

Tvorba a bezpečná distribuce instalačních balíčků na platformě Microsoft Windows

The Development and Secure Distribution of Installation Packages
for the Microsoft Windows Platform

Bc. Antonín Blaha

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Antonín BLAHA**
Osobní číslo: **A10410**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Tvorba a bezpečná distribuce instalačních balíčků na platformě Microsoft Windows**

Zásady pro vypracování:

1. Zpracujte technickou dokumentaci pro technologii Microsoft Windows Installer.
2. Popište techniky tvorby a úpravy instalačních balíčků.
3. Vytvořte ukázkový instalační balíček.
4. V ukázkové instalaci implementujte kontrolu integrity vstupních souborů spolu s implementací elektronického podpisu.
5. Prostřednictvím ukázkové instalace proveďte zavádění a následnou správu software prostřednictvím zásad skupin a pomocí nástroje System Center Configuration Manager.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **WILSON, Phil.** The definitive guide to Windows Installer. New York: Distributed to the Book trade in the United States by Springer-Verlag, c2004, 301 s. ISBN 15-905-9297-2.
2. **BAKER, Bob.** The official Installshield for Windows Installer developers guide. Foster City, Calif.: M, 2001, 950 s. ISBN 07-645-4723-2.
3. **DOSTÁLEK, Libor a Marta VOHNOUTOVÁ.** Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. vyd. 1. Brno: Computer Press, 2006, 534 s. ISBN 80-251-0828-7.
4. **STANEK, William R.** Group Policy: zásady skupiny ve Windows : kapesní rádce administrátora. Vyd. 1. Brno: Computer Press, 2010, 351 s. ISBN 978-802-5129-203.
5. **MOSBY, Chris, Ron D CRUMBAKER a Christopher W URBAN.** Mastering System Center Configuration Manager 2007 R2. Indianapolis, Ind.: Wiley Pub., c2009, 595 s. Serious skills. ISBN 04-701-7367-X.

Vedoucí diplomové práce:

Ing. Dalibor Slovák

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

24. února 2012

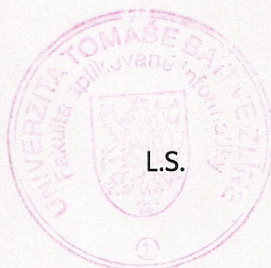
Termín odevzdání diplomové práce:

15. května 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.

děkan



doc. RNDr. Vojtěch Křesálek, CSc.

ředitel ústavu

ABSTRAKT

Práce se zabývá způsobem nasazení aplikací, převážně se zaměřuje na technologii Windows Installer. Seznamuje čtenáře s danou problematikou. Rozšiřuje problematiku o postupy vývoje pomocí dostupných nástrojů. Seznamuje čtenáře s možnostmi nasazení pomocí nástroje System Center Configuration Manager.

Klíčová slova: Windows Installer, MSI, MSP, MST, Digitální podpis, Hash, SCCM, System Center Configuration Manager, Microsoft, Windows

ABSTRACT

This thesis deals with the ways of applications deployment, it mainly focuses on Windows Installer technology. It informs the reader of the given topic. It extends the issue by development processes using available tools. It introduces readers with the possibilities of deployment using System Center Configuration Manager.

Keywords: Windows Installer, MSI, MSP, MST, Digital signature, Hash, SCCM, System Center Configuration Manager, Microsoft, Windows

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu mé diplomové práce Ing. Daliboru Slovákovi za možnost pracovat na tématu této práce a za cenné konzultace a rady, které mi poskytl. Také bych chtěl poděkovat své rodině, která mě po celou dobu studia podporovala.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 30.9.2012

.....
podpis diplomanta

OBSAH

| | |
|---|-----------|
| ÚVOD | 10 |
| I TEORETICKÁ ČÁST | 11 |
| 1 PROBLEMATIKA TVORBY INSTALAČNÍCH BALÍČKŮ | 12 |
| 1.1 5 DŮVODŮ, PROČ BY MĚL PROGRAM OBSAHOVAT INSTALACI..... | 12 |
| 1.1.1 Jednoduchost a přehlednost..... | 12 |
| 1.1.2 Četnější používání..... | 12 |
| 1.1.3 Dostupnost..... | 13 |
| 1.1.4 Zpětná vazba přes web | 13 |
| 1.1.5 Windows je navrženy pro instalaci aplikací..... | 13 |
| 1.2 OBSAH INSTALAČNÍHO BALÍČKU..... | 13 |
| 2 SEZNÁMENÍ S TECHNOLOGIÍ WINDOWS INSTALLER | 15 |
| 2.1 WINDOWS INSTALLER - OTEVŘENÁ ARCHITEKTURA | 17 |
| 2.2 SPRÁVA SDÍLENÝCH ZDROJŮ | 18 |
| 2.3 SPRÁVA INTEGRITY POČÍTAČE | 18 |
| 2.4 ÚPRAVY INSTALACÍ..... | 18 |
| 2.5 OPRÁVNĚNÍ INSTALACE..... | 19 |
| 2.6 WINDOWS INSTALLER DATABÁZE..... | 19 |
| 2.7 TABULKY | 20 |
| 2.7.1 Tabulka Property | 20 |
| 2.7.2 Tabulka Feature..... | 20 |
| 2.7.3 Tabulka Component..... | 21 |
| 2.7.4 Tabulka File..... | 22 |
| 2.7.5 Ostatní tabulky | 23 |
| 2.8 GUID | 23 |
| 2.9 MOŽNOSTI SLUŽBY WINDOWS INSTALLER MSIEXEC | 23 |
| 2.9.1 Možnosti instalace..... | 23 |
| 2.9.2 Možnosti zobrazení | 24 |
| 2.9.3 Možnosti restartování..... | 24 |
| 2.9.4 Možnosti protokolování | 25 |
| 2.9.5 Možnosti aktualizace..... | 25 |
| 2.9.6 Možnosti opravy..... | 26 |
| 2.10 AUTOMATIZACE INSTALACE | 26 |
| 2.11 UPDATY, UPGRADY A PATCHE | 27 |
| 2.11.1 Patche | 29 |
| 2.12 MERGE MODULE..... | 29 |
| 2.13 CUSTOM ACTION - VLASTNÍ AKCE | 29 |
| 2.14 BOOTSTRAPPER | 30 |
| II PRAKTICKÁ ČÁST | 32 |

| | | |
|----------|---|-----------|
| 3 | INTEGRITA VSTUPNÍCH SOUBORŮ | 33 |
| 3.1 | UKÁZKA APLIKACE..... | 34 |
| 4 | UKÁZKA TVORBY INSTALAČNÍHO BALÍČKU | 36 |
| 4.1 | INSTALLSHIELD..... | 36 |
| 4.1.1 | Installation Information..... | 36 |
| 4.1.2 | General Information | 37 |
| 4.1.3 | Organization | 37 |
| 4.1.4 | Ukázka Wizardu Component | 39 |
| 4.1.5 | Vytváření registrů..... | 40 |
| 4.1.6 | Vytváření služeb..... | 41 |
| 4.1.7 | ODBC Resources | 42 |
| 4.1.8 | Shortcuts..... | 42 |
| 4.1.9 | Release | 44 |
| 5 | DIGITÁLNÍ PODPIS INSTALAČNÍCH BALÍČKŮ | 45 |
| 5.1 | TECHNOLOGIE ŠIFROVÁNÍ..... | 45 |
| 5.2 | TECHNOLOGIE DIGITÁLNÍHO PODPISU | 46 |
| 5.3 | VÝHODY PODPISU INSTALAČNÍCH BALÍČKŮ..... | 46 |
| 5.4 | PODPIS INSTALAČNÍCH BALÍČKŮ NA PLATFORMĚ WINDOWS | 47 |
| 5.4.1 | Certifikační autorita..... | 47 |
| 5.4.2 | Obecný proces vydání certifikátu | 48 |
| 6 | NASZENÍ POMOCÍ GROUP POLICY | 49 |
| 6.1 | UKÁZKA DISTRIBUCE POMOCÍ GROUP POLICY | 50 |
| 7 | SPRÁVA HROMADNÉ DISTRIBUCE INSTALAČNÍCH BALÍČKŮ POMOCÍ NÁSTROJE SYSTEMCENTER CONFIGURATION MANAGER | 53 |
| 7.1.1 | Distribuce softwaru | 53 |
| 7.1.2 | Správa softwarových aktualizací..... | 53 |
| 7.1.3 | Možnosti instalace..... | 53 |
| 7.2 | INSTALACE SCCM 2007 NA WINDOWS SERVER 2008 R2 | 54 |
| 7.2.1 | Krok 1: Nastavení SCCM domény..... | 54 |
| 7.2.2 | Krok 2: SCCM prerequisites | 55 |
| 7.2.3 | Krok 3: Instalace SQL Server..... | 59 |
| 7.2.4 | Krok 4: Připravte Active Directory pro SCCM..... | 60 |
| 7.2.5 | Krok 5: Instalace SCCM | 64 |
| 7.3 | SCCM 2007 PRVNÍ NASTAVENÍ A KONFIGURACE | 67 |
| 7.3.1 | Krok 1: SCCM nastavení Boundary..... | 67 |
| 7.3.2 | Krok 2: SCCM konfigurace a nasazení agenta | 68 |
| 7.3.3 | Krok 4: SCCM nastavení Discovery methods | 70 |
| 7.3.4 | Krok 5: SCCM nastavení instalace klientů | 72 |
| 7.3.5 | Krok 6: SCCM Instalace klienta | 74 |
| | ZÁVĚR | 77 |
| | ZÁVĚR V ANGLIČTINĚ | 78 |

| | |
|---|-----------|
| SEZNAM POUŽITÉ LITERATURY..... | 79 |
| SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK | 81 |
| SEZNAM OBRÁZKŮ | 82 |
| SEZNAM TABULEK..... | 85 |
| SEZNAM PŘÍLOH..... | 86 |

ÚVOD

Instalace nového softwaru může být pro uživatele noční můrou a není těžké pochopit proč. Uživatel odevzdává kontrolu nad částí systému nebo nad systémem celým instalační aplikaci, která může vyžadovat oprávnění správce. Uživatel možná zná produkt, který instaluje, případně má k produktu uživatelskou dokumentaci, ale zřídka má informace o tom, co se systémem provádí instalátor samotný, přitom instalační aplikace může provádět zásah do jádra systému, instalací ovladačů a služeb, změnu systémových registrů, změnu osobního nastavení uživatele bez jeho souhlasu, a tohle může vést k nefunkčnosti jiných aplikací, které již dříve fungovaly.

Ve své práci se zabývám technologií pro nasazení aplikací zaměřenou na technologii Windows Installer. Zpočátku uvádím stručný úvod do problematiky vývoje a nasazení aplikací obecně a následně se věnuji tomu, jak se dají tyto problémy řešit právě pomocí technologie Windows Installer. V závěru se věnuji možnostem nasazení aplikace pomocí nástroje System Center Configuration Manager.

I. TEORETICKÁ ČÁST

1 PROBLEMATIKA TVORBY INSTALAČNÍCH BALÍČKŮ

Máme-li naprogramovanou aplikaci a výsledná aplikace bezproblémově funguje, stojíme před úkolem umožnit uživatelům co nejkomfortnější pořízení aplikace a její instalaci. Pokud je naše aplikace tvořena jediným spustitelným souborem například *.exe, nebylo by teoreticky nutné lámat si hlavu s nějakým instalátorem: stačilo by zkopírovat tento spustitelný soubor na cílové umístění a odtud rovnou spustit aplikaci. Tento způsob je velmi jednoduchý a mnozí uživatelé jej jistě vítají především z toho důvodu, že mají nad aplikací kontrolu a vědí, že jim „nezanese“ systém mnoha megabytů jakýchsi pochybných dat, která budou někde nakopírována a která budou sloužit k neznámému účelu. Na druhou stranu bychom si ale měli uvědomit, že žijeme v době, kdy počítač používají i neznalí uživatelé, kteří ve většině případů neumí spustit aplikaci jinak než z nabídky start nebo z plochy. I z těchto a jiných důvodů je doporučeno vytvářet instalační balíčky pro všechny aplikace, a to i pro ty jednodušší. Instalační balíček v podstatě automaticky vytvoří potřebné položky v nabídce Start, zkopíruje aplikaci včetně všech nutných souborů do požadovaného umístění a zavede příslušné záznamy do registru Windows.[1][2][3]

1.1 5 důvodů, proč by měl program obsahovat instalaci

1.1.1 Jednoduchost a přehlednost.

Průvodce instalací je plně přizpůsobitelný, jednoduchý a přehledný. Je navržený tak, aby seděl každému uživateli. Pohodlně nainstaluje aplikaci, zařadí ji do seznamu nainstalovaných aplikací, podle volby uživatele vytvoří ikonu na ploše, v nabídce start nebo v rychlém spuštění. Samozřejmě také obsahuje stejně rychlou možnost odstranění aplikace.[4]

1.1.2 Četnější používání

Díky tomu, že je program uveden v seznamu nainstalovaných aplikací, v nabídce start a případně i jako ikona na ploše, uživatel dotyčný program nejen nezapomene používat, ale také ho používá častěji. Nejhorší, co může být, je program bez instalace, který si uživatel uloží na plochu Windows a pojmenuje ho „Nová složka“. Další výhodou je, že se program neobjeví na internetu ke stažení v podobě zip formátu a navíc nebude kompletní, ale bude obsahovat jen nejdůležitější část. Podobným způsobem se často chovají vývojáři pracující

na tvorbě portable verze (aplikaci je možné spouštět a přenášet například z flash disku). Tímto je možné dosáhnout, že si každý stáhne originální instalaci, a to z umístění k tomu určenému.[4]

1.1.3 Dostupnost

Není potřeba žádný winzip ani winrar, nemusíte pro nic vytvářet složku, program je dostupný každému.[4]

1.1.4 Zpětná vazba přes web

Po instalaci je u aplikace v nabídce start odkaz na internetové stránky, což je velmi jednoduchá cesta, jak přilákat spokojeného uživatele například pro další software nebo informace.[4]

1.1.5 Windows je navržený pro instalaci aplikací

Všichni, kteří odsuzují instalace, si musí uvědomit, že Microsoft Windows je systém navržený pro instalace. A instalační balíček je způsob, kterým se aplikace do systému instalují. Ten, kdo instalace nepoužívá, se ochuzuje o možnosti systému Windows, které pro nainstalované aplikace nabízí (mám na mysli nabídku start, nainstalované aplikace a podobně).[4]

1.2 Obsah instalačního balíčku

Kromě nezbytného spustitelného souboru *.exe obsahuje následující druhy souborů:

aplikační soubory: programové soubory *.exe,*.dll, soubory nápovědy *.hlp, *.cnt, *.toc a jiné, soubory s databázovými a dalšími údaji – např. *.dbf, *.mdx, *.dbt, *.ndx, *.db, *.mb, *.val a řada dalších

soubory balíků (packages): soubory *.bpl a *.bcp. Pokud aplikace používá běhové balíky (run-time packages), je nutné šířit soubory s těmito balíky společně s aplikací.

připojené moduly: připojené moduly slouží k zajištění podpory právě takových funkcí, které v aplikaci vyžadujeme.

ovládací prvky: soubory *.ocx apod.

další podpůrné aplikace: s aplikací je někdy nutné šířit další podpůrné aplikace, bez nichž by funkčnost aplikace nebyla dostatečná. Příkladem může být třeba u databázových aplikací používajících InterBase utilitu ServerManager, která administruje databáze. Při šíření takových podpůrných aplikací je ovšem nutné konzultovat jejich licenční podmínky.

Podstatné je, že všechny shora uvedené druhy souborů je možné připojit do aplikace pomocí vhodného instalátoru. [1]

Proces tvorby instalačního balíku je podobný všem druhům aplikací, tedy alespoň ve své základní podobě. Pro tvorbu instalačních balíčků existuje řada specializovaných nástrojů. Pomocí těchto nástrojů je možné projít kompletním procesem tvorby instalačního balíku, obvykle bez nutnosti psát jakýkoliv programový kód. Aplikací, které se zaměřují na tvorbu instalačních balíčků, je celá řada a dříve bylo nutné použít takový, který měl certifikaci firmy Borland. Dnes již není tato certifikace nutná a stačí, když je tento nástroj postaven na technologii Microsoft Windows Installer.[1]

2 SEZNÁMENÍ S TECHNOLOGIÍ WINDOWS INSTALLER

Instalační služba systému Windows je součástí operačního systému Windows a slouží ke zjednodušení procesu instalace aplikací.

Instalační služba systému Windows spravuje instalaci a odebrání aplikací pomocí sady centrálně definovaných pravidel pro instalaci používaných v průběhu instalace. Kromě toho lze tuto službu použít také k úpravě, opravě nebo odebrání existující aplikace. Technologie Instalační služby systému Windows je tvořena Instalační službou systému Windows pro operační systémy Windows a souborem balíčku (MSI), který obsahuje informace o instalaci a nastavení dané aplikace.

Instalační služba systému Windows představuje kromě jiného rozšiřitelný systém správy softwaru. Spravuje instalaci, přidání a odstranění softwarových součástí, sleduje odolnost souborů proti chybám a spravuje základní zotavení po zhroucení systému metodou vrácení provedených změn. Instalační služba systému Windows podporuje instalaci a spuštění softwaru z více zdrojů a mohou ji upravovat vývojáři, kteří chtějí instalovat vlastní aplikace.

Instalační služba systému Windows vznikla jako odpověď na nutnost otevřeného a standardizovaného přístupu k instalaci aplikací na platformě Windows. Instalační služba systému Windows byla původně vytvořena pro Microsoft Office a jeho první verze 1.0 byla použita právě pro Office 2000, a to v červnu 1999. Krátce poté se stala Instalační služba součástí operačního systému Windows 2000, tato verze nese označení 1.1. Současně se zabudováním Instalační služby do Windows 2000 vznikla i redistribuovatelná verze určená pro starší verze operačního systému Windows 95 a 98. Od té doby vznikají vždy nové verze s novým operačním systémem Windows viz Tab. 1 a Tab. 2.[5][6]

Tab. 1. Historie verzí 1/2

| Vydaná verze | Verze | Poznámka |
|------------------------|---------------|--|
| Windows Installer 1.0 | 1.00.5104.0 | Vydáno s Office 2000 a jako samostatný balíček k distribuci. |
| Windows Installer 1.1 | 1.10.1029.0 | Vydáno s Windows 2000. |
| | 1.10.1029.1 | Vydáno jako samostatný balíček k distribuci. |
| Windows Installer 1.11 | 1.11.1314.0 | Vydáno s Windows 2000 Service Pack 1. |
| | 1.11.2405.0 | Vydáno s Windows 2000 Service Pack 2. |
| Windows Installer 1.2 | 1.20.1410.0 | Vydáno s Windows Millennium Edition. |
| | 1.20.1827.1 | Vydáno jako samostatný balíček k distribuci. |
| Windows Installer 2.0 | 2.0.2600.0 | Vydáno s Windows XP. |
| | 2.0.2600.1 | Vydáno s Windows 2000 Service Pack 3. |
| | 2.0.2600.1183 | Vydáno s Windows 2000 Service Pack 4. |
| | 2.0.2600.2 | Vydáno jako samostatný balíček k distribuci. |
| | 2.0.2600.1106 | Vydáno s Windows XP Service Pack 1. |
| | 2.0.3790.0 | Vydáno s Windows Server 2003. |
| Windows Installer 3.0 | 3.0.3790.2180 | Vydáno s Windows XP Service Pack 2 a jako samostatný balíček k distribuci. |
| Windows Installer 3.1 | 3.1.4000.1823 | Vydáno jako samostatný balíček k distribuci. Tato verze má stejnou funkčnost jako verze 3.1.4000.2435. |
| | 3.1.4000.1830 | Vydáno s Windows Server 2003 Service Pack 1 a Windows XP Professional x64 Edition. |
| | 3.1.4000.3959 | Vydáno s Windows Server 2003 Service Pack 2. |
| | 3.1.4000.2435 | Vydáno s Windows XP Service Pack 3. |
| | 3.1.4001.5512 | Vydáno s Windows XP Service Pack 3 aktualizace. |

Tab. 2. Historie verzí 2/2

| Vydaná verze | Verze | Poznámka |
|-----------------------|----------------|--|
| Windows Installer 4.0 | 4.0.6000.16386 | Vydáno s Windows Vista. |
| | 4.0.6001.18000 | Vydáno s Windows Vista Service Pack 1 a Windows Server 2008. |
| Windows Installer 4.5 | 4.5.6002.18005 | Vydáno s Windows Vista Service Pack 2 a Windows Server 2008 Service Pack 2. |
| | 4.5.6000.20817 | Vydáno jako samostatný balíček k distribuci pro Windows Vista. |
| | 4.5.6001.22162 | Vydáno jako samostatný balíček k distribuci pro Windows Server 2008 a Windows Vista Service Pack 1. |
| | 4.5.6001.22159 | Vydáno jako samostatný balíček k distribuci pro Windows XP Service Pack 2+, a Windows Server 2003 Service Pack 1+. |
| Windows Installer 5.0 | 5.0.7600.16385 | Vydáno s Windows Server 2008 R2 a Windows 7. |

2.1 Windows Installer - otevřená architektura

Dříve než vznikla Instalační služba Windows Installer, se používaly a v korporátní sféře stále používají různé nástroje třetích stran. Mnohé z nich jsou sice velice dobře propracované, ale mají jeden obrovský problém, z pohledu uživatele se totiž chovají jako "Black box" (černá skříňka). Jinak řečeno uživatel neví, co se uvnitř děje. Toto sice nemusí být problém, pokud se jedná o nasazení pro jednoho uživatele na jeden počítač, ale z pohledu systémového administrátora, který je zodpovědný za nasazení aplikace na desítkách či stovkách strojů v kontrolovaném či zabezpečeném prostředí, se jedná o bezpečnostní riziko, které nesmí podstoupit. Dalším podobným problémem může být pro správce požadavek na určitou kontrolu nad nastavením, konfiguraci jednotlivých uživatelů a skupin.

Instalační služba systému Windows tyto problémy řeší pomocí otevřené architektury svých instalačních balíčků, které jsou navrženy jako relační databáze, a její schéma je plně zdokumentované a veřejně přístupné. Kromě toho poskytuje Instalační služba v rámci Windows Installer SDK nástroje, které umožňují, aby správci systému mohli upravovat i takové balíčky, které byly napsány někým jiným.

Windows installer je standardem. Je jedno, jaký se použije vývojový nástroj k vytvoření instalačního balíčku (InstallShield Express, InstallShield Developer, Wise for Windows Installer, nebo Windows Installer SDK), výsledkem je vždy standardizovaná databáze, kterou Instalační služba použije k instalaci. Přestože instalační pokyny uvedené v jednotlivých databázích budou různé, tak sled jednotlivých kroků používaných v Instalační službě systému Windows, které jsou určeny pro zpracování databáze, budou pro všechny aplikace stejné. Tím je zajištěna vysoká úroveň konzistence a spolehlivost pro všechny instalace softwaru bez ohledu na jejich dodavatele.[8][9]

2.2 Správa sdílených zdrojů

Jedním z největších problémů, které vedly k tvorbě Instalační služby Windows, byla možnost, že sdílený prostředek, jako například DLL knihovna, bylo možné přepsat jinou nekompatibilní nebo starší verzí souboru. Podobný problém mohl nastat v případě, že dvě aplikace sdílely stejnou knihovnu. V případě odstranění jedné aplikace byla daná knihovna odstraněna také. Windows Installer tento problém, nazývaný jako „DLL hell“, řeší tím, že řídí registraci sdílených zdrojů.[8][9]

2.3 Správa integrity počítače

Jedním z dalších rizik spojených s instalací nového softwaru je možnost uvedení počítače do nestabilního stavu v případě, že se instalace nezdaří nebo je zrušena před dokončením. Tento problém může nastat v případě, že se povede nakopírovat jen část souborů nebo registrů, ale jiná část nikoliv. Instalační služba systému Windows řeší tento problém řízením celé instalace, a umožňuje tak funkce typu „rollback“. Tímto způsobem je možné vrátit cílový počítač do stabilního stavu tak, jak byl před spuštěním instalace.[8][9]

2.4 Úpravy instalací

Jak již bylo zmíněno, správci rozsáhlých počítačových systémů často požadují možnost různé konfigurace instalace pro každou cílovou skupinu uživatelů/počítačů v rámci organizace. Instalační služba systému Windows toto umožňuje dvěma způsoby. Jedním z nich je vytvoření instalační aplikace pomocí funkcí typu „features“ (funkce mohou být různé), jejichž výsledkem bude volba, zda se daná část má či nemá nainstalovat. Druhou možností je využít otevřenost databáze MSI a vytvořit tzv. instalační transformaci úpravou

instalačního balíčku tak, aby vyhovovala konfigurace danému prostředí. MST transformace popisuje rozdíl mezi původní databází a databází novou. Vzhledem k tomu, že MSI balíčky jsou určeny k distribuci na konkrétní prostředí, bývá nejčastějším využitím transformace jazyková úprava jednotlivých instalačních balíčků. Například, pokud existuje anglický instalační balíček s transformačním souborem obsahujícím českou lokalizaci, je možné spustit instalaci pomocí příkazu

```
MSIEXEC název_balíčku.msi TRANSFORMS=Český_Transform.mst
```

a instalace tak proběhne v českém jazyce. Toto je nejčastěji využíváno v případě distribuce pomocí Group Policy. Nicméně toto využití nemusí být jediné, kromě transformace jazykových balíčků se využívá transformace konfiguračních souborů určených pro různé operační systémy nebo konkrétní uživatele, jedná se například o soubory *.ini nebo různé části registrů.[8][9]

2.5 Oprávnění instalace

Dalším požadavkem systémových správců je možnost instalace uživatelem, který nemá potřebná oprávnění k instalaci. Takto spuštěná instalace by nemusela být správně dokončena v případě, že instalační balíček vyžaduje ke své činnosti oprávnění správce. Tento problém řeší Instalační systém tím, že umí spustit instalační balíčky s jiným oprávněním, než je oprávnění uživatele, a to i v případě, že přihlášený uživatel tato oprávnění nemá.[8][9]

2.6 Windows Installer databáze

Jádrem instalačního balíčku a zároveň základním stavebním kamenem Instalační služby Windows je databáze MSI. Databáze MSI je typem relační databáze, která využívá datové tabulky pro všechna nastavení určená pro instalační proces. Jako každá relační databáze využívá primární a cizí klíče ke stanovení vztahů a vazeb mezi jednotlivými tabulkami.

