

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## BLUETOOTH OBEX FTP KLIENT PRO POCKET PC

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ ŠKARECKÝ

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## BLUETOOTH OBEX FTP KLIENT PRO POCKET PC

BLUETOOTH OBEX FTP CLIENT FOR POCKET PC

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ ŠKARECKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV ŠKARVADA

BRNO 2008

## **Abstrakt**

Tato bakalářská práce se zabývá problematikou synchronizace zařízení s operačním systémem Microsoft Windows Mobile za použití technologie Bluetooth. Popisuje princip synchronizace, technologii Bluetooth a vývoj aplikací pro operační systém Windows Mobile. Tyto znalosti jsou poté využity při návrhu a implementaci synchronizační aplikace.

## **Klíčová slova**

OBEX, Pocket PC, Synchronizace, Bluetooth

## **Abstract**

This bachelors thesis deals with problematic of Microsoft Windows Mobile device synchronization via Bluetooth technology. It describes principle of data synchronization, Bluetooth technology and application development process for Windows Mobile. The design and implementation of synchronizing application is also presented.

## **Keywords**

OBEX, Pocket PC, Synchronization, Bluetooth

## **Citace**

Tomáš Škarecký: Bluetooth OBEX FTP klient pro Pocket PC, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Bluetooth OBEX FTP klient pro Pocket PC

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Škarvady a uvedl všechny literární prameny a publikace, ze kterých jsem čerpal

.....  
Tomáš Škarecký  
12. května 2008

## Poděkování

Chtěl bych poděkovat mému vedoucímu panu Ing. Jaroslavovi Škarvadovi za ochotu a čas, který mi věnoval.

© Tomáš Škarecký, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>7</b>
1.1 Členění práce	7
<b>2 Synchronizace</b>	<b>9</b>
2.1 Synchronizace souborů	9
2.2 Rozšiřující možnosti synchronizace souborů	9
2.3 ActiveSync	10
2.3.1 Nedostatky ActiveSync	10
<b>3 Bluetooth</b>	<b>11</b>
3.1 Úvod	11
3.1.1 Historie	11
3.1.2 Využití	11
3.2 Protokoly	11
3.2.1 Bluetooth Radio	12
3.2.2 Baseband	12
3.2.3 HCI	12
3.2.4 L2CAP	12
3.2.5 RFCOMM	12
3.3 Profily	13
3.3.1 Generic Access Profile	13
3.3.2 Serial Port Profile	13
3.3.3 Generic Object Exchange Profile	13
3.4 Bluetooth a WiFi	13
3.5 Komunikace	14
3.5.1 Párování	14
3.5.2 Zabezpečení	15
<b>4 Protokol OBEX</b>	<b>16</b>
4.1 Formát příkazu	16
4.2 Operace protokolu OBEX	16
4.2.1 Operace Spojení (Connect)	16
4.2.2 Operace Odpojení (Disconnect)	17
4.2.3 Operace vložení (Put)	17
4.2.4 Operace stažení (Get)	17
4.2.5 Operace nastavení cesty (SetPath)	17
4.2.6 Operace přerušování (Abort)	18
4.3 Profily využívající protokol OBEX	18

4.3.1	File Transfer Profile . . . . .	19
<b>5</b>	<b>Vývoj aplikací pro Windows Mobile</b>	<b>20</b>
5.1	Softwarové požadavky . . . . .	20
5.2	Vývojové prostředí . . . . .	20
5.2.1	Ladění . . . . .	21
5.3	Zdroje informací a dokumentace . . . . .	21
<b>6</b>	<b>Rozbor problému</b>	<b>22</b>
6.1	Komunikace . . . . .	22
6.2	OBEX FTP a jeho omezení . . . . .	22
6.2.1	Informace o souboru na serveru . . . . .	22
6.2.2	Omezení . . . . .	23
6.3	Synchronizace . . . . .	23
6.3.1	Porovnání souborů . . . . .	23
6.3.2	Konfliktní stav . . . . .	23
6.3.3	Současný přístup více zařízení . . . . .	23
6.3.4	Chybové stavy . . . . .	24
6.4	Konfigurace . . . . .	24
<b>7</b>	<b>Popis implementace</b>	<b>25</b>
7.1	Bluetooth . . . . .	25
7.1.1	Rozhraní pro OBEX . . . . .	25
7.1.2	Vyhledávání zařízení . . . . .	25
7.1.3	Připojení . . . . .	25
7.1.4	Přenos dat . . . . .	26
7.2	Synchronizace . . . . .	26
7.2.1	Informace o synchronizaci adresáře . . . . .	26
7.2.2	Řešení porovnání souboru . . . . .	26
7.2.3	Synchronizace více zařízení . . . . .	27
7.2.4	Ošetření chybových stavů . . . . .	28
7.3	Konfigurace . . . . .	28
7.4	Instalace . . . . .	28
<b>8</b>	<b>Závěr</b>	<b>29</b>
8.1	Možná rozšíření do budoucna . . . . .	29
8.2	Testování . . . . .	29
<b>A</b>	<b>Vývojové diagramy</b>	<b>31</b>
<b>B</b>	<b>Uživatelská příručka</b>	<b>33</b>
B.1	Konfigurační aplikace . . . . .	33
B.1.1	Nastavení vzdálených zařízení . . . . .	33
B.1.2	Nastavení adresářů . . . . .	34
B.1.3	Nastavení obecných vlastností . . . . .	35
B.1.4	Nastavení logu . . . . .	35
B.2	Synchronizační aplikace . . . . .	35
<b>C</b>	<b>Obsah CD</b>	<b>38</b>

# Kapitola 1

## Úvod

Mobilita elektronických zařízení stoupá každým dnem. Dostupnost kapesních počítačů, laptopů a mobilních telefonů, které již pomalu začínají dosahovat funkčnosti, na kterou jsme zvyklí spíše u klasických stolních počítačů, je na velmi vysoké úrovni. Mnoho lidí využívá pro svou práci i zábavu více těchto zařízení. Stav, kdy pracují na více než jednom zařízení, ovšem zcela logicky vyvolává požadavek na aktuálnost dat, která k práci využívají. Z tohoto důvodu je nutné zařízení mezi sebou synchronizovat a to nejlépe bez nutnosti jakékoliv činnosti uživatele.

V zařízeních tohoto typu se stále častěji objevuje technologie Bluetooth, která je určena pro přenos dat na krátkou vzdálenost. Nahrazuje zastaralý a problémový přenos přes infračervený port. Větší rychlost a možnost komunikovat, i pokud mezi zařízeními není přímá viditelnost, této technologii přináší velkou oblibu a rozšířenost.

Možnosti technologie Bluetooth mohou uspokojit některé požadavky, které jsou kladeny na synchronizaci dat. Aplikace, která jich bude využívat, umožní synchronizaci dat zcela automaticky mezi téměř libovolným počtem zařízení vybavených Bluetooth adaptérem.

Tato práce navazuje na znalosti získané v předmětu semestrální projekt. Cílem tohoto předmětu bylo seznámit se s principem vývoje aplikací pro operační systém Windows Mobile a s protokolem OBEX.

### 1.1 Členění práce

V kapitole 2 je popsán základní princip synchronizace. Dále jsou zde zmíněny nejznámější nástroje, které je možné pro synchronizaci použít a také nedostatky, kterými tyto aplikace trpí.

Kapitola 3 se zabývá technologií Bluetooth. Stručně popisuje možnosti této technologie a základní protokoly. Nakonec uvádí rozdíly mezi jednotlivými verzemi.

Kapitola 4 popisuje podrobněji protokol OBEX. Zvláštní podkapitola je věnována profilu FTP, který je využíván pro přenos souborů mezi zařízeními pomocí Bluetooth.

V kapitole 5 je popsán způsob vývoje aplikací pro operační systém Microsoft Windows Mobile. Jsou zde vypsány nutné softwarové prostředky a dostupná vývojová prostředí.

Kapitola 6 rozebírá problematiku synchronizace.

Kapitola 7 popisuje samotnou implementaci programu.

Na závěr dokumentu jsou popsány možná budoucí rozšíření a nedostatky, které stávající implementace obsahuje.



## Kapitola 2

# Synchronizace

Synchronizace je proces, při kterém dochází ke sjednocení dat, nebo stavů na více různých místech.

