xt2 ... ,BSTR itemte xt10	-	grid1.AddString(-1,0,"Hello 0815"); // Add Headline to Grid
AddColumn()	Set Head of Grid	BSTR itemte xt1 , BSTR itemte xt2 ... ,BSTR itemte xt10	-	grid1.AddColumn("Field1","Field2","Field3"); // Add Headline to Grid
Update()	Redraw the control, need if u change cords,colors etc via AddProperty	-	-	field1.Update();
Reset()	Drop and Init Content in Grids,Combo and Tree	-	-	grid1.Reset();
InsertItem()	Insert Item into TreeView	VT_I4 iData, BSTR iTText, VT_I4 image 1, VT_I4 image 2, VT_I4 parent , BSTR on_select,	VT_BOOL true	tree1.InsertItem(1001,"Test 1001",10,10,0,"","","");

		BSTR on_getdata, BSTR on_hold		
.type	return control element typ (static,combo,grid,...)	readonly	BSTR	mo.alert(myedit.type);
.text	Member of Textcontent/Value	R/W	BSTR / BSTR	mo.alert(myedit.text);
.currentItem	Member of Current Item in GRID/TREE	R/W	BSTR / BSTR	mo.alert(mygrid.currentItem);
.previousItem	Member of Previous Selected Item in GRID/TREE	R/W	BSTR / BSTR	mo.alert(mygrid.previousItem);
.px	X Position from Last Mouseclick	R	VT_I4	
.py	Y Position from Last Mouseclick	R	VT_I4	
SetWidth()	Set Header Width in GRID Controls and Values for Slider	VT_I4 width col1 / VT_I4 width col2VT_I4 width col10	-	liste1.SetWidth(30,60,40,40);
SetFocus()	Set Focus on Control (drop down if combo)	-	-	liste1.SetFocus();
Show()	Set visible status of control	VT_BOOL	-	liste1.Show(true);
ActiveXObject()	Get Interface of ActiveX Objects with UI on Statics	BSTR CLSID	VT_UNKNOWN Dispatch er Object	movie=myflash.ActiveXObject("ShockwaveFlash.ShockwaveFlash");