Je mnoho aplikací určených pro úpravu a nahlížení do MSI databází. Jedním z nich je i aplikace Orca, která je součástí Windows Installer SDK. Samotná instalace aplikace Orca je již napsána pomocí Windows Installer technologie a samotnou instalaci je možné spustit instalačním souborem ORCA.MSI. Jakmile je Orca nainstalována, je možné ji použít

k otevření libovolného souboru MSI. Vzhledem k tomu, že je Orca editor, je možné v databázi rovnou provádět i změny.[9]

2.7 Tabulky

Informace v MSI databázi jsou umístěny v tabulkách. V MSI 2.0 bylo definováno 89 stálých tabulek. Není sice nutné všechny znát, ale postupem času by bylo dobré se s nimi částečně seznámit. V tabulkách se definují soubory, data registrů a další informace o aplikacích, jako například COM objekty, knihovny atd. Vztah mezi jednotlivými tabulkami je postaven na primárních a cizích klíčích, což zajišťuje v prvním případě jedinečnost dat a v druhém případě vztahy mezi tabulkami.[9]

2.7.1 Tabulka Property

Jednou z prvních tabulek, kterou je důležité znát, je tabulka Property. Tato tabulka obsahuje název produktu, verzi produktu, product code GUID, upgrade code GUID a velké množství dalších informací o databázi.[8]

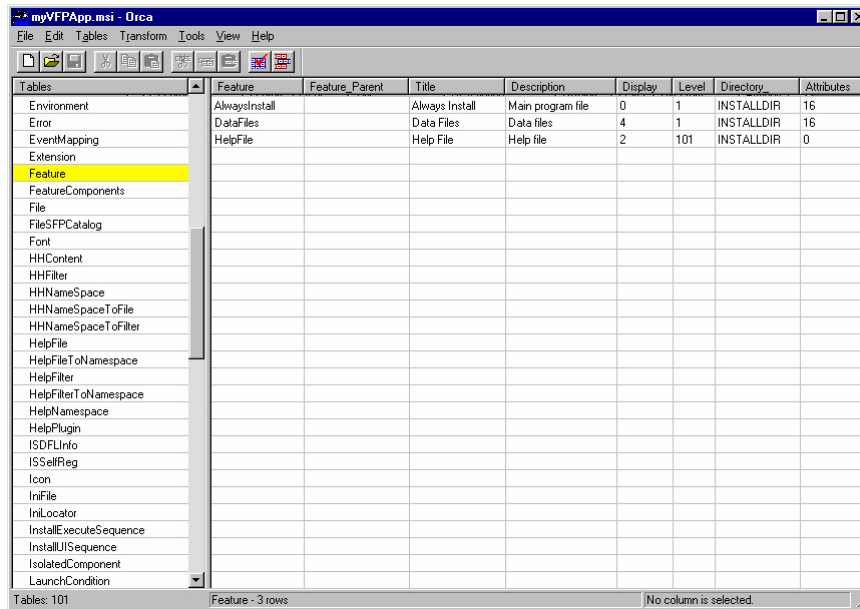
| Property | Value |
|-----------------------|--|
| LaunchCondition | Typical |
| ListBox | 1 |
| ListView | ErrorDialog |
| LockPermissions | SetupError |
| MIME | 100 |
| Media | ISCHECKFORPRODUCTUPDATES |
| ModuleComponents | 1 |
| ModuleDependency | ISVRDOT_PORT_NO |
| ModuleExclusion | 0 |
| ModuleSignature | InstallChoice |
| MoveFile | AR |
| MsiAssembly | LAUNCHPROGRAM |
| MsiAssemblyName | 1 |
| MsiDigitalCertificate | LAUNCHREADME |
| MsiDigitalSignature | 1 |
| MsiFileHash | Manufacturer |
| ODBCAttribute | Information Technology Associates |
| ODBCDataSource | 12345:###-%%%%%% @@@@ |
| ODBCDriver | PIDTemplate |
| ODBCSourceAttribute | ProductCode |
| ODBCTranslator | (48A9C9E5-E6A8-11D7-8387-0050DA64B851) |
| Patch | ProductLanguage |
| PatchPackage | 1033 |
| ProgId | ProductName |
| Property | myVFPApp |
| PublishComponent | ProductVersion |
| RadioButton | 1.0.0 |
| RegLocator | ProgressType0 |
| Registry | install |
| | ProgressType1 |
| | Installing |
| | ProgressType2 |
| | installed |
| | ProgressType3 |
| | installs |
| | RebootYesNo |
| | Yes |
| | ReinstallFileVersion |
| | o |
| | ReinstallModeText |
| | omus |
| | ReinstallRepair |
| | r |
| | SERIALNUMVALSUCCESSRETVAL |
| | 1 |
| | SetupType |
| | Typical |
| | UpgradeCode |
| | (48A9C9E5-E6A8-11D7-8387-0050DA64B851) |
| | _IsMaintenance |
| | Change |
| | _IsSetupTypeMin |
| | Typical |

Obr. 1. Tabulka Property

2.7.2 Tabulka Feature

Jednou ze základních koncepcí pro produkty instalované pomocí služby Windows Installer je, že potřebují být uspořádány do tzv. features. Feature je nejmenší jednotka produktu,

kteřou uživatel může vybrat k nainstalování nebo nenainstalování. Toto uspořádání je ukládáno do tabulky Feature. Obsahuje Položky, jako je například název, popis, cílový instalační adresář a další podmínky nutné k instalaci.[8]



| Feature | Feature_Parent | Title | Description | Display | Level | Directory | Attributes |
|---------------|----------------|----------------|-------------------|---------|-------|------------|------------|
| AlwaysInstall | | Always Install | Main program file | 0 | 1 | INSTALLDIR | 16 |
| DataFiles | | Data Files | Data files | 4 | 1 | INSTALLDIR | 16 |
| HelpFile | | Help File | Help file | 2 | 101 | INSTALLDIR | 0 |

Obr. 2. Tabulka Feature

2.7.3 Tabulka Component

Každá Feature obsahuje jednu nebo více komponent. Komponenty reprezentují logické seskupování zdrojů z pohledu vývojáře. Každá komponenta má své jedinečné GUID, tím je umožněna vždy jen jedna instance dané komponenty. Komponenta je vlastně atomická jednotka celé MSI databáze, obsahuje zdroje (dll, ocx, exe), klíčové cesty (každá komponenta má definovanou klíčovou cestu) a entry points, což je vlastně definice, jak Windows Installer přistupuje k určitým zdrojům, například MS Word je entry points pro koncovku souboru.[8]

| Tables | Component | ComponentId | Directory | Attributes | Condition | KeyPath |
|----------------------|-----------------------------------|----------------------|----------------------|------------|-----------|----------------------|
| ActionText | AIIOtherFiles | {48A9C9E7-E6A8-1... | INSTALLDIR | 2 | | |
| AdminExecuteSequence | AIIOtherFiles1 | {48A9C9E8-E6A8-1... | INSTALLDIR | 2 | | |
| AdminUISequence | AIIOtherFiles2 | {48A9C9F1-E6A8-1... | INSTALLDIR | 2 | | |
| AdvExecuteSequence | Global_Controls_CDMCATDLL.3207... | {3207D1B1-80E5-1... | System.3207D1B0... | 24 | | Global_Controls_C... |
| AdvUISequence | Global_Controls_MSCDMCT20CX.32... | {3207D1B8-80E5-1... | System.3207D1B6... | 24 | | Global_Controls_M... |
| Appld | Global_System_OLEAUT32.8C0C59... | {997FA962-E067-1... | System.8C0C59A0... | 24 | | Global_System_OL... |
| AppSearch | Global_System_OLEPRO32.8C0C59... | {10048713-2C96-1... | System.8C0C59A0... | 24 | | Global_System_OL... |
| BBControl | Global_System_STDOLE.8C0C59A0... | {474F61F1-7342-11... | System.8C0C59A0... | 24 | | Global_System_ST... |
| Billboard | ISX_DEFAULTCOMPONENT | {48A9C9ED-E6A8-... | INSTALLDIR | 2 | | |
| Binary | ISX_DEFAULTCOMPONENT1 | {48A9C9F3-E6A8-... | INSTALLDIR | 2 | | |
| BindImage | ISX_DEFAULTCOMPONENT2 | {48A9C9F9-E6A8-... | DATABASEDIR | 2 | | |
| CCPSearch | Myvfpapp.chm | {48A9C9E9-E6A8-1... | INSTALLDIR | 2 | | Myvfpapp.chm |
| CheckBox | Myvfpapp.exe | {48A9C9EF-E6A8-1... | INSTALLDIR | 2 | | Myvfpapp.exe |
| Class | VFP_HTML_Help8_Support_X86... | {AAC459BA-9F6C-4... | VFP.3643236F_FC... | 8 | | FL_foxhhelp8_exe... |
| ComboBox | _VFP8_RUNTIME_DLL_X86.36... | {1128C875-6403-4... | VFP.3643236F_FC... | 8 | | FL_vfp7_dll_... |
| CompLocator | _VFP8_RUNTIME_RES_CORE_... | {A1B1EBA4-E3EA-... | VFP.3643236F_FC... | 0 | | FL_VFP7RENU_D... |
| Complus | _VFP8_Runtime_X86.3643236F... | {18D7A8A1-5F25-4... | VFP.3643236F_FC... | 8 | | FL_vfp7_dll_... |
| Component | _VFP_GDIPlus_DLL_X86.3643... | {235D886E-2F97-1... | VFP.3643236F_FC... | 8 | | gdplus_dll_2_... |
| Condition | _vc_CRT_X86.3643236F_FC7... | {C988BC60-48EC-1... | SystemFolder.3643... | 0 | | |
| Control | | | | | | |
| ControlCondition | | | | | | |
| ControlEvent | | | | | | |
| CreateFolder | | | | | | |
| CustomAction | | | | | | |
| Dialog | | | | | | |
| Directory | | | | | | |
| DLocator | | | | | | |
| DuplicateFile | | | | | | |
| Environment | | | | | | |

Obr. 3. Tabulka Component

2.7.4 Tabulka File

Poslední důležitou tabulkou je tabulka souborů. Tabulka obsahuje jeden řádek pro každý soubor, který má být nainstalován jako součást instalačního balíčku. Primárním klíčem je hodnota ve sloupci File, jehož hodnota bývá obvykle spojena s názvem souboru. Druhý sloupec Component_ je cizí klíč do tabulky Component. Tato vazba vytváří vztah mezi Componenty a soubory, které se instalují.[8]

| Tables | File | Component | FileName | FileSize | Version | Language | Attributes | Sequence |
|------------------------|---------------------------|-----------------|------------------|----------|-------------|----------|------------|----------|
| Environment | FL_VFP7RENU_DLL_... | _VFP8_RUN... | VFP8RENU... | 1150976 | 8.0.0.2521 | 1033 | 8192 | 10 |
| Error | FL_foxhhelp8_exe... | X... | VFP_HTML... | 57344 | 8.0.0.2518 | 1033 | 8192 | 13 |
| EventMapping | FL_foxhhelp8_dll... | VFP_HTML... | FOXHHE*1... | 6656 | 8.0.0.2518 | 1033 | 8192 | 14 |
| Extension | FL_msvcr70_dll_X86... | _vc_CRT_... | msvcr70.dll... | 303104 | 7.0.9111.0 | 1033 | 8192 | 15 |
| Feature | FL_vfp7_dll_X86.36... | _VFP8_Runti... | vfp8.dll\vfp8... | 4292608 | 8.0.0.2521 | 1033 | 8192 | 9 |
| FeatureComponents | FL_vfp7_dll_X86.36... | _VFP8_RUN... | vfp8.dll\vfp8... | 3768320 | 8.0.0.2521 | 1033 | 8192 | 11 |
| File | Global_Controls_CDMCAT... | Global_Contr... | comcat.dll | 22288 | 4.71.1460.1 | 1033 | 8192 | 3 |
| FileSFPCatalog | Global_Controls_MSCDMC... | Global_Contr... | mscomct2.ocx | 647872 | 6.0.88.4 | 1033 | 8192 | 4 |
| Font | Global_System_OLEAUT3... | Global_Syste... | asyclit.dll | 147728 | 2.40.4275.1 | 1033 | 8192 | 7 |
| HHContent | Global_System_OLEAUT3... | Global_Syste... | oleaut32.dll | 598288 | 2.40.4275.1 | 1033 | 8192 | 8 |
| HHFilter | Global_System_OLEPRO3... | Global_Syste... | olepro32.dll | 164112 | 5.0.4275.1 | 1033 | 8192 | 5 |
| HHNameSpace | Global_System_STDOLE... | Global_Syste... | stdole2.tlb | 17920 | 2.40.4275.1 | 1033 | 8192 | 6 |
| HHNameSpaceToFile | Myvfpapp.chm | Myvfpapp.chm | Myvfpapp.chm | 12540 | | | 8192 | 16 |
| HHNameSpaceToFilter | Myvfpapp.exe | Myvfpapp.exe | Myvfpapp.exe | 459958 | 1.0.0.0 | 1033 | 8192 | 2 |
| HelpFile | Readme.txt | ISX_DEFAULT... | Readme.txt | 37 | | | 8192 | 1 |
| HelpFileToNamespace | gdplus_dll_2_X86.36... | _VFP_GDIPlu... | gdplus.dll\gd... | 1700352 | 5.1.3097.0 | 1033 | 8192 | 12 |
| HelpFilter | myData.DBF2 | ISX_DEFAULT... | myData.DBF | 5106 | | | 8192 | 17 |
| HelpFilterToNamespace | myData.FPT2 | ISX_DEFAULT... | myData.FPT | 24448 | | | 8192 | 18 |
| HelpNamespace | | | | | | | | |
| HelpPlugin | | | | | | | | |
| ISDDLInfo | | | | | | | | |
| ISSellReg | | | | | | | | |
| Icon | | | | | | | | |
| IniFile | | | | | | | | |
| IniLocator | | | | | | | | |
| InstallExecuteSequence | | | | | | | | |
| InstallUISequence | | | | | | | | |
| IsolatedComponent | | | | | | | | |
| LaunchCondition | | | | | | | | |

Obr. 4. Tabulka File

2.7.5 Ostatní tabulky

Další tabulky vždy vycházejí z propojení tabulek File, Feature a Component. Mohou obsahovat podrobnosti k jednotlivým komponentám, jako například jejich registry a jiné detailní informace určené k instalaci dané komponenty.

2.8 GUID

Každý balíček musí obsahovat tzv. GUID (Globally Unique IDentifiers), jedná se o jedinečné referenční číslo. Hodnota GUID je reprezentována jako hexadecimální řetězec o délce 32 znaků a je obvykle uložena jako stovacetiosmibitové celé číslo. Celkový počet unikátních klíčů je tedy 2 na 128, což je tak vysoké číslo, že pravděpodobnost vygenerování sejného čísla dvakrát je zanedbatelná. Každá MSI Databáze musí mít unikátní Package Code, a to platí i pro každou další verzi téhož produktu, k tomuto účelu se právě využívá GUID, ale je na vývojáři, aby zajistil, že každá jeho MSI databáze má jedinečný Package Code GUID. Naštěstí většina Windows Installer nástrojů změni Package Code GUID automaticky pokaždé, když bude programátor vytvářet release, takže to není obecně něco, na co je potřeba pamatovat a provádět tuto operaci ručně. Další dva GUID, které jsou také důležité pro práci se soubory MSI, jsou Product Code a Upgrade Code. Každá hlavní verze produktu by měla mít unikátní Product Code a všechny verze téhož produktu by měly mít společný Upgrade Code. Product Code a Upgrade Code MSI databáze je možné najít v tabulce Property MSI souboru.[7]

2.9 Možnosti služby Windows Installer msiexec

```
msiexec /Option <požadovaný_parametr> [volitelný_parametr] [9]
```

2.9.1 Možnosti instalace

```
</package | /i> <Produkt.msi>
```

Instaluje nebo konfiguruje produkt

```
/a <Produkt.msi>
```

Administrativní instalace - nainstaluje produkt do sítě

```
/j<u|m> <Produkt.msi> [/t <transformační_seznam>] [/g <ID_jazyka>]
```

Publikuje produkt. Přepínače: m - všichni uživatelé,

u - aktuální uživatel

```
</uninstall | /x> <Produkt.msi | kód_produkту>
```

Odinstaluje produkt [9]

2.9.2 Možnosti zobrazení

```
/quiet
```

Tichý režim (bez interakce uživatele)

```
/passive
```

Bezobslužný režim (bude zobrazen průběh)

```
q[n|b|r|f]
```

Úroveň uživatelského rozhraní (UI)

n - Vypnuto

b - Základní UI

r - Zjednodušené UI

f - Úplné UI (výchozí nastavení)

```
/help
```

Nápověda [9]

2.9.3 Možnosti restartování

```
/norestart
```

Po dokončení instalace nerestartovat

```
/promptrestart
```

V případě vyžadování restartování zobrazit dotaz

```
/forcerestart
```

Po dokončení instalace vždy restartovat počítač [9]

2.9.4 Možnosti protokolování

/l[i|w|e|a|r|u|c|m|o|p|v|x|+|!]*] <soubor_protokolu>

i - Stavové zprávy

w - Nekritická varování

e - Všechny chybové zprávy

a - Začátek akcí

r - Záznamy specifické pro akci

u - Uživatelské požadavky

c - Počáteční parametry UI

m - Chyba nedostatku paměti či jiné kritické informace

o - Zprávy o nedostatku místa na disku

p - Vlastnosti terminálu

v - Rozšířený výstup

x - Dodatečné ladicí informace

+ - Připojit k existujícímu souboru protokolu

! - Připsat každý řádek do protokolu

* - Protokolovat všechny informace (kromě možností v a x)

/log <soubor_protokolu>

Stejně jako /l* <soubor_protokolu> [9]

2.9.5 Možnosti aktualizace

/update <aktualizace1.msp>[;aktualizace2.msp]

Nainstaluje aktualizace

/uninstall <GUID_opravy>[;aktualizace2.msp] /package <Produkt.msi | kód_produkту>

Odinstaluje aktualizace produktu [9]

2.9.6 Možnosti opravy

/f[p|e|c|m|s|o|d|a|u|v] <Produkt.msi | kód_produkту>

Opraví produkt

p - pouze v případě chybějícího souboru

o - v případě chybějícího souboru nebo nainstalované starší

verze souboru (výchozí nastavení)

e - v případě chybějícího souboru nebo starší či stejné verze souboru

d - v případě chybějícího souboru či rozdílné verze souboru

c - v případě chybějícího souboru či rozdílného kontrolního součtu

a - reinstalace všech souborů

u - všechny požadované uživatelské záznamy v registru

m - všechny požadované záznamy v registru o počítači

s - všechny stávající odkazy

v - spuštění ze zdroje a uložení balíčku do mezipaměti [9]

2.10 Automatizace instalace

Aby bylo možné plně automatizace při procesu zavádění, musejí instalované balíčky podporovat bezobslužnou instalaci viz Tab. 3. Řada instalačních programů podporuje přepínače /s nebo /q příkazového řádku zajišťující tiché instalace, pro jiné to ovšem neplatí. Podporu bezobslužné instalace balíčku je možné často zjistit zadáním setup /? na příkazový řádek, kde „setup“ představuje název souboru instalačního programu. Nenabídne-li instalátor žádné informace, pak je potřeba vědět, jaký produkt byl použit k vytvoření balíčku. To lze obvykle zjistit spuštěním instalačního programu a hledáním nějakých logů nebo zobrazením vlastností tohoto souboru. [9]

Tab. 3. Možnosti bezobslužné instalace

| Typ balíčku | Příkaz bezobslužné instalace |
|--------------------------------------|--|
| Windows Installer | msiexec.exe /i balíček.msi /qn ALLUSERS=2 |
| InstallShield Windows Installer | setup.exe /s /v"/qn" Volitelně můžete extrahovat databázi služby Windows Installer z komprimovaného souboru a použít k instalaci příkaz msiexec.exe /i setup.msi ISSETUPDRIVEN=1 /qn. |
| Starší InstallShield Package pro web | setup.exe /a /s /sms Soubor Setup.iss nezbytný k tichému běhu instalace vytvoříte zadáním setup.exe /a /r. Soubor Setup.iss se vytvoří podle vašich reakcí na dialogová okna instalátoru a vy jej pak zkopírujete ze složky %SYSTEMROOT% do složky obsahující balíček. |
| Wise Installation System | setup.exe /s |

2.11 Updaty, upgrady a patche

Pravděpodobně žádná jiná vlastnost Instalační služby Windows nevyvolala u vývojářů větší rozruch než funkce jednoduché aktualizace již existujících produktů. Windows Installer sice přináší mnoho skvělých funkcí, ale právě proto se občas na některých místech projevuje svojí složitostí a právě aktualizace stávajících balíčků je jeden z typických příkladů potvrzujících toto pravidlo. [8]

Aby bylo možné vysvětlit podstatu a funkci aktualizace instalačních balíčků, je potřeba zavést pár důležitých pojmů. V rámci Windows Installer se používají tři základní pojmy update, upgrade, patch a vzhledem k jejich významové blízkosti jej není ani vhodné překládat do češtiny a ani se to tak raději nikde nedělá. Každý z těchto pojmů má svůj specifický význam a vztahuje se právě k procesu aktualizace. Pokud se používá

v aplikacích slovíčko „update“, je to právě tehdy, pokud se mluví obecně o aktualizaci. Například pokud je potřeba vytvořit upgrade nebo patch (obecně jakoukoliv aktualizaci), je potřeba vyhledat sekci s názvem „update“. Dalo by se říci, že update má v základě dvě volby, a to právě „upgrade“ a „patch“. První z nich „upgrade“ by se dal dále rozdělit do tří podskupin „small upgrade“, „minor upgrade“ a „major upgrade“. Databáze MSI se identifikuje v systému pomocí GUID a svou verzí. Rozdíl mezi těmito typy aktualizací se dá vysvětlit právě pomocí toho, které z těchto hodnot se liší právě mezi původní verzí a aktualizací. Následující tabulka zachycuje právě tyto rozdíly.[8]

Tab. 4. Small, minor, major upgrades

| Typ instalace | Package Code | Product Version | Product Code | Upgrade Code |
|---------------|--------------|-----------------|--------------|--------------|
| Small upgrade | změna | původní | původní | původní |
| Minor upgrade | změna | změna | původní | původní |
| Major upgrade | změna | změna | změna | původní |

Jak je z tabulky patrné, tak „package kód“, který identifikuje balíček, se ve všech typech této aktualizace mění. Je to z toho důvodu, že je nutné zachovat pravidlo, že každý balíček musí mít jedinečné GUID. V případě MSI se pro tyto účely používá „package code“. „Small upgrade“ a „minor upgrade“ se používá pro aktualizace, které nejsou natolik významné, aby odůvodňovaly změnu v „product code“. „Small upgrade“ změní „package code“, ale nemění žádnou z ostatních tří vlastností. „Minor upgrade“ oproti tomu už mění navíc „product version“. Rozdíl mezi „small“ a „minor“ je tedy, že po instalaci „small“ nelze rozlišit mezi původní a novou verzí, protože verze produktu se nezměnila, zatímco po instalaci „minor“ lze zjistit rozdíl, a z toho důvodu je i ve většině případů vhodnější právě „minor“ typ aktualizace. [9]

„Major upgrade“ se používá při nasazení zásadních změn v aplikaci, navíc by mělo být pravidlem, že se současně změní první číslice ve verzi produktu. Například MYApp verze 2.0 nahradí verzi 1.0.[8]

Další rozdíl mezi menšími typy „small“, „minor“ a větším „major“ je taky v chování. Při aktualizaci menší typy provádí tzv. přeinstalaci starší instalace a naopak větší „major“ provádí kompletní odstranění původní instalace a teprve následnou instalaci. Ve všech

třech případech se tak provede instalace nová. Dalším rozdílem je, že u malých instalací je nutné, aby uživatel ručně zvolil typ reinstalace, pokud již produkt existuje. V těchto případech se většinou používá příkaz `msiexec /i MYApp.msi REINSTALLMODE=vomus REINSTALL=ALL` [8]

2.11.1 Patche

Dalším způsobem, jak aktualizovat již nainstalovanou aplikaci, jsou tzv. „patche“. Patch na rozdíl od předchozích typů nelze instalovat tam, kde aplikace ještě není nainstalována a navíc umožňuje aktualizaci pouze mezi dvěma po sobě jdoucími verzemi. Aby bylo možné patch vytvořit, jsou potřeba dva po sobě jdoucí balíčky, ze kterých se v podstatě pomocí funkce `diff` vytvoří rozdílový balíček, který obsahuje pouze změny. Takto vznikne opravný balíček MSP, který lze použít právě jako „patch“. Výhodou je pak i velikost samotných patchů.[8]

2.12 Merge module

Merge moduly poskytují možnost sdílet komponenty mezi různými instalačními balíčky. Merge module je soubor MSM, který obsahuje instalaci nějaké komponenty, ale na rozdíl od MSI souboru ho nelze rovnou instalovat, ale je ho možné přidat do jiného MSI souboru. Vytvářet merge moduly je možné úplně stejně a pomocí stejných nástrojů jako při vývoji standardní databáze MSI. Samotné sloučení pak probíhá přidáním MSM modulu do MSI balíčku tak, že je přidán do sekce Features. [9]

2.13 Custom action - Vlastní akce

Instalační služba poskytuje mnoho základních vestavěných funkcí, pokud je však potřeba provádět akce, které ve standardním seznamu nejsou, je možné použít tzv. Custom Action. Takové akce mohou být například:

- Spouštění souboru, který je již na počítači uživatele nainstalován nebo je instalován spolu s aplikací, během instalace.
- Volání speciálních funkcí, které jsou definovány v příložené DLL knihovně.
- Spouštění samostatných skriptů napsaných v Microsoft JScript.

Zde je potřeba jen upozornit, že takto nelze jako součást instalace spouštět instalování jiných MSI.

Jako příklad Custom akce je možné tedy nastavit spouštění exe souboru. Akce se spustí během instalace ve fázi po nakopírování aplikačních souborů. V praxi takto můžeme například provést nastavení připojením databáze k instalované aplikaci, pozměnit aplikační config soubor a podobně. Protože počítáme, že náš exe soubor bude sloužit pouze pro provádění akcí během instalace, je možné ho také zahrnout přímo jako Binary zdroj instalace, odkud se spustí a nebude se tedy vůbec kopírovat do aplikačního adresáře (rozbalí se do temp adresáře, odkud se spustí a následně je smazán). [8]

2.14 Bootstrapper

V minulé části jsem naznačil, že nelze jako vlastní akci volat jiný MSI balíček. Pokud bych se měl držet pouze technologie MSI, tak moc možností není, v podstatě je pouze možné vytvořit tzv. prerequisite, která před spuštěním samotné instalace zjistí, zda daná aplikace je již nainstalována a pokud není, vyvolá dialog s touto informací a nedovolí dokončit instalaci. [3]

Jak tedy vyřešit například problém, kdy na počítači není z nějakých důvodů nainstalovaný Windows Installer, .NET Framework 4.0, komponenta ReportViewer nebo třeba samotný MS SQL Server apod. Tyto komponenty mají většinou vlastní instalační balíčky (jiné MSI) a ty nelze zahrnout přímo jako součást MSI balíčku. Jejich instalaci je potřeba spouštět již před instalací vlastní aplikace pomocí programu Bootstrapper, který je potřeba vytvořit. Celá instalace se tedy bude skládat ze souboru setup.exe (program Bootstrapper), MSI nebo exe souborů, potřebných komponent a samotné MSI aplikace. [3]

Existuje více možností, jak takový program vytvořit. Je možné použít některý z placených nástrojů, například Installshield v plné verzi. Nebo je potřeba sáhnout k Visual Studiu. [3]

Ve Visual Studiu na vytvoření Bootstrapper instalátoru je možné využít nástroj ClickOnce deployment feature. Jedná se o stejný nástroj, který se používá v instalačních projektech Visual Studio Installer. Je zde ale jeden problém, komponenty se musí do projektu zadávat ručně. (V projektu Visual Studio Installer se komponenty vybírají ve vlastnostech projektu tlačítkem Prerequisites). Nevýhodou tvorby bootstrapperu ve Visual Studiu je pak fakt, že tímto způsobem lze nadefinovat pouze komponenty, které se nacházejí v adresáři Packages,

tj. pouze ty, které jsou dodávány pro Visual Studio. Pokud je potřeba instalovat jinou nebo třeba i vlastní komponentu, použít takto generovaný bootstrapper není doporučeno. Naopak, pokud potřebujete klasickou instalaci .NET aplikace a k ní např. kombinaci Installer 3.1, .NET Framework 4.0 a ještě třeba ReportViewer, nebo Installer 4.5, .NET Framework 4.0 a MS SQL Server Express, je toto řešení ideální. [3]

A jak to pak vypadá v praxi? Program Setup.exe při spuštění nejprve zkontroluje, jestli definované komponenty nejsou již nainstalované, pokud ano, spustí se rovnou MSI instalace naší aplikace. Pokud některá komponenta není nainstalovaná, zobrazí se dialog na potvrzení její instalace a její licence. Poté se postupně spouští instalace chybějících komponent v režimu bez zobrazení UI instalace (Quiet mode - instalace na pozadí bez zásahu uživatele), zobrazen je průběh přímo v okně programu Setup.exe. [3]

Každopádně, pokud je alespoň nějaká možnost získat od výrobce komponenty MSM merge modul, tak je lepší využít tuto variantu. Vyhnete se tak například problémům s instalací v Quiet modu. [3]

II. PRAKTICKÁ ČÁST

3 INTEGRITA VSTUPNÍCH SOUBORŮ

První fází distribuce aplikace k zákazníkovi je většinou samotná distribuce binárních souborů aplikací od vývojáře k testerům nebo tvůrcům instalačních balíčků.

V prostředí velkých firem tato data pocházejí z různých míst a přes různá zařízení. Proto je nutné zajistit integritu vstupních souborů. Toto je možné zajistit využitím malé aplikace IntegrityCheck (zdrojové kódy a aplikace jsou v příloze č. 1), která zajistí, aby byly předány požadované soubory v neporušené podobě.

Program funguje na principu tzv. hašovací funkce. Jedná se o matematickou funkci určenou pro převod vstupních dat do sice relativně velkého čísla (například 256 bitů), přesto může být jeho délka v porovnání s velikostí vstupních dat relativně malá. Výstupem takovéto funkce je hash. Hash je v podstatě něco jako otisk vstupních dat o fixní délce.[10]

Mezi hlavní vlastnosti takovéto funkce patří:

1. jakékoliv množství vstupních dat poskytuje stejně dlouhý výstup (otisk),
2. malou změnou vstupních dat je dosažena velká změna na výstupu (tj. výsledný otisk se od původního zásadně na první pohled liší),
3. z hashe je prakticky nemožné rekonstruovat původní text zprávy,
4. pokud je zvolen správný algoritmus (u starších typů algoritmů je možné kolizi nalézt) je vysoce nepravděpodobné, že dvěma různými zprávami odpovídá stejný hash, jinými slovy pomocí hashe lze v praxi identifikovat právě jednu zprávu (ověřit její integritu).

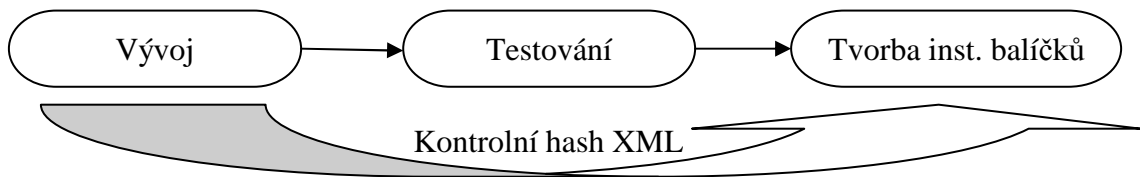
Hašování v základní variantě dovoluje testovat vstupní data na shodu, tedy rovnost.

Program má dvě hlavní funkce:

1. Vygenerovat z adresáře xml dokument s otisky jednotlivých souborů. Takovýto dokument pak bude sloužit pro následnou kontrolu integrity.
2. Porovnat aktuální obsah adresáře s dříve vytvořeným xml dokumentem obsahujícím hashe souborů.

Výsledkem využití takovéto aplikace je tedy jistota, že soubory vstupující do instalačních balíčků nejsou poškozeny a že jsou dodány v požadovaných verzích.

Typickým využitím může být například následující postup. Vývojáři připraví soubor aplikací, ke kterému vygenerují kontrolní xml, to je následně distribuováno k tvůrci instalačního balíčku, ale aplikace je mezi tím testována testery a teprve následně jsou data z aplikací předána tvůrci instalačního balíčku. Ten pak může jednoduše ověřit porovnáním hashů, zda je aplikace kompletní a zda mu byly dodány požadované soubory. [11][12][13][14]



Obr. 5. Diagram postupu vývoje

3.1 Ukázka aplikace

Aplikaci lze spustit s dvěma základními parametry:

1. Vygenerování xml s klíči pro pozdější kontrolu integrity adresáře.

-generateKey [DirectoryPath]

Vygenerované xml může vypadat například takto:

```

<?xml version="1.0" encoding="utf-8"?>
<IntegrityCheckKey>
  <directory>
    <fileHash path="Interop.Ac.dll"
version="1.0.0.0">BlK+3zQdSss8sfzIDYiADg==</fileHash>
    <fileHash path="Interop.CE.dll"
version="1.0.0.0">mjbwauVLi8BpROJN0IBb9g==</fileHash>
    <fileHash path="Interop.CERTEN.dll"
version="1.0.0.0">AZcOXjf4DU04dNvNLe/8tw==</fileHash>
    <fileHash path="LotA.exe"
version="1.3.1148.5">CgHS2owCktLpO+xznU2ZGg==</fileHash>
    <fileHash path="LotA.exe.config">GHjDz56Y4cMe4u5YjDCNkg==</fileHash>
    <directory path="system32">
      <fileHash path="system32\msvc80.dll"
version="8.00.50727.6195">HRCe0NZgZU6n/xV0VYAxXA==</fileHash>
    </directory>
  </directory>
</IntegrityCheckKey>
  
```

Obr. 6. Vygenerované xml

2. Kontrolu integrity adresáře s použitím již vygenerovaného klíče.

-checkIntegrity [DirectoryPath] [IntegrityKeyPath]

Výsledkem kontroly pak může být například následující hlášení:

```
V adresáři došlo ke změně.  
Pro pozdější analýzu konkrétních změn porovnejte obsahy souborů  
C:\IntegrityCheckKey.xml a nově vygenerovaného  
ActualIntegrityKeyDocument.xml  
Seznam změn  
Změna v souboru LotA.exe.config.
```

Obr. 7. Kontrola integrity

4 UKÁZKA TVORBY INSTALAČNÍHO BALÍČKU

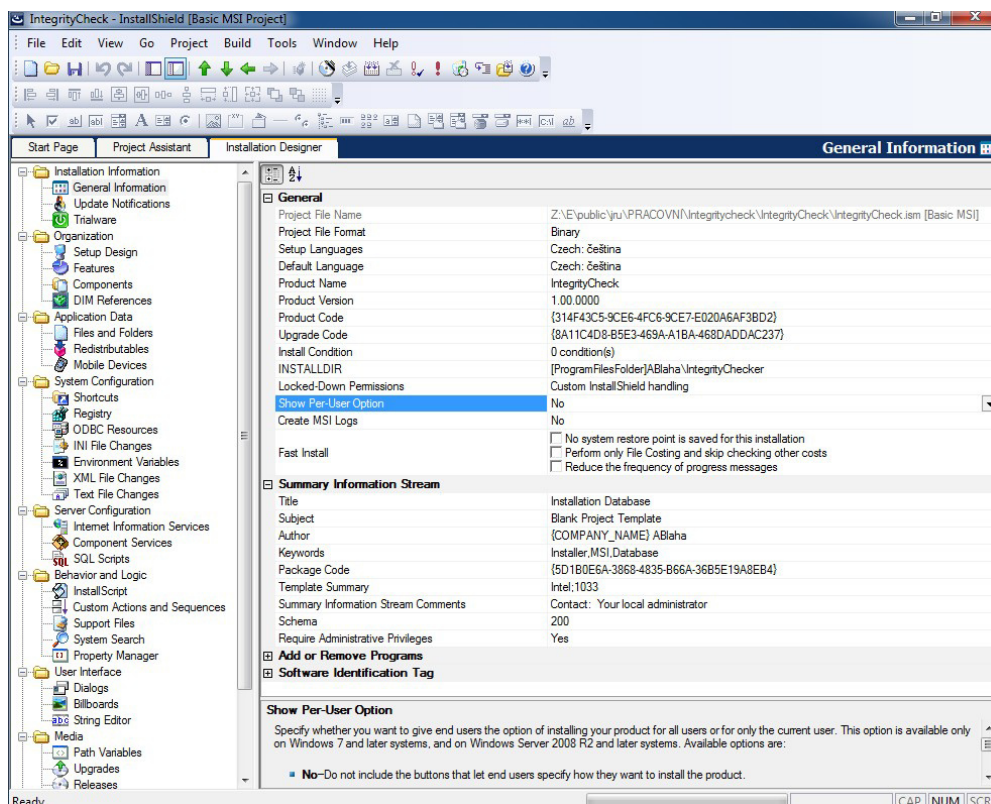
V samotném Visual Studiu je v základu možné vytvářet instalační projekty buď pomocí Visual Studio Installer, nebo od verze 2010 je také možné doinstalovat InstallShield Limited Edition. Oba tyto nástroje umožňují instalaci nadefinovat pomocí různých grafických průvodců. Je možné však použít i jiné nástroje, které však mohou mít různá omezení, například WIX Open Source projekt od Microsoftu, který však nemá grafické rozhraní a je založen na technologii xml. Pokud však budete vytvářet instalační balíčky ve velkém množství, nejspíš si vyberete Installshield v nějaké z nabízených plných verzí.

4.1 Installshield

Installshield je nejkompexnější nástroj na tvorbu instalačních balíčků pro platformu Microsoft Windows. Je dodáván v různých verzích, od verze zdarma, přes verze, které jsou například součástí Visual Studia, po verze placené. Umožňuje tvorbu instalačních balíčků založenou především na technologii Windows Installer, ale umožňuje i jiné technologie (převážně v placených verzích).

4.1.1 Installation Information

V sekci Installation Information viz Obr. 8 se zadávají základní údaje potřebné pro instalační proces, jako informace o produktu (verzích, autorovi, podpoře apod.) a informace o stavebních částech používaných pro jednotlivé druhy instalace (Minimální, Typická, Uživatelská). [2][9]



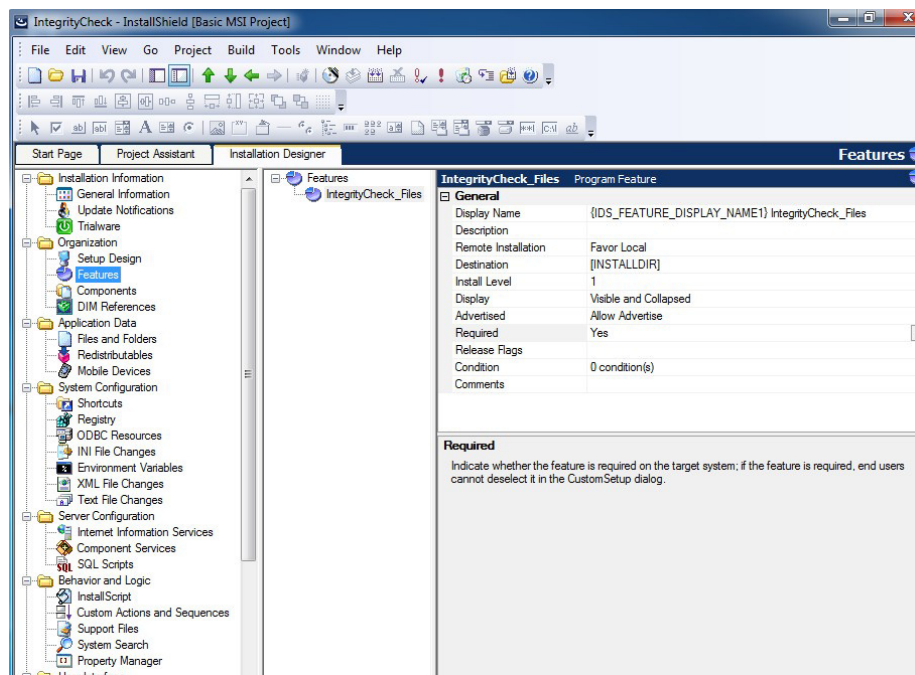
Obr. 8. Installation Information

4.1.2 General Information

V této části je možné nastavit všechny požadované obecné informace o projektu, jeho názvu, autorovi, klíčových slovech, verzi, telefonních číslech apod. Zde je také možné zvolit umístění souboru readme. Některé informace, které se zadávají v tomto kroku, se zobrazují v dialogu Support Information Dialog neboli Informace o odborné pomoci v panelu Přidat nebo odebrat programy ve Windows. Jiné mohou sloužit pouze pro interní účely a nebudou zobrazeny koncovým uživatelům programu. [2][9]

4.1.3 Organization

Tento krok slouží k nastavení různých vlastností instalace. V základu lze rozdělit instalaci na dva typy: automatická instalace a uživatelská instalace, při níž je možné vybírat jednotlivé položky, které se mají, respektive nemají, instalovat. Implicitně je vytvořena vždy minimálně jedna položka Features viz Obr. 9, která se instaluje, v některých verzích je označována jako Always Install Feature, nebo obsahuje parametr Required. Tato položka by měla být alespoň u jedné Feature, protože slouží k zajištění alespoň jedné funkční konfigurace, která povede k úspěšné instalaci a používání aplikace.



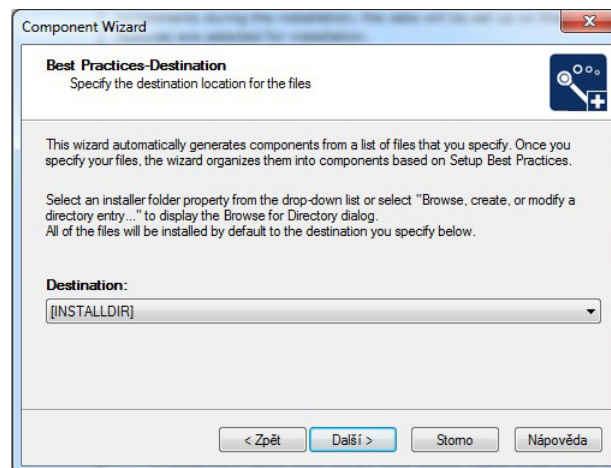
Obr. 9. Features

Klíčové pojmy v tomto kroku jsou vlastnosti (features) a podvlastnosti (subfeatures). Vlastnosti jsou stavební bloky aplikace (z uživatelského úhlu pohledu). Obsahují tedy určitou (uzavřenou) množinu funkcí. Jako příklad vlastnosti si lze představit třeba nápovědu, kterou uživatel buď chce, nebo nechce instalovat. Plyne z toho, že celá aplikace může být „rozdělena“ do jakýchsi bloků (vlastností), z nichž si uživatel při instalaci zvolí ty, které hodlá využívat. Přidání takovéto vlastnosti se pak provede jednoduše kliknutím pravým tlačítkem a volbou New Feature. „Podvlastnosti“ následně slouží jako členění vlastností. Uživatel si tak může dále volit podrobnější funkční bloky, které chce instalovat. Přidání nové podvlastnosti se provádí analogicky jako u vlastností.

V případě, že již je vytvořena struktura Feature, přichází na řadu přidání jednotlivých komponent. Komponenta je o řád menší stavební jednotka než vlastnost a každá vlastnost může obsahovat neomezené množství komponent. Každá komponenta navíc odkazuje na již konkrétní soubory, registry, odkazy a další specifická nastavení. Komponentu lze vytvořit buď ručně, podobným způsobem jako vlastnost, nebo pomocí tzv. Wizardu, což může být lepší volbou, protože umožňuje přímou a hromadnou volbu souborů. [2][9]

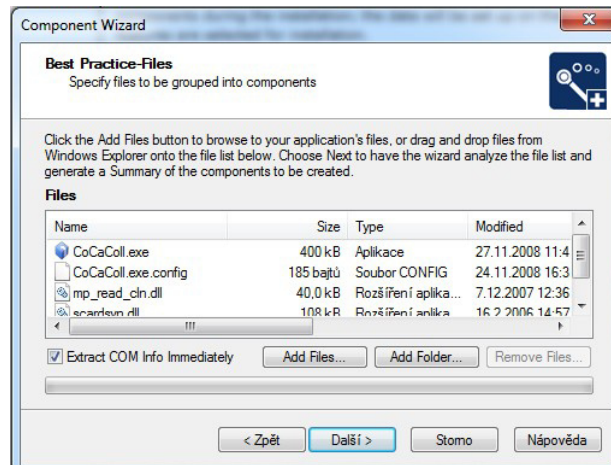
4.1.4 Ukázka Wizardu Component

Na Obr. 10 je vidět volba umístění instalovaných souborů, v tomto případě je zastoupená proměnnou [INSTALLDIR], která zároveň určuje defaultní umístění souborů.



Obr. 10. Wizard Destination

Na Obr. 11 je následně vidět výběr souborů určených pro instalaci i s volbou vyčtení COM objektů (pokud existují) z daných souborů.



Obr. 11. Wizard files

Tímto způsobem se vytvoří pro každý soubor jedna komponenta, se kterou je pak možné následně pracovat a přiřadit ji do příslušné Feature, tedy vlastnosti.

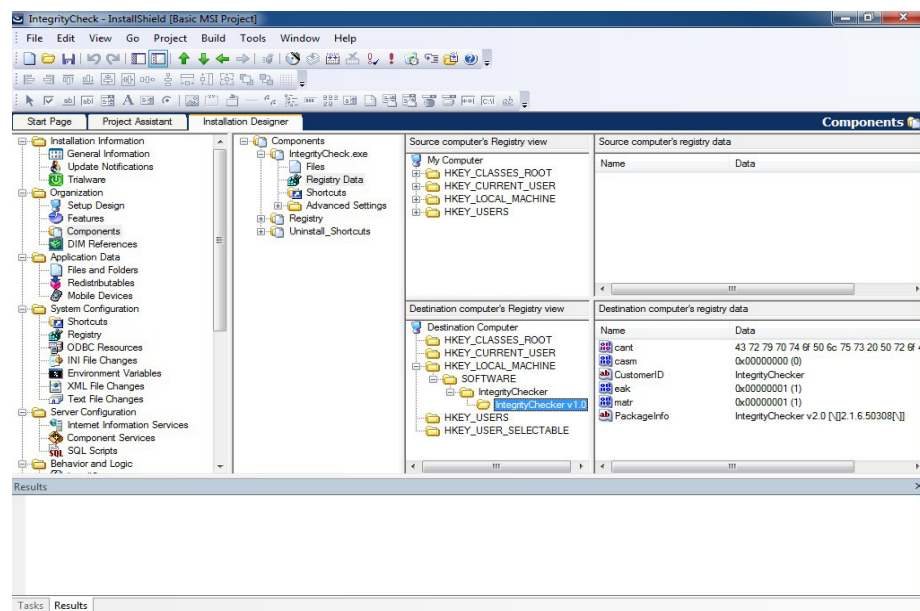
4.1.5 Vytváření registrů

Když už máme v balíčku všechny soubory a k nim vytvořené komponenty, je dalším krokem vložení, respektive vytvoření registrů. V základu jsou dvě možnosti, jak registry vložit do balíčku, a to buď ručním založením, nebo pomocí importní funkce.

Nejjednodušší způsob, jak stanovit požadované změny v registrech na cílovém počítači, je přetáhnout odpovídající klíče z registrů na zdrojovém počítači pomocí funkcí export import. Pokud tedy máme aplikaci, která ovlivňuje nějaký klíč, najdeme požadovaný klíč na zdrojovém počítači a přetáhneme jej v podstatě do téhož umístění v cílovém počítači.

Instalační program se poté při instalaci sám postará o to, aby požadované a potřebné změny v registrech byly provedeny. Pokud naše aplikace při svém běhu něco zapisuje do registrů, bude tedy fungovat jak na našem počítači (kde jsme příslušné klíče sami vytvořili), tak i na cílových počítačích (kde příslušné záznamy vytvoří instalační program).

Aby bylo možné s registry pracovat, je potřeba je asociovat s nějakou komponentou. Buď je možné registry přidat k již existující komponentě, nebo vytvořit komponentu novou, do které je třeba registry vložit. Výhodou samostatné komponenty s vybranými registry je pak možnost během instalace s danou komponentou pracovat tak, že může, ale nemusí být, daný registr za určitých podmínek nainstalován. [2][9]

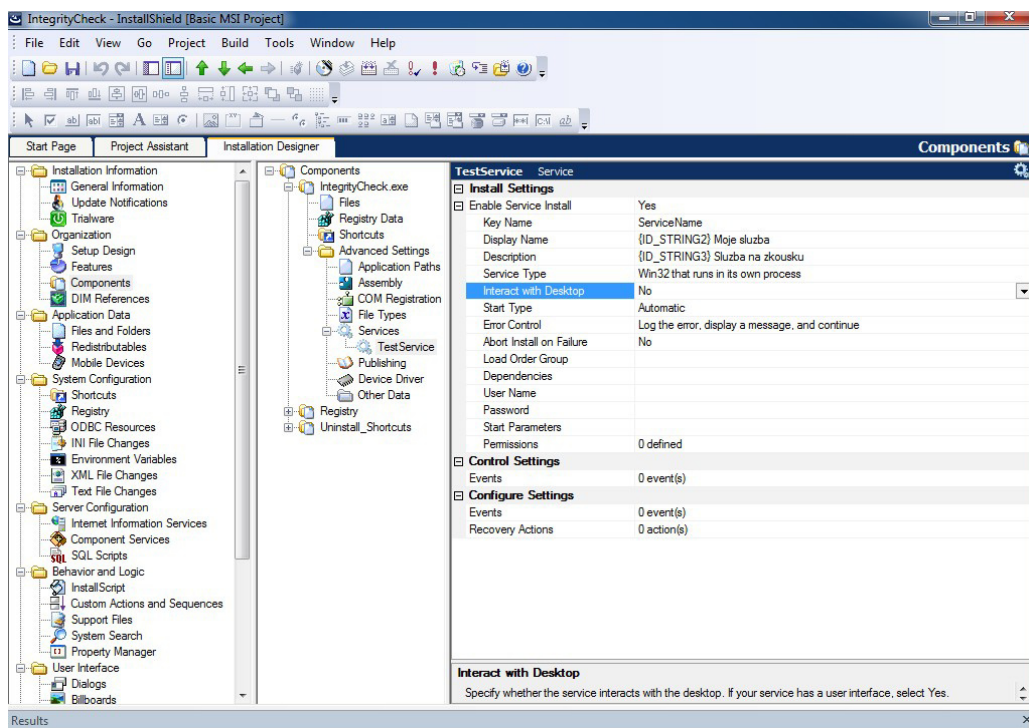


Obř. 12. Registry

4.1.6 Vytváření služeb

Windows služby jsou spustitelné soubory, které běží na pozadí a většinou řídí různé systémové úlohy i v době, kdy není uživatel právě přihlášen. Automaticky takto nelze použít jakoukoliv aplikaci, ale aplikace musí být již pro tento běh navržena. Obecně platí, že služby systému Windows mohou být spuštěny automaticky pokaždé, když systém startuje nebo se spouští ručně na vyžádání uživatele. Windows Installer umožňuje instalovat a konfigurovat nové služby bez nutnosti spouštět externí příkaz, například z příkazové řádky.

Ve vývojovém prostředí InstallShieldu je možné služby jednoduše vytvářet právě v nastavení jednotlivých komponent. Pokud tedy vytvoříme komponentu, ve které bude součástí nějaká aplikace vyžadující běh jako služba, je možné jednoduše vytvořit i parametry služby jako jsou Display Name (název, pod kterým je služba zobrazována v nastavení Windows), Description (popisek), Start Type (jak a kdy se má služba spouštět), Error Control (jakým způsobem se mají ukládat logy) a jiné viz Obr. 13. [9]



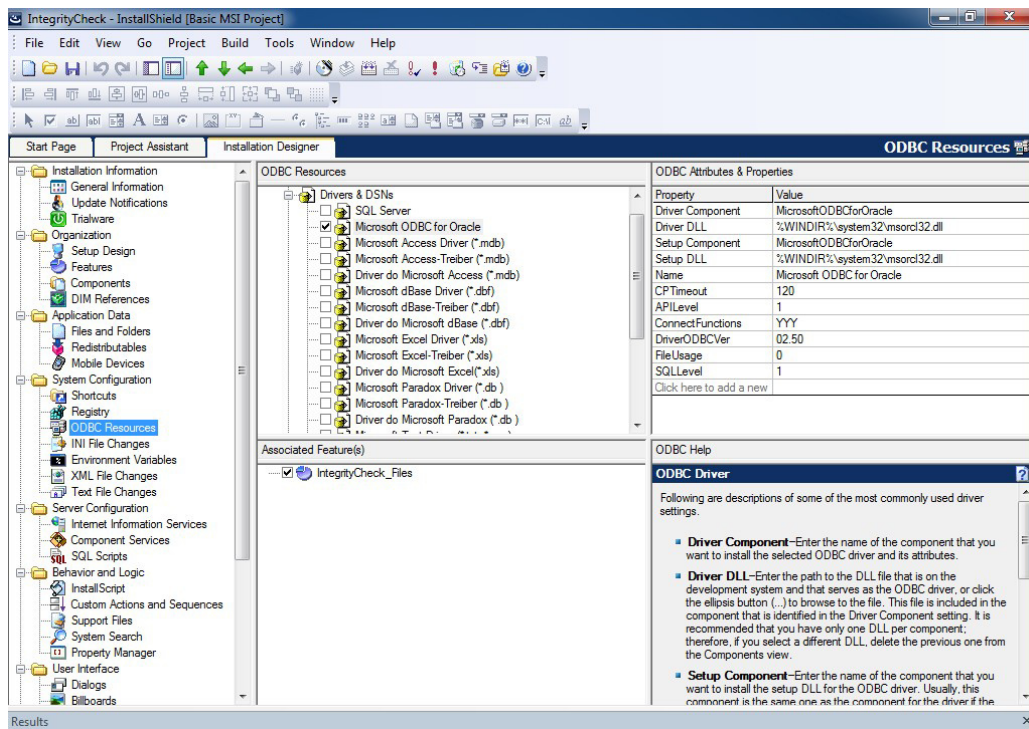
Obr. 13. Služby

4.1.7 ODBC Resources

Pokud aplikace pracuje s datovými zdroji ODBC, například máme databázovou aplikaci, která využívá ODBC datový zdroj Oracle ODBC Driver, je možné tento zdroj instalovat jako součást instalačního balíčku. Stačí pouze vybrat položku ODBC Resources a vybrat ten správný typ. Datový zdroj pak bude zahrnut do instalace.

Po zatržení každého požadovaného datového zdroje je nutné v dolní části obrazovky označit vlastnosti (Features), pro které chceme příslušný ODBC zdroj zahrnout do instalace. Kromě toho je samozřejmě možné v pravé části obrazovky modifikovat vlastnosti daného datového zdroje ODBC viz Obr. 14.

Přidání ODBC datového zdroje do instalace nám usnadní práci s konfigurací aplikace na cílovém počítači. Stačí zajistit přístup k serveru (Oracle) a datový zdroj bude nainstalován současně s instalací aplikace automaticky. [2][9]



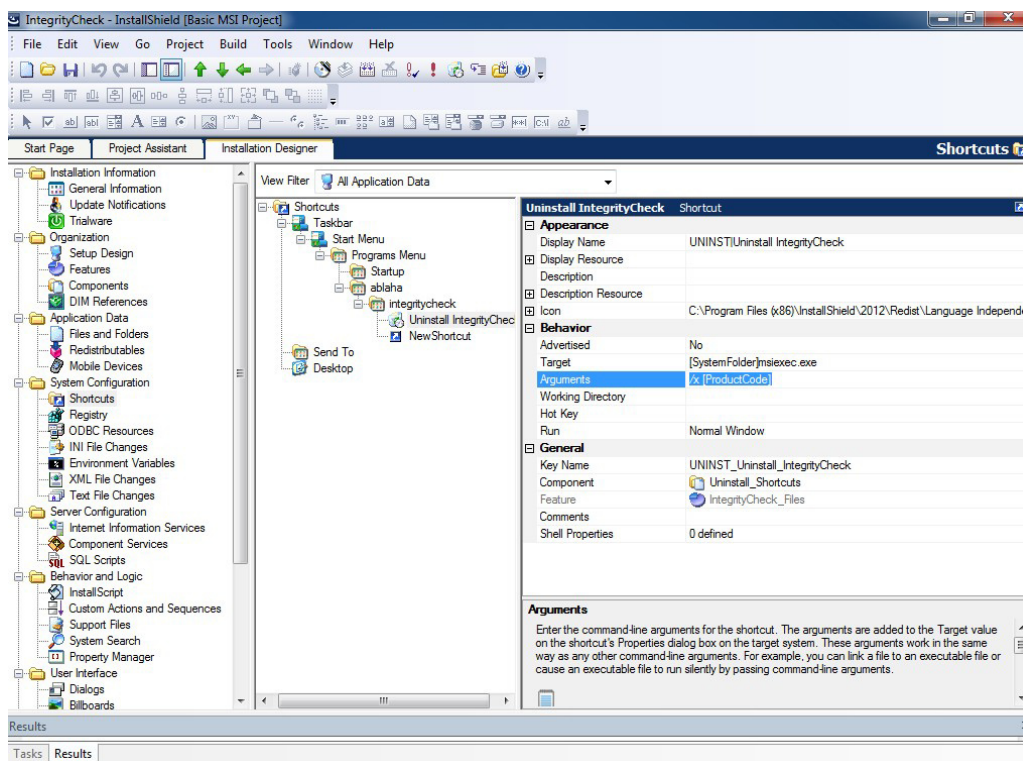
Obr. 14. ODBC

4.1.8 Shortcuts

V této části se určuje, jací zástupci se mají vytvořit na cílovém počítači při instalaci aplikace a jaké se mají vytvořit složky. Chceme-li, aby aplikace při své instalaci v umístění, jako je například nabídka Start nebo Plocha vytvořila novou složku nebo nového zástupce,

klepneme na příslušné umístění pravým tlačítkem a zvolíme New Shortcut nebo New Folder.

Velmi častým odkazem bývá spuštění hlavní aplikace, kde stačí pouze nasměrovat odkaz na daný spustitelný program, případně ho doplnit o spustitelný parametr nebo odkaz, který vede k odstranění aplikace.



Obr. 15. Shortcuts

Odkaz na odstranění aplikace se vytváří podobně jako u aplikace standardní viz Obr. 15 s tím rozdílem, že není směřován na aplikaci, která je obsažena v instalačním balíčku, ale na program, který se předpokládá, že již na systému existuje. V tomto případě se jedná o program msiexec.exe, která je součástí Windows Installer balíčku a je umístěna ve složce system32. Jako cíl tedy zvolíme [SystemFolder]msiexec.exe (proměnná [SystemFolder] odkazuje na složku system32) a jako spustitelný argument pak parametr /x [ProductCode], kde parametr /x určuje právě spuštění odinstalace balíčku a proměnná [ProductCode] určuje, který balíček se má odinstalovat. V tomto případě se spustí odinstalace ve standardním grafickém režimu, ale je samozřejmě možné odinstalaci spustit na pozadí bez nutnosti další uživatelské interakce, postačí přidat parametr /qn. [2][9]

4.1.9 Release

Pokud je instalační balíček připraven, je potřeba ho zkompilevat do výstupního formátu. Ať už bude výsledný formát jakýkoliv (msi, msm, exe), vše se definuje v části Release. Můžeme si zvolit formát, název souboru, výstupní adresář a v placené verzi je možné i balíček digitálně podepsat. Následně už stačí spustit kompilace tlačítkem Build a vytvoří se první instalační balíček.