### 2.1 Synchronizace souborů

Synchronizace souborů je proces, po jehož proběhnutí má dva a více adresářů požadovaný aktuální obsah. Pokud dojde v jednom z nich ke změně souboru, smazání nebo naopak k přidání souboru je tato změna automaticky promítnuta do druhého adresáře při další synchronizaci.

Rozlišujeme dva druhy synchronizace a to **jednosměrnou** a **obousměrnou**. Při jednosměrné synchronizaci jsou data přenášena pouze ze zdrojového místa na místo cílové, k přenosu opačným směrem však nedochází. Tento typ synchronizace je také označován jako zrcadlení dat a může být využíván například u zálohování. Obousměrná synchronizace probíhá, jak již název napovídá, oběma směry. Jejím výsledkem jsou tedy dva identické adresáře.

### 2.2 Rozšiřující možnosti synchronizace souborů

Některé synchronizační nástroje obsahují rozšiřující funkce, jejichž účelem je ušetřit čas, místo nebo zabránit možným ztrátám dat a poskytnout tak uživateli větší komfort než jaký nabízí jednoduchá synchronizace. Mezi tyto funkce patří například:

- Komprese přenášených dat. Při přenosu dat přes síť může díky jejich velikosti docházet ke zpomalení procesu synchronizace. Redukce přenášených dat pomocí komprimace tento jev omezuje.
- Detekce změn v souborech ve zdrojovém i cílovém místě a odpovídající reakce na případný konflikt.
- Možnost přenášet pouze části souborů, které byly změněny. Tato schopnost výrazně snižuje množství dat, které je nutné přenést během synchronizace.
- Schopnost zobrazit chystané změny ještě předtím, než jsou provedeny.
- Schopnost zobrazit rozdíly mezi cílovým a zdrojovým místem.

## 2.3 ActiveSync

ActiveSync je nástroj dodávaný firmou Microsoft, který zajišťuje veškerou komunikaci mezi počítačem a pocket PC. Umožňuje přímý přístup k souborům, sdílení sítě, synchronizaci souborů a synchronizaci s Microsoft Outlook. V současnosti je to zřejmě nejlepší nástroj, který je k těmto činnostem dostupný. Pocket PC je s ním velmi silně provázáno a využívají ho i jiné aplikace.

ActiveSync se skládá ze dvou programů, z nichž jeden musí běžet na počítači, druhý v pocket PC, a komunikují mezi sebou. Propojení je možné pomocí USB kabelu, nebo pomocí technologie Bluetooth.

### 2.3.1 Nedostatky ActiveSync

Nevýhodou synchronizace souborů přes ActiveSync je především omezená možnost nastavení adresářů, u kterých má synchronizace probíhat. Za nevýhodu může být považován i fakt, že je nutné aby byl ActiveSync spuštěn i na počítači, nejen na Pocket PC. Tento nedostatek je však bohatě vykoupěn výhodami, které přináší. Hlavně v možnostech rozhodování při konfliktním stavu dvou souborů.

Nemožnost komunikace se zařízeními, na kterých není spuštěn program ActiveSync, vylučuje z procesu synchronizace všechna zařízení, která neobsahují operační systém Windows. Tedy například počítače s operačním systémem linux. Synchronizace není možná také s přístroji jako jsou mobilní telefony. Jejich kapacita neustále vzrůstá, takže to může být pro mnohé uživatele významné omezení.

## Kapitola 3

# Bluetooth

### 3.1 Úvod

Bluetooth [4] je systém bezdrátového přenosu dat na krátkou vzdálenost. Byl navržen za účelem nahradit kabelové spojení mezi dvěma zařízeními. Mnoho vlastností a schopností je volitelných, takže každý výrobce může zvolit pouze to co potřebuje, aniž by došlo k porušení vzájemné kompatibility.

Tento dokument čerpá ze specifikace Bluetooth verze 1.1 [1]

#### 3.1.1 Historie

Vývoj Bluetooth byl zahájen v roce 1998 firmami Erricson, IBM, Toshiba a Nokia. Specifikace Bluetooth verze 1.0 byly uvolněny 26. června 1999, ale v době uvedení nebyla tato technologie dostatečně levná, aby se dočkala masového rozšíření.

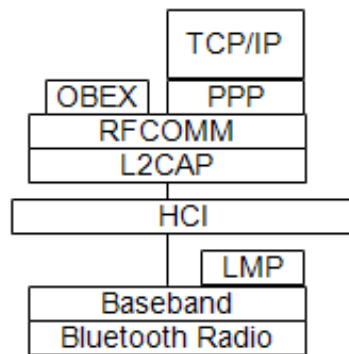
Pojmenování Bluetooth bylo zvoleno podle jména Dánského krále Haralda "Modrozuba" Blaatanda, který žil v 10. století a díky svým dobrým diplomatickým schopnostem dokázal sjednotit mezi sebou válčící kmeny. Technologie Bluetooth měla také sjednotit, avšak tentokrát telekomunikační a výpočetní techniku.

#### 3.1.2 Využití

Bluetooth je dnes použito v mnoha zařízeních. Jeho výhodou je poměrně levný hardware a také jeho nízká náročnost na energii. Hodí se především tam, kde není nutné přenášet velké objemy dat. Časté využití nachází u mobilních telefonů, které ho používají pro výměnu informací mezi sebou, k připojení náhlavní soupravy a přídatných zařízení. Dalším místem využití jsou tiskárny, některé bezdrátové klávesnice a přehrávače.

### 3.2 Protokoly

Popis technologie Bluetooth lze rozdělit do dvou částí, a to na popis protokolů a popis profilů. V této sekci je popis základních protokolů. Protokoly jsou seřazeny do vrstev, kde každá vrstva, kromě první, je postavena nad vrstvou předchozí. Toto uspořádání je zobrazeno na obrázku 3.1. Nejdůležitější protokoly jsou popsány níže. Protokolu OBEX je věnována samostatná kapitola.



Obrázek 3.1: Vrstvy protokolů

### 3.2.1 Bluetooth Radio

Bluetooth Radio je nejnižší vrstva, která definuje požadavky na přenos dat na frekvenčním pásmu 2.4 GHz. Toto pásmo je rozděleno na 79 kanálů po 1 MHz, začíná na frekvenci 2.402 GHz a končí na 2.480 GHz. Reálně je však používáno pouze 23 kanálů.

### 3.2.2 Baseband

Baseband je fyzická vrstva Bluetooth, která leží nad vrstvou Bluetooth Radio. Má za úkol řízení spojení a přenosu dat. Poskytuje služby jako zabezpečení přenosu, korekci chyb. Zpřístupňuje také příchozí pakety vyšším vrstvám a umožňuje vyhledávání jiných zařízení, aby k nim bylo možno přistupovat.

### 3.2.3 HCI

Host Controller Interface (HCI) poskytuje rozhraní pro řízení fyzické vrstvy, pro přístup k stavovým informacím hardwaru a ke kontrolním registrům, vyšším vrstvám v hierarchii protokolů.

### 3.2.4 L2CAP

L2CAP je zkratkou pro Logical Link Control and Adaptation Protocol. Nachází se nad vrstvou Baseband a vyšším protokolům poskytuje následující funkce.

- Přenášení dat mezi různými vyššími protokoly.
- Rozdělování a znovu spojování paketů.
- Jednotnou správu přenosu ke skupině jiných bluetooth přístrojů.
- Quality of service (QoS) správu kvality pro vyšší protokoly.

### 3.2.5 RFCOMM

RFCOMM je jednoduchý protokol, který emuluje klasický RS232 sériový port nad protokolem L2CAP. Protokol je založen na standardu ETSI TS 07.10. RFCOMM. Podporuje

až 60 současně probíhajících spojení mezi dvěma Bluetooth zařízeními. Počet skutečně možných spojení může být však nižší, závisí na konkrétní implementaci Bluetooth adaptéru. Komunikaci přes tento protokol mohou využívat dva druhy zařízení.

- Prvními jsou koncové zařízení, které komunikují mezi sebou. Například počítače, tiskárny apod.
- Druhým typem jsou zařízení, přes které pouze komunikace prochází. Například modemy.

Protokol RFCOMM je navržen tak, že podporuje oba typy zařízení a jeho využití je pro ně totožné.

### 3.3 Profily

Profil definuje, kterou část specifikace je nutné implementovat pro požadovanou funkčnost a jakým způsobem je využívána. V Bluetooth je velké množství profilů, které jsou seřazeny v hierarchické struktuře. Pokud chce zařízení poskytovat funkčnost nějakého vyššího profilu, musí také implementovat všechny profily, které se nacházejí pod ním. U většiny zařízení nejsou všechny profily k dispozici.

#### 3.3.1 Generic Access Profile

Základní Bluetooth profil. Umožňuje vyhledávání a ustanovení spojení mezi zařízeními. Všechny ostatní profily jsou založeny na tomto profilu.

#### 3.3.2 Serial Port Profile

Definuje procedury, které jsou potřebné pro ustanovení a provoz sériového spojení mezi dvěma zařízeními. Používá protokol RFCOMM. Na tomto profilu jsou postaveny například profily:

- Headset Profile - Definuje procedury pro komunikaci mezi telefonem a zařízením headset.
- Dial-up Networking Profile - Definuje procedury pro připojení a používání zařízení jako modem.
- LAN Access Profile - Umožňuje přístup k zařízení pomocí LAN sítě. Využívá protokol Point to Point (PPP), který je postavený nad protokolem RFCOMM.

#### 3.3.3 Generic Object Exchange Profile

Definuje procedury pro využívání protokolu OBEX. Tomuto protokolu se věnuje celá kapitola 4.

### 3.4 Bluetooth a WiFi

WiFi je v současnosti zřejmě nejrozšířenější bezdrátová technologie. Přímé srovnání s Bluetooth je ovšem nepřesné, protože obě technologie jsou určeny pro jiný typ využití. WiFi funguje jako klasický ethernet, zatímco Bluetooth je určeno k nahrazení kabelového spojení

	Wifi	Bluetooth
Dosah	Může být i několik kilometrů	Dle třídy 1/10/100m
Rychlost přenosu	až 54 Mb/s	až 3 Mb/s
Pásmo	2.4 nebo 5 GHz	2.4 GHz

Tabulka 3.1: Tabulka porovnání vlastností

mezi dvěma zařízeními. Obě technologie komunikují v pásmu 2.4 GHz, ale používají jiný typ modulace. Výhodou WiFi je vyšší rychlost přenosu a větší dosah signálu. Naopak pro Bluetooth mluví nižší cena hardwaru a také menší energetické nároky. Zvláště nízké energetické nároky jsou v zařízeních, ve kterých je Bluetooth použito, velmi důležitým požadavkem. Porovnání vlastností shrnuje tabulka 3.1.

## 3.5 Komunikace

Při komunikaci mezi Bluetooth zařízeními je jedno ustanoveno jako řídicí (master). Na řídicí zařízení se může připojit až 7 podřízených zařízení (slave). Takto vzniká ad-hoc síť, která je označovaná jako piconet. Všechny zařízení v síti piconet se synchronizují s taktem řídicího zařízení.

Data mohou být přenášena mezi nadřazeným zařízením a některým podřízeným. Role v síti se však mohou podle potřeby měnit, velmi rychle se může nadřazené zařízení změnit v podřízené.

### Piconet

Piconet je ad-hoc počítačová síť mezi zařízeními využívající technologii Bluetooth, která umožňuje komunikaci mezi jedním nadřazeným zařízením a až sedmi podřízenými. Až 255 zařízení může čekat na připojení do sítě a může být připojeno v kterémkoli okamžiku. Specifikace dovoluje použití až deseti těchto sítí na ploše o průměru deset metrů.

### Scatternet

Scatternet je síť složená ze dvou a více sítí typu piconet. Umožňuje tedy spojit více než 8 zařízení do jedné sítě. Scatternet může být ustanoven, pokud se člen jedné sítě piconet dostane do dosahu člena jiné sítě piconet. Zařízení patřící do těchto sítí piconet mezi sebou mohou komunikovat přes spojení ustanovené mezi členy, kteří se vidí.

#### 3.5.1 Párování

Párování zajistí bezpečné šifrované spojení mezi dvěma zařízeními, které již není nutné obnovovat při dalším spojení. Toto spojení je vázáno na adresu Bluetooth adaptéru a je možné ho kdykoliv zrušit oběma zařízeními. Na začátku párování je nutné zadat na obou zařízeních stejný kód tzv. PIN. Tento kód slouží jednak k potvrzení souhlasu obou stran s komunikací a je také použit k vytváření klíčů.

Výsledkem párování je 128 bitový klíč, který je uložen v obou zařízeních a slouží k šifrování probíhající synchronizace. Tento klíč je vygenerován kombinací dvou náhodných čísel a adres obou zařízení.

Více informací o párování a zabezpečení přenosu přes Bluetooth lze nalézt na stránkách [3].

### **3.5.2 Zabezpečení**

Zabezpečení dat je v dnešní době velmi důležitým pojmem. Bezpečnost Bluetooth můžeme rozdělit do dvou částí. Za prvé je to zabezpečení přenosu dat a za druhé zabezpečení proti neoprávněnému přístupu ke službám adaptéru.

#### **Zabezpečení přenosu dat**

Protože přenos dat probíhá vzduchem, kde je možné data odposlouchávat, je důležité, aby tato data byla, při neoprávněném odposlechu nečitelná. Z toho důvodu je nutné přenášená data šifrovat. Pro použití šifrování je třeba nejdříve ustanovit bezpečné spojení mezi dvěma zařízeními. Tento proces již byl vysvětlen v sekci věnované párování. Klíč, který je vytvořen, je použit právě k zašifrování probíhající komunikace.

#### **Zabezpečení služeb zařízení**

Pro znemožnění zneužití adaptéru je využito několika vlastností Bluetooth.

- Možnost zakázat nalezení zařízení při vyhledávání. Takové zařízení se může spojit s jinými, ale ostatní zařízení ho při vyhledávání nenajdou. Nemohou se na něj tedy připojit.
- Možnost nastavit jednotlivým službám, jaké zařízení k ní smí přistupovat. Existují celkem tři možnosti nastavení. U služeb s nejvyšším zabezpečením je vyžadována autentizace i autorizace. Nižší stupeň zabezpečení vyžaduje jen autentizaci. Poslední možností je nastavit službu přístupnou pro všechny.

# Kapitola 4

## Protokol OBEX

Object Exchange Protocol je binární komunikační protokol popisující přenos dat mezi dvěma zařízeními jednoduchým a účinným způsobem. Původně byl navržen pro přenos dat přes infračervený port (IrDA), později byl však použit i pro přenos dat v Bluetooth. Pro použití tohoto protokolu musí mít Bluetooth adaptér implementován Generic Object Exchange Profile. OBEX je založen na komunikaci klient - server. Klient se připojí na server a poté požaduje, nebo poskytuje data. Tento princip je podobný jako u známějšího protokolu HTTP. Data jsou posílána ve formátu Big Endian, tedy nejvýznamnější bit je vpravo a nejméně významný bit je vlevo.

### 4.1 Formát příkazu

Každý příkaz, který je odeslán z klienta na server má pevně daný formát, který je zobrazen v tabulce 4.1. Skládá se z Typu, který určuje o jaký příkaz se jedná, dále z délky dat a samotných dat. V položce data může obsahovat další informace, které mají stejný formát jako samotný příkaz.

### 4.2 Operace protokolu OBEX

Každá operace začíná odesláním požadavku na server. Klient poté čeká na odpověď. Dokud odpověď nedorazí, není možné posílat požadavek další.

#### 4.2.1 Operace Spojení (Connect)

OBEX sezení je ustanoveno při prvním požadavku na přenos objektu. Klient nejdříve pošle požadavek na spojení, odpovídající příkaz je v tabulce 4.2. Tento požadavek je zpracován na serveru, který odpoví posláním odpovědi reprezentující úspěch, její formát je v tabulce 4.3. Pokud dojde k poslání nějaké jiné odpovědi, znamená to neúspěch. Jakmile je spojení ustanoveno, zůstává aktivní dokud není zrušeno nebo dokud nedojde k chybě.

Typ	Délka dat	Data
-----	-----------	------

Tabulka 4.1: Formát příkazu



Byt 0 0x80 požadavek na spojení	Byty 1 a 2 Délka požadavku	Byt 3 Verze OBEX
Byt 4 Flagy	Byty 5 a 6 Maximální délka paketu	Byt 7 až nparametry Volitelné

Tabulka 4.2: Požadavek na spojení

Byt 0 Kód odpovědi	Byty 1 a 2 Délka odpovědi	Byt 3 Verze OBEX
Byt 4 Flagy	Byty 5 a 6 Maximální délka paketu	Byt 7 až nparametry Volitelné parametry

Tabulka 4.3: Tvar odpovědi

#### 4.2.2 Operace Odpojení (Disconnect)

Operace odpojení se skládá opět z požadavku a odpovědi. Jejich formáty jsou v tabulkách 4.4 a 4.5.

#### 4.2.3 Operace vložení (Put)

Jakmile je ustanoveno spojení, klient může posílat objekty na server. K tomu slouží operace Put. Formát požadavku na tuto operaci a odpověď na něj je v tabulkách 4.6 a 4.7. Počet požadavků na operaci Put závisí na velikosti objektu a také na velikosti paketu. Každý požadavek musí být serverem potvrzen.

#### 4.2.4 Operace stažení (Get)

Při operaci Get je stažen soubor ze serveru. Tato operace je velmi podobná operaci Put. Rozdíl je jen v kódu operace, který je 0x03 a 0x83 pro finální paket.

#### 4.2.5 Operace nastavení cesty (SetPath)

Tato operace umožňuje měnit aktuální adresář na serveru. Má několik parametrů určujících, kterým směrem se budeme pohybovat. ID spojení obsahuje hodnotu, která je vrácena při operaci připojování. Konstanty jsou vždy nulové, příznaky se mění dle směru pohybu. Formát požadavku popisuje tabulka 4.8.

- Pohybujeme se dopředu. Položka Flagy má hodnotu 0x02.
- Vracíme se na úroveň rodiče. Položka Flagy má hodnotu 0x03.

Byt 0	Byty 1 a 2t	Byt 3
0x81	Délka paketu	Volitelné parametry

Tabulka 4.4: Požadavek na odpojení

Byt 0	Byty 1 a 2t	Byt 3
0xA1	Délka paketu	Volitelné parametry

Tabulka 4.5: Tvar odpovědi

Byt 0	Byty 1 a 2t	Byt 3
0x02 (0x82 pokud se jedná o závěrečný paket)	Délka paketu	Volitelné parametry

Tabulka 4.6: Požadavek na operaci Put

- Skok do kořenového adresáře. Položka Flagy má hodnotu 0x02 a položka se jménem je prázdná.
- Vytvoření adresáře. Položka Flagy má hodnotu 0x00 a položka se jménem obsahuje jméno vytvářeného adresáře.

Jakmile server obdrží požadavek na změnu cesty, pokusí se ho provést. Podle výsledku operace pošle odpověď zpět klientovi. Následující tabulka zobrazuje možné odpovědi.

#### 4.2.6 Operace přerušení (Abort)

Operace Abort ukončí aktuálně prováděnou operaci.

### 4.3 Profily využívající protokol OBEX

Protokol OBEX je využíván v několika dalších profilech, které určují jeho uplatnění. Tyto profily jsou:

- Object Push Profile slouží pro přenos objektů, jako jsou například vizitky.
- Synchronization Profile slouží k synchronizaci kontaktů, poznámek apod. mezi dvěma zařízeními.
- Basic Printing Profile umožňuje tisk krátkých textů, jako jsou emaily, vizitky a textové zprávy přímo z mobilních zařízení.
- Basic Imaging Profile umožňuje tisk, přenos a změnu některých parametrů u obrázků.
- File Transfer Profile umožňuje přenos souborů mezi dvěma zařízeními. Tento profil je podrobně popsán v následující kapitole.

Ne všechny zde vyjmenované profily musí zařízení, jež má implementován protokol OBEX, podporovat.

Byt 0	Byty 1 a 2t	Byt 3
Kód odpovědi	Délka paketu	Volitelné parametry

Tabulka 4.7: Tvar odpovědi

0x85	Délka paketu	Flagy	Konstanty	ID spojení	Jméno složky
------	--------------	-------	-----------	------------	--------------

Tabulka 4.8: Tvar požadavku na změnu cesty

Kód	Popis
0xA0	Úspěch
0xC4	Požadovaný adresář neexistuje. U návratu zpět značí, že jsme již v kořenovém adresáři
0xC1	Procházení adresářů je zakázáno. Adresář je jen pro čtení při vytváření adresáře

Tabulka 4.9: Tvar požadavku na změnu cesty

### 4.3.1 File Transfer Profile

File Transfer Profile umožňuje operace se soubory na serveru. Nejčastěji tento profil podporují zařízení jako PDA, notebooky a klasické PC. Stále častěji se však objevuje i v mobilních telefonech. Profil poskytuje tři možnosti využití:

- Zobrazení adresářů na serveru a jejich procházení.
- Přenos souborů ze serveru na klienta a naopak.
- Manipulaci ze soubory na serveru. Umožňuje mazat a vytvářet soubory a složky.

## Kapitola 5

# Vývoj aplikací pro Windows Mobile

Vývoj aplikací pro Microsoft Windows Mobile je velmi podobný vývoji pro klasický Microsoft Windows. Aplikační rozhraní obsahuje obdobné funkce jako u klasického win32 win API. Rozdíly jsou jen v absenci některých funkcí, které nebyly implementovány. Například některé funkce pro práci s adresáři. Dalším rozdílem jsou velmi omezené zdroje, které má program k dispozici, oproti klasické PC platformě. Především se jedná o velikost paměti a úložného prostoru. S tímto omezením je potřeba počítat a přizpůsobit jim návrh programu.

### 5.1 Softwarové požadavky

U následujících požadavků se předpokládá vývoj pod operačním systémem Microsoft Windows XP a novější. U starších verzí nebo pod jiným operačním systémem nemusí být tyto požadavky dostačující pro vývoj aplikací. Jednotlivé požadavky:

- Windows XP a vyšší
- Některé vývojové prostředí od Microsoftu: Visual Studio nebo eMbedded Visual C++
- Software Development Kit (SDK) pro příslušný cílový operační systém. V případě použití Visual Studia 2005 a vyšší je již SDK pro Windows Mobile 2003 jako volitelná položka při instalaci.

### 5.2 Vývojová prostředí

V této kapitole jsou popsána vývojová prostředí. Zmíněna jsou pouze vývojová prostředí od firmy Microsoft.

- eMbedded Visual Tools 3.0 je starší vývojové prostředí určené pro vývoj pro operační systém Windows Mobile 2002. Obsahuje grafické rozhraní, integrovaný překladač a ladící nástroje.
- eMbedded Visual C++ 4.0 je odlehčenou verzí Visual Studia. Obsahuje možnosti eMbedded Visual Tools, které obohacuje o několik funkcí, například nástroj Just in Time debugger, který umožňuje ladění programu i v případě, že nebyl spuštěn z vývojového prostředí.

- Visual Studio 2003 a vyšší je plnohodnotné vývojové prostředí, které není zaměřeno jen na vývoj pro operační systém Windows Mobile, ale i pro klasické Windows.

### 5.2.1 Ladění

Nalezení chyb v programu je velmi důležitá část v procesu vývoje. Ve zmíněných vývojových prostředích je několik nástrojů, které tuto činnost usnadňují. Nejdůležitějším z těchto nástrojů je bezpochyby integrovaný debugger, který umožní krokování kódu, výpis hodnot proměnných a jejich modifikaci.

Součástí každého SDK je i emulátor přístroje s příslušným operačním systémem, který umožňuje téměř plnohodnotnou náhradu fyzického přístroje. Problém nastává v případě, že v programu potřebujeme pracovat s periferními zařízeními, nebo s některými integrovanými moduly jako jsou GPS, Bluetooth nebo GSM. Stoprocentní emulace zatím není dostupná.

Další možností je program spustit přímo na existujícím zařízení. Pro tento účel je nutný program ActiveSync, který zajistí přenos dat mezi vývojovým prostředím a přístrojem.

## 5.3 Zdroje informací a dokumentace

Základním zdrojem informací pro vývoj jsou zdroje poskytované firmou Microsoft na serveru MSDN [6], kde se nachází dokumentace knihoven funkcí, nástin postupů, a také příklady již hotového kódu. Existuje množství další stránek zabývajících se touto problematikou, např. The Code Project [5], kde se nachází velké množství zdrojových kódů.

# Kapitola 6

## Rozbor problému

Cílem je navrhnout aplikaci, která bude schopná synchronizovat adresáře mezi Pocket PC a serverem. Synchronizace bude probíhat pomocí technologie Bluetooth a protokolu OBEX. Aplikace bude určena pro operační systém Microsoft Windows Mobile 5.0 nebo Microsoft Windows Mobile 6.0.

### 6.1 Komunikace

Pro přenos dat byla zvolena technologie Bluetooth a funkce, které poskytuje File Transfer Profile. Podrobnější popis těchto technologií byl předmětem předchozích kapitol.

### 6.2 OBEX FTP a jeho omezení

OBEX FTP slouží k přenosu souborů mezi serverem a klientem. Umožňuje také zobrazit strukturu adresáře a přerušit aktuálně prováděnou operaci. Podrobnosti o možnostech tohoto profilu jsou v kapitole o protokolu OBEX.

#### 6.2.1 Informace o souboru na serveru

Informace o struktuře adresáře jsou ze serveru posílány jako odpověď na požadavek Folder Browsing. Odpovědí je dokument ve formátu XML. Zde je uveden příklad takového dokumentu, který je získán z adresáře, který obsahuje jeden adresář Directory a jeden soubor File.dat.

```
<?xml version="1.0"?>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version="1.0">
  <parent-folder />
  <folder name="Directory" modified="20080408T162002Z" created="20080408T162001Z" />
  <file name="File.dat" size="240395" modified="20080408T162010Z"
  created="20080408T162005Z" accessed="20080407T220000Z" />
</folder-listing>
```

Kromě jména souboru a označení, zda se jedná o adresář, nebo soubor, je zde ještě několik parametrů.

- size udává velikost souboru v bytech.

- modified je čas poslední změny souboru.
- create je čas, kdy byl soubor vytvořen.
- accessed je čas posledního přístupu do souboru.

Některé implementace serveru mohou posílat i další položky, například přístupová práva k souborům a typ média, na kterém jsou data uložena, nebo naopak nemusí posílat čas změny a posledního přístupu.

### 6.2.2 Omezení

Vzhledem k malému množství funkcí je nutné složitější operace provést pomocí více operací jednodušších. Vzniká zde prostor pro vznik chyb. Je nutné ošetřit situace, kdy se mezi jednotlivými operacemi vyskytne problém a nebude možné dokončit celou zamýšlenou úlohu. Mezi tyto operace patří například operace změny části souboru, nebo přejmenování souboru, u kterých je nutný přenos na klienta a poté, po úpravě, zpět na server.

V ideálním případě je při nemožnosti kompletního dokončení proveden okamžitý návrat do původního stavu na obou stranách, tedy jak na klientovi, tak na serveru. Tato možnost však většinou není dosažitelná. Je tedy nutné uložit dostatečné informace, aby bylo možné při dalším přístupu nekompletní operaci rozpoznat a zabránit přenosu neplatných dat.

## 6.3 Synchronizace

Hlavním úkolem aplikace je porovnání zadaných adresářů. V případě, že jsou adresáře rozdílné, aplikace přenesou soubory tak, aby tyto rozdíly odstranil. Kontrola musí proběhnout při nalezení zařízení a poté se musí periodicky opakovat ve stanovených časových intervalech, dokud se zařízení nachází v dosahu.

### 6.3.1 Porovnání souborů

První operací při samotné synchronizaci je porovnání dvou souborů. Při porovnání se zjišťuje, ve kterém souboru došlo ke změně.

### 6.3.2 Konfliktní stav

Při konfliktu souborů má stejný soubor na serveru i na klientovi jiný obsah. Není tedy možné přenést jeden soubor bez toho, aby byla vyloučena možnost ztráty dat. Tyto stavy je potřeba detekovat a řešit.

### 6.3.3 Současný přístup více zařízení

Pokud se o přístup k serveru snaží v jednom okamžiku více klientů, musí být zajištěno, aby nedošlo k současnému přístupu do jednoho adresáře. Pokud by k této situaci došlo, mohlo by to způsobit nekonzistentní stav adresáře, při kterém by část souborů, která zůstala nezměněna, byla starší verze, a část, která se nahrála z klienta, byla verze novější. Vzniku tohoto stavu je nutné zabránit.

### 6.3.4 Chybové stavy

V průběhu synchronizace může dojít k situacím, při kterých nebude možné operaci dokončit. Může dojít ke ztrátě spojení při vypnutí zařízení, opuštění dosahu signálu, nebo z nějakého jiného důvodu. Další problém nastane v případě nedostatku místa na disku při přenosu dat. Uživatel má také možnost aplikaci ukončit a znemožnit tak dokončení synchronizace.

Při všech těchto případech musí být zajištěno, aby nedošlo ke ztrátě dat, nebo používání nekompletního souboru.

## 6.4 Konfigurace

Aplikace musí umožňovat několik volitelných nastavení. Pro nastavení je nutné vytvořit rozhraní, které bude umožňovat jejich změnu. Seznam volitelných nastavení je uveden níže.

- Volba zařízení, se kterými má probíhat synchronizace.
- Nastavení adresářů, ke každému zařízení, se kterými má probíhat synchronizace
- Možnost nastavit logování chyb, informací o přenesených souborech
- Nastavení chování v případě konfliktu



# Kapitola 7

## Popis implementace

Celý projekt je rozdělen do dvou aplikací. První aplikace může běžet na pozadí a zajišťuje synchronizaci souborů a druhá aplikace slouží ke konfiguraci přenosů, jak byla popsána v sekci o popisu konfigurace v předchozí kapitole. Obě aplikace jsou implementovány za použití programovacího jazyka C++ s využitím vývojového prostředí Microsoft Visual Studio 2005 a SDK pro Windows Mobile 5.0.

### 7.1 Bluetooth

#### 7.1.1 Rozhraní pro OBEX

V SDK pro Windows Mobile jsou k dispozici rozhraní objektů, které usnadňují práci s protokolem OBEX. Přehled použitých rozhraní je v tabulce 7.1.

#### 7.1.2 Vyhledávání zařízení

K vyhledávání zařízení je možné použít několik metod. Funkce `IObex::StartDeviceEnum`, která je zde využita, spustí vyhledávání, které běží neustále na pozadí. S využitím rozhraní `IObexSink` je možné přepsat funkci `IObexSink::Notify`, která je volána vždy při události na zařízení. Událostí je myšleno například objevení zařízení, ztráta zařízení z dosahu, nebo změna na vzdáleném zařízení.

#### 7.1.3 Připojení

Jedním z parametrů funkce `Notify` je i proměnná `PropertyBag`, která obsahuje informace o vzdáleném zařízení, s jejichž pomocí je možné zařízení identifikovat a připojit se na něj.

<code>IObex</code>	Poskytuje primární rozhraní pro přístup k objektu OBEX.
<code>IDeviceEnum</code>	Rozhraní slouží k vyhledávání vzdálených zařízení.
<code>IObexDevice</code>	Rozhraní slouží k přenosu dat z a na konkrétní zařízení.
<code>IObexSink</code>	Rozhraní umožňuje nastavit oznámení v případě události na zařízení.
<code>IHeaderCollection</code>	Rozhraní sloužící k manipulaci s objekty hlaviček.
<code>IPropertyBagEnum</code>	Rozhraní umožňující přístup k proměnným třídy <code>PropertyBag</code> .

Tabulka 7.1: Přehled použitých rozhraní

Před samotným připojením je potřeba svázat vzdálené zařízení s rozhraním IObexDevice. To zajistí funkce IObex::BindToDevice. Poté je již možné se pomocí funkce IObexDevice::Connect připojit.

#### 7.1.4 Přenos dat

K přenosu dat je využito rozhraní IObexDevice a to konkrétně funkce Put, pro přenos dat na server, a Get, pro přenos opačným směrem. Přenos dat je možné přerušit funkcí Abort.

## 7.2 Synchronizace

Postup při synchronizaci je znázorněn na následujícím diagramu. Jedná se o postup při synchronizaci jednoho již připojeného zařízení, ve kterém je synchronizován jeden adresář. V případě více zařízení s více adresáři by byl postup obdobný, jen by se opakoval vícekrát. Postup synchronizace je znázorněn na obrázku [A.2](#).

### 7.2.1 Informace o synchronizaci adresáře

Ještě předtím než rozebereme porovnání souborů, je nutné popsat princip, jakým jsou ukládány informace o již dokončených synchronizacích. Po proběhnutí synchronizace adresáře je v něm uložen soubor se strukturou, která je znázorněna v tabulce [7.1](#):

Jméno souboru
Čas modifikace na serveru
Čas modifikace na klientovi

Obrázek 7.1: Struktura souboru s informacemi o adresáři

O každém souboru jsou zde uloženy informace o poslední změně, která v něm proběhla, a to jak na serveru, tak na klientovi. Tyto informace umožňují porovnávat soubory při synchronizaci a určovat, u kterých došlo ke změně, aniž by bylo nutné soubor stahovat. Postačí nám tedy data, která nám poskytne server při požadavku na obsah adresáře.

### 7.2.2 Řešení porovnání souboru

Pro porovnání souborů jsou implementovány dva postupy. Výběr postupu, který bude použit, záleží na několika faktorech. V některých případech mohou být dokonce použity oba. Více o těchto postupech i o tom, v jakých situacích jsou použity je popsáno v následujících podkapitolách.

#### Porovnání již synchronizovaného souboru

Tento postup je ve většině případů použit jako první. Základem je soubor s informacemi o předchozí synchronizaci popsáný v sekci 7.2.1. Nejprve jsou získány aktuální informace o souborech na serveru a na klientovi v příslušném adresáři. Tyto informace jsou poté

porovnávány s informací ze souboru s daty o synchronizaci. Pokud došlo v souboru ke změně, data se liší a je možné vyhodnotit, který soubor byl změněn a provést přenos.

Bohužel některé přístroje, například mobilní telefony, neukládají čas poslední změny v souboru, takže popsáný postup zde nelze použít.

### **Kontrolní součet**

Předchozí postup však může selhat. Pokud není soubor s informacemi z předchozí synchronizace k dispozici, ať už z důvodu, že se jedná o první synchronizaci, nebo byl poškozen, je nemožné jej aplikovat. V tomto případě je nutné soubor ze serveru stáhnout a uložit do dočasného souboru. Poté je vypočten kontrolní součet u staženého souboru a u souboru na klientovi. Kontrolní součty jsou porovnány a pokud se nerovnájí, jsou soubory v konfliktu. Pokud se rovnají, jsou soubory identické a není třeba je přenášet.

Pro výpočet kontrolního součtu byl použit algoritmus MD5, jehož detailní popis je k nalezení například v dokumentu rfc 1321 [8]. Funkce pro výpočet je napsána v jazyce C a je vložena do samostatné dynamické knihovny, která musí být distribuována spolu s programem.

Tento postup je však již časově náročnější, než předchozí. Navíc zde může dojít k problému s nedostatkem místa na disku. Soubor se vyskytuje na disku dvakrát. Jednou v původním místě a podruhé jako dočasný soubor.

### **Řešení konfliktu souborů**

Program není schopen v případě konfliktu souborů vyhodnotit, který soubor se má přenášet. Musí být nastaveno pravidlo, jak se v tomto případě chovat. Možnosti nastavení chování jsou popsány v uživatelské příručce, která je umístěna v příloze tohoto dokumentu.

### **Vyhodnocování samostatného souboru**

V případě, že se soubor nalézá na jednom ze zařízení, které se synchronizují, je automaticky přenesen na zařízení, na kterém se nenachází.

#### **7.2.3 Synchronizace více zařízení**

U každého zařízení, které se objeví v dosahu, je vyhodnoceno, zda se má dojít k synchronizaci. Pokud ano, je vloženo do fronty. Jednotlivá zařízení ve frontě jsou postupně synchronizována. Jakmile jsou synchronizována všechna zařízení, spustí se časovač, který zajistí opětovné spuštění synchronizace po požadované době.

Pokud se zařízení vzdálí z dosahu signálu, nebo jinak odstraněno z viditelnosti klienta, dojde k jeho vyjmutí z fronty. Také v případě, že dojde při synchronizaci k chybě je zařízení z fronty odstraněno.

### **Uzamykání adresářů**

Zamykání adresáře slouží k zamezení přístupu více klientů k jednomu adresáři na serveru. Probíhá formou nahrání souboru, jehož existence je klientem testována před každým pokusem o přístup k adresáři.

Zámek je na server nahrán před začátkem přenosu dat a poté je aktualizován před přenosem každého souboru. Soubor obsahuje dvě informace. První z nich je odhad doby, po kterou bude trvat přenos aktuálně přenášeného souboru. Druhá je k dispozici v případě, že

je soubor na server nahráván a je to jméno tohoto souboru. Slouží v případě nedokončeného přenosu k identifikaci a následnému smazání tohoto souboru. Postup vyhodnocení, zda je adresář uzamčen, je na diagramu [A.1](#).

#### 7.2.4 Ošetření chybových stavů

U všech případů, kdy nedojde k dokončení synchronizace, je důležité mít možnost identifikovat soubor, který nebyl celý přenesen a tento kompletní soubor smazat. Z důvodu identifikace je vždy uloženo jméno aktuálně přenášeného souboru do souboru v adresáři, do kterého je soubor nahráván. V případě, že bude přenos neočekávaně ukončen bude nekompletní soubor rozpoznán při příštím pokusu o synchronizaci.

### 7.3 Konfigurace

Pro konfiguraci synchronizace je vytvořena samostatná aplikace, při jejíž implementaci byla použita knihovna MFC [7]. Použití knihovny MFC při programování aplikací popisuje kniha [2] Nastavení je ukládáno do registrů systému Windows do klíče:

```
HKEY_CURRENT_USER\Software\ObexFTP synchronize\ObexConfig\
```

Z registru jsou nastavení načteny aplikací pro synchronizaci. Nastavení, které zařízení synchronizovat a nastavení adresářů na nich, jsou uloženy v samostatných souborech.

### 7.4 Instalace

Pro vytvoření instalačního balíčku byl zvolen soubor ve formátu cab. Jedná se o speciální typ archivu, který umožňuje uchovat všechny potřebné informace o požadovaném umístění jednotlivých souborů, které obsahuje. Také je možné uložit data, která se mají vložit do registru operačního systému. Soubory typu cab jsou operačním systémem Windows Mobile podporovány a instalace z nich proběhne automaticky.

# Kapitola 8

## Závěr

Byla vytvořena aplikace, která umožňuje synchronizaci zadaných adresářů mezi serverem a klientem pomocí technologie Bluetooth. Při implementaci byly využity znalosti získané v průběhu studia, především z předmětů, které se týkaly programování a algoritmizace.

### 8.1 Možná rozšíření do budoucna

Program splňuje požadovanou funkčnost. I přesto je zde několik možností, jak dosáhnout zvýšení uživatelského komfortu a vylepšení schopností programu.

V této podobě je velmi nepřesné porovnávání času poslední změny v případě konfliktu souborů. Čas není načten ze serveru a pokud nejsou nastavení stejná, nemusí být vyhodnocení správné. Může tedy dojít k nahrání staršího souboru i při požadavku na soubor novější.

Dalším možným vylepšením je možnost synchronizovat pouze jednotlivé soubory, bez nutnosti synchronizovat celý adresář.

Pokud by měl být program použit pouze pro zálohování dat, bylo by to v současné konfiguraci velmi problematické. Pro tento účel by bylo nutné implementovat možnost pouze jednosměrné synchronizace, aby nedocházelo k přepisu souborů z místa zálohy na klienta.

### 8.2 Testování

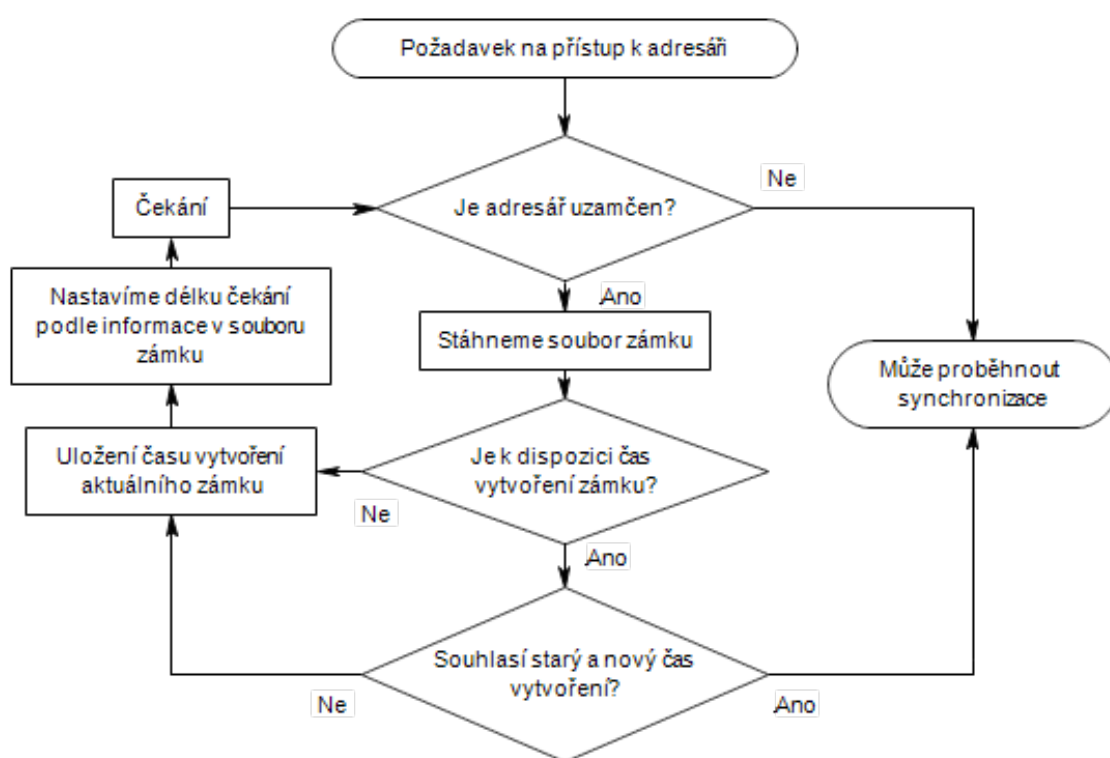
Testování programu probíhalo na operačním systému Microsoft Windows Mobile 6.0 na přístroji E-ten X500. Jako server bylo použito několik zařízení: HP nx7100 s integrovaným Bluetooth adaptérem, Bluetooth adaptér MSI BToes 2.0 a mobilní telefony Nokia 3110c a Nokia 6234.

# Literatura

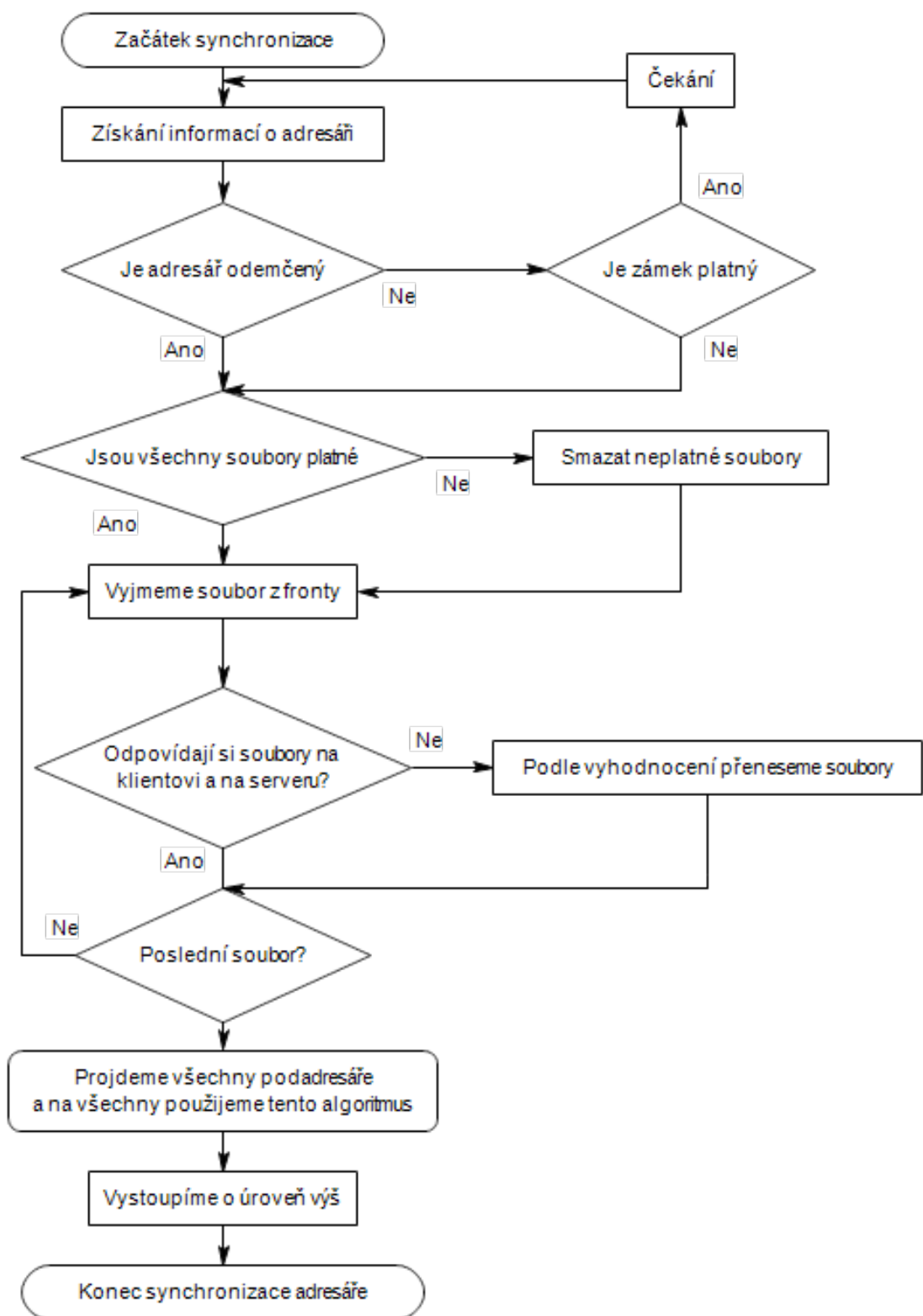
- [1] Bluetooth Special Interest Group. *Bluetooth specification version 1.1*. 2002.
- [2] Wingo Scot Kruglinski David, Shepherd George. *Programujeme v Microsoft Visual C++*. Computer Press, 2002. ISBN 80-7226-362-5.
- [3] Wool Avishai Shaked Yaniv. Cracking the bluetooth pin.  
<http://www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/>.
- [4] WWW stránky. Bluetooth special interest group oficial site.  
<http://www.bluetooth.com/bluetooth/>.
- [5] WWW stránky. The code project. <http://www.codeproject.com/KB/mobile/>.
- [6] WWW stránky. Microsoft developer network.  
[http://msdn2.microsoft.com/cs-cz/default\(en-us\).aspx](http://msdn2.microsoft.com/cs-cz/default(en-us).aspx).
- [7] WWW stránky. Microsoft fundation class.  
[http://msdn2.microsoft.com/en-us/library/d06h2x6e\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/d06h2x6e(VS.80).aspx).
- [8] WWW stránky. Rfc 1321 - the md5 message-digest algorithm.  
<http://www.faqs.org/rfcs/rfc1321.html>.

## Dodatek A

# Vývojové diagramy



Obrázek A.1: Diagram vyhodnocení zámku adresáře



Obrázek A.2: Diagram algoritmu synchronizace



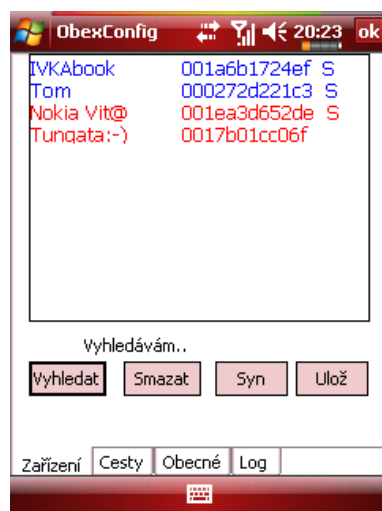
## Dodatek B

# Uživatelská příručka

### B.1 Konfigurační aplikace

Tato aplikace slouží ke konfiguraci synchronizace. Obsahuje čtyři panely, každý z nich je zaměřen na nastavení spolu souvisejících vlastností. Konfigurace je načítána před každým pokusem o synchronizaci a při startu aplikace pro synchronizaci.

#### B.1.1 Nastavení vzdálených zařízení



Obrázek B.1: Panel nastavení zařízení

Hlavní panel obsahuje seznam zařízení, se kterými již klient někdy navázal kontakt. Každá položka má formát, který je na obrázku B.1. Modrá barva označuje zařízení, které je v dosahu, červená to, které není viditelné. Pro umístění zařízení do tohoto seznamu je nutné jej nejdříve vyhledat. K tomuto účelu slouží tlačítko vyhledat, které jednak spustí Bluetooth, pokud již není spuštěné a pak začne vyhledávat zařízení v dosahu.

Tlačítko Syn, slouží k nastavení zařízení do stavu synchronizovat, nebo nesynchronizovat. V případě, že je zobrazeno písmeno S, zařízení se bude s klientem synchronizovat.

Zařízení je možné ze seznamu odstranit tlačítkem smazat. Při změně v seznamu je nutné tento seznam uložit tlačítkem Ulož.

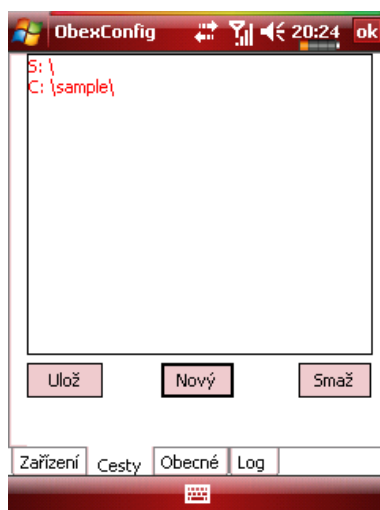
Jméno	Adresa	Písmeno S, označující, že se má se zařízením synchronizovat
-------	--------	---

Tabulka B.1: Položka se zařízením

S:	Cesta na serveru
C:	Cesta na klientovi

Tabulka B.2: Cesty k adresářům určených k synchronizaci

### B.1.2 Nastavení adresářů



Obrázek B.2: Panel nastavení adresářů k synchronizaci

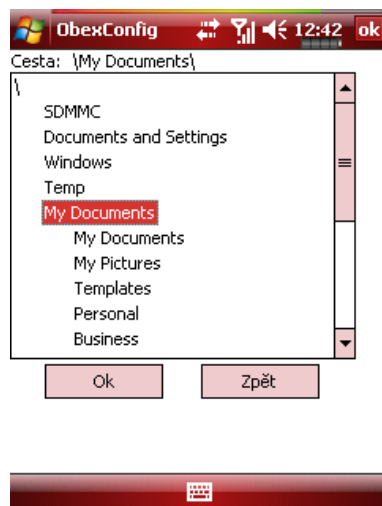
Tato sekce slouží k nastavení cest, které se mají synchronizovat na konkrétním zařízení. Hlavní panel obsahuje informace o aktuálně nastavených cestách ve formátu zobrazeném na obrázku B.2. Zařízení, o kterém jsou informace zobrazeny, je vybíráno na předchozím panelu Zařízení.

Novou cestu nastavíme pomocí tlačítka Nový, které zobrazí dialog výběru cest, zobrazený na obrázku B.4. Možnost je aktivní pouze v případě, že je vybráno zařízení na panelu nastavení zařízení a toto zařízení je v dosahu.

Další dvě možnosti jsou Ulož a Smaž, které slouží k uložení současného nastavení a ke smazání vybrané položky.

Obrázek B.3 zobrazuje dialog pro výběr adresáře na klientovi. Tlačítko Ok potvrzuje volbu, tlačítko Zpět vrací na předchozí nabídku a ruší výběr.

Dialog umožňuje výběr cest na serveru a na klientovi, které se mají synchronizovat. Horní část umožňuje vybrat cestu na klientovi. Dolní část slouží k procházení adresářů na serveru. Jakmile jsou vybrány cesty, tak tlačítko OK potvrdí volbu. Tlačítkem Zpět se lze vrátit na panel pro konfiguraci adresářů.



Obrázek B.3: Dialog pro výběr adresáře na klientovi

### B.1.3 Nastavení obecných vlastností

- Nastavení akce, která se má provést v případě konfliktu souborů na serveru a na klientovi.
- Nastavení adresářů, které se mají synchronizovat v případě, že nejsou žádné nastaveny.
- Poslední volbou je cesta pro dočasné uložení staženého souboru, v případě požadavku na porovnání obsahu souboru na klientovi a na serveru.

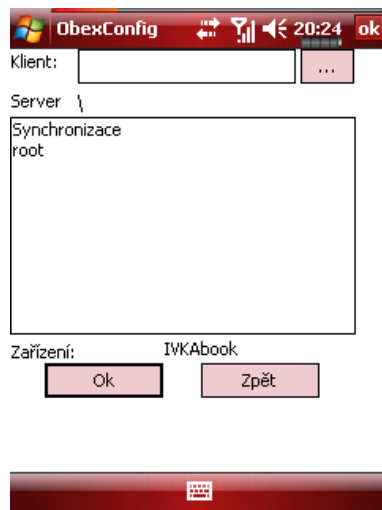
### B.1.4 Nastavení logu

Posledním panelem je panel pro nastavení logu. Na tomto panelu se nacházejí následující volby

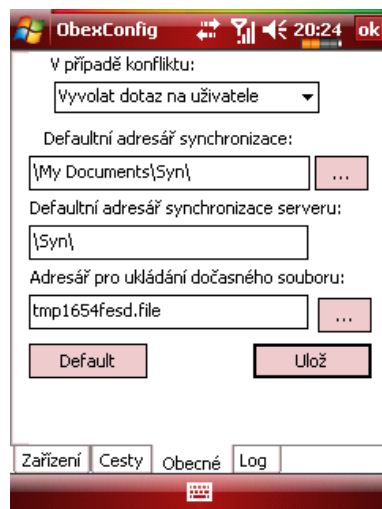
- Nastavení souboru pro ukládání chybových hlášek.
- Nastavení souboru pro ukládání logu synchronizace.
- Povolení logu chyb.
- Povolení logu synchronizace.
- Nastavení zda ukládat čas vzniku události do logu.
- Nastavení zda logovat i soubory, které nejsou přenášeny.

## B.2 Synchronizační aplikace

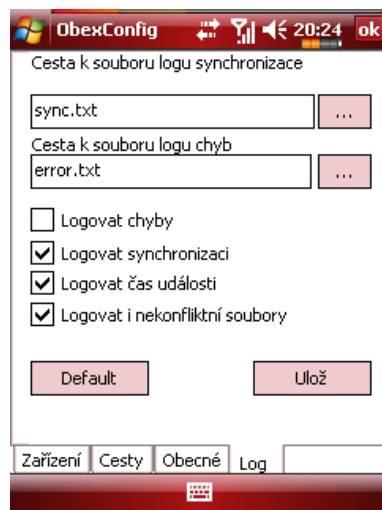
Tato část je po uživatelské stránce velmi jednoduchá. Při spuštění aplikace je pouze nutné spustit synchronizaci tlačítkem START. V průběhu synchronizace je možné přerušit právě probíhající akci tlačítkem Zrušit, které je umístěno na dialogu indikace průběhu přenosu souboru. Aplikaci je možné ukončit tlačítkem KONEC.



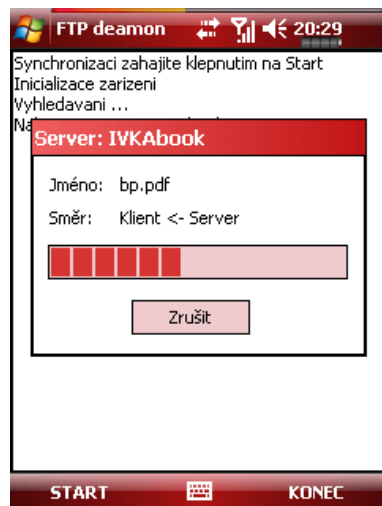
Obrázek B.4: Dialog výběru cest k synchronizaci



Obrázek B.5: Panel nastavení obecných vlastností



Obrázek B.6: Panel nastavení logu



Obrázek B.7: Obex Daemon

## Dodatek C

# Obsah CD

Na CD se nachází několik složek. Zde je popis jejich obsahu:

- Instalacni balicek - obsahuje dva instalační balíčky pro Windows Mobile 6.0 a 2003
- md5DLLCE - obsahuje zdrojový kód funkci pro výpočet MD5
- ObexConfig - obsahuje zdrojový kód konfigurační aplikace
- ObexDaemon - obsahuje zdrojový kód synchronizační aplikace
- ObexInstall - obsahuje zdrojový kód pro vytvoření instalačního balíčku
- Spustitelne soubory - obsahuje spustitelné soubory pro WM 6.0 a WM 2003
- Technicka zprava - obsahuje technickou zprávu a její zdrojový kód