5 DIGITÁLNÍ PODPIS INSTALAČNÍCH BALÍČKŮ

Jestliže na papírový dokument připojíme svůj podpis, ručíme tím vlastně za jeho obsah, takže dokument je pro příjemce důvěryhodný. Protože každý člověk má svůj charakteristický rukopis, vlastnoruční podpis jednoznačně identifikuje osobu, která podpis vytvořila. Pokud chceme zabránit dodatečnému pozměnění dokumentu, vytvoříme jej prostě ve dvou kopiích, přičemž jednu kopii si ponechá podepisující strana, druhou kopii příjemce. Případnou jednostrannou modifikaci dokumentu lze potom velmi snadno prokázat.[16]

U elektronických dokumentů je situace poněkud složitější. Není sice problém pod dokument připojit svůj podpis, ale jeho důvěryhodnost je prakticky nulová. Zatímco vlastnoruční podpis není zrovna lehké napodobit (případný grafologický rozbor by zfalšovaný podpis prokázal), v případě elektronického textu je tomu přesně naopak. Podepsat se cizím jménem může kdokoliv. Elektronický dokument je navíc možné velmi snadno modifikovat, aniž by to příjemce dokázal rozpoznat. Z tohoto důvodu bylo potřeba najít prostředek, který by umožnil ověřit, že elektronický dokument skutečně podepsala daná osoba a že v dokumentu nedošlo k žádným dodatečným změnám. Řešení nabídla technologie digitálního podpisu, která využívá výhod asymetrické kryptografie. [16]

5.1 Technologie šifrování

Šifrování zpráv se používá již velmi dlouho a jeho cílem je zabránit odposlouchávání zpráv na jejich cestě od odesílatele k příjemci. S nějakou jednodušší šifrou se zřejmě potkal každý: pomocí klíče zprávu zakódujeme (například nahradíme písmena jinými symboly), odešleme ji a příjemce ji pomocí stejného klíče dekóduje do původní čitelné podoby. Toto takzvané symetrické šifrování je sice velmi jednoduché a rychlé, ale má jednu obrovskou nevýhodu. Je nutné nějakým bezpečným způsobem předat příjemci šifrovací klíč. Proto byla vyvinuta asymetrická šifra, která využívá dvojice klíčů. Tyto klíče jsou zvoleny tak, že zprávu zakódovanou jedním klíčem je možné dekódovat pouze druhým klíčem z dvojice, a naopak. Jeden klíč se nazývá soukromý klíč a zná jej pouze příjemce zprávy. Druhý, veřejný klíč, potom předá odesílateli. Odesílatel pomocí veřejného klíče zprávu zašifruje a pošle ji příjemci, který ji pomocí svého soukromého klíče dešifruje. Protože zprávu

zašifrovanou veřejným klíčem není možné stejným klíčem dešifrovat, je možné veřejný klíč předat odesílateli nechráněným kanálem nebo jej prostě přímo zveřejnit. [16]

5.2 Technologie digitálního podpisu

Digitální podpis funguje v podstatě na opačném postupu než klasické šifrování zpráv. Zprávu, kterou zašifrujeme pomocí soukromého klíče, je možné dešifrovat klíčem veřejným. Na první pohled to sice nedává smysl, ale pokud dokážeme zprávu dešifrovat, znamená to, že ji nemohl zašifrovat nikdo jiný než vlastník soukromého klíče. Zjednodušeně můžeme říci, že pro podepsání stačí k elektronickému dokumentu připojit tentýž dokument zašifrovaný soukromým klíčem. Pro ověření potom stačí zašifrovanou část dešifrovat pomocí veřejného klíče a porovnat s nešifrovanou zprávou. Pokud se obě zprávy shodují, prokázali jsme, že zprávu zašifroval majitel soukromého klíče. Protože je prakticky nemožné z jednoho klíče odvodit druhý párový klíč, není možné takovýto podpis zfalšovat. [16]

Vzhledem k faktu, že asymetrické šifrování je výpočetně náročné, šifrování delších zpráv by trvalo velmi dlouho. Navíc připojením celé šifrované zprávy by se zbytečně zvětšoval datový objem zprávy původní. Z tohoto důvodu se v praxi nešifruje celá zpráva, ale pouze její tzv. otisk respektive hash, který je vysvětlen v kapitole Integrita vstupních souborů. [16]

Aby byl digitální podpis opravdu důvěryhodný, musíme ještě prokázat, že veřejný klíč, pomocí kterého byl podpis vytvořen, skutečně patří podepisující osobě. K tomu slouží takzvaný certifikát, což je vlastně dokument vydaný třetí důvěryhodnou stranou (certifikační autoritou), která stvrzuje, že podepisující osoba je vlastníkem daného páru klíčů. Tento certifikát je potom připojen k vlastnímu podpisu. [16]

5.3 Výhody podpisu instalačních balíčků

Digitální podpis instalačních balíčků napomáhá chránit uživatele před nežádoucími změnami a to ověřením, že hash uložený v digitálním podpisu odpovídá hashi aktuálního instalačního balíčku. Dalším bezpečnostním prvkem digitálního podpisu je také možnost ověřit, že instalační balíček přišel z určitého (požadovaného) zdroje, a to zašifrováním

hashe s vydavatelem privátního klíče. Ověření podpisu pak může proběhnout buď pomocí veřejného klíče vydavatele, nebo důvěryhodnou certifikační autoritou.

Výhodou využití instalace pomocí Instalační služby Windows je i fakt, že není nutné ručně kontrolovat digitální podpisy balíčků. Instalační služba systému Windows automaticky ověřuje, jestli byl balíček změněn právě tehdy, pokud obsahuje digitální podpis, a to hned při pokusu o instalaci tím, že volá nativní funkci WinVerifyTrust.

5.4 Podpis instalačních balíčků na platformě Windows

Jak tedy instalační balíček podepsat? Většina developerských nástrojů má většinou svůj vlastní způsob podpisu, ale existuje i univerzální řešení, a to pomocí aplikace SignTool. Tato aplikace je volně šiřitelná a je součástí Microsoft Windows SDK, a je tedy ke stažení na stránkách Microsoftu. Použití je celkem snadné, je možné využít grafického průvodce, nebo přímo zadat parametry z příkazové řádky.

Spuštění pomocí průvodce.

```
signtool signwizard
```

Příklad spuštění pomocí zadání parametrů z příkazové řádky.

```
signtool sign /f certfile.pfx /p password /t timestampURL app.msi
```

Jak je z příkladu vidět, je v první řadě potřeba získat samotný certifikát. Vzhledem k tomu, že plánujeme poskytnout digitálně podepsaný, dokument (v našem případě instalační balíček) jiným uživatelům a chceme, aby příjemce mohl ověřit pravost našeho digitálního podpisu, je nejvhodnější požádat o certifikát některou známou certifikační autoritu (CA).

5.4.1 Certifikační autorita

Certifikační autorita je v asymetrické kryptografii subjekt, který vydává digitální certifikáty a svojí autoritou potvrzuje pravdivost údajů. Na českém trhu působí mnoho komerčních certifikačních autorit, které mají většinou své veřejné klíče umístěny přímo ve webových prohlížečích a dalších programech, a tím tak napomáhají v procesu rozhodování o míře důvěry webových serverů, digitálně podepsaných e-mailů a jiných dat. Vybereme si tedy CA a projdeme procesem vydání takového certifikátu, který umožňuje podpis dat.

5.4.2 Obecný proces vydání certifikátu

Před procesem vlastního vydání certifikátu si žadatel nejprve ve svém počítači vygeneruje soukromý a veřejný klíč a vytvoří elektronickou žádost o certifikát. Celá žádost je poté podepsána vygenerovaným soukromým klíčem, a tím žadatel o certifikát prokazuje, že je vlastníkem daného páru klíčů. Dále je nutné žádost doručit CA, která ověří, že údaje uvedené v žádosti jsou pravdivé (kontrola dokladu totožnosti žadatele, výpis z obchodního rejstříku) a že podpis žádosti je platný. CA následně vystaví na základě žádosti platný certifikát; ten je doručen žadateli, který si jej nainstaluje do svého systému

V praxi je výše uvedený proces téměř automatický; většinou je potřeba navštívit webové stránky CA, zvolit žádost o certifikát a do zobrazeného formuláře vyplnit požadované údaje. Vygenerování klíčů i žádosti a její odeslání je poté zcela automatické. Vystavený certifikát je pak možné stáhnout z webových stránek CA.

Pro úplnost je potřeba dodat, že existuje několik typů certifikátů, rozdělených podle toho, k jakému účelu budou používány. Certifikát tedy nemusí být vydán pouze fyzické osobě, ale třeba i osobě právnické, jež může, např. elektronicky, podepisovat produkovany software, a to je právě případ podpisu instalačních balíčků. [16]

6 NASZENÍ POMOCÍ GROUP POLICY

Některé aplikace je vhodné instalovat napříč celou firmou a s co nejmenší námahou. Pro tuto instalaci slouží jako základní metoda Group policy (Skupiny zásad), která umí distribuovat aplikace do celé sítě prostřednictvím Microsoft Windows Serveru.

Skupiny zásad je nástroj pro hromadnou správu oprávnění a nastavení, které je možné aplikovat jak na celý počítač, tak na přihlášeného uživatele. Ve skupinách zásad je možné vytvářet kolekce nastavení, kterým říkáme Group Policy Object (GPO), které umožňují měnit konkrétní parametry chování počítače nebo uživatele. Samotné nastavení GPO se pak odkazuje na jednotlivé organizační jednotky (OU) v Active Directory (AD).

Každý počítač od Windows 2000 má lokální politiky (local Group Policy), které ovlivňují lokální počítač a přihlášené uživatele. V našem případě se ale budeme pohybovat v prostředí, kde jsou jednotlivé klientské počítače zařazeny do politiky doménové. Doménové politiky lze použít výhradně u počítačů a uživatelů, kteří jsou členy nějaké domény. Existují dvě základní doménové politiky, které jsou vytvořeny již při instalaci AD.

- Default Domain Policy - Tato politika je spojena s kontejnerem domény a ovlivňuje všechny objekty v doméně.
- Default Domain Controller Policy - Tato politika je spojena s kontejnerem doménového kontroleru (DC) a ovlivňuje všechny DC v doméně.

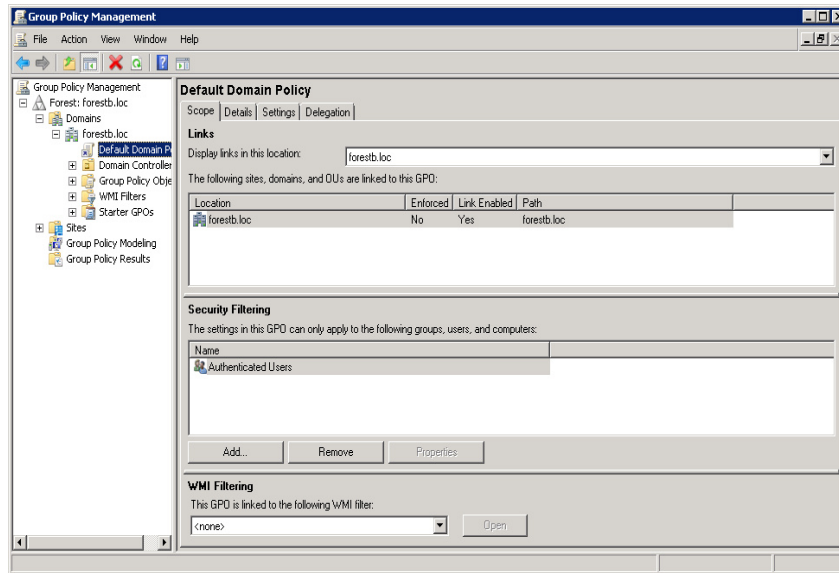
Group Policy obsahuje něco přes 2 400 různých nastavení ovlivňujících chování jak počítače, tak přihlášeného uživatele. Tímto způsobem tedy můžeme spravovat potenciálně tisíce počítačů nebo uživatelů změnou jednoho GPO.[18]

Nevýhodou GPO je, že umí distribuovat pouze aplikace formátu MSI a velká většina aplikací jsou stále ve starém formátu EXE. V takových případech je možné (pokud to jde) aplikaci převést do formátu MSI, nebo použít nějaký z placených nástrojů, jako je například System Center Configuration Manager (SCCM).

Pokud máme MSI balíček, který chceme distribuovat do rozlehlé sítě, a nechceme investovat do dalších profesionálních nástrojů, je nejjednodušší metodou distribuce na cílové klientské počítače prostřednictvím Group Policy.

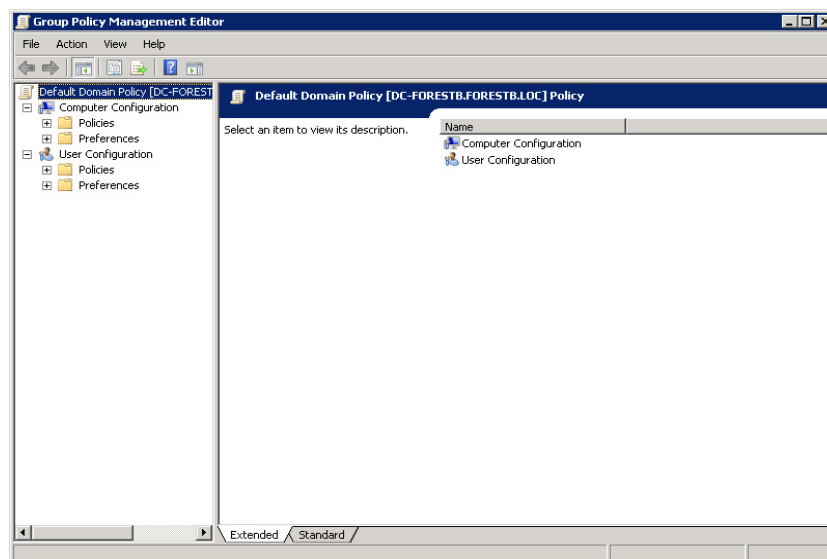
6.1 Ukázka distribuce pomocí Group Policy

Nejčastějším nástrojem je Group Policy Management konzole (Obr. 16), kterou na Windows Serveru je možné nalézt v nástrojích pro správu a editor objektu zásad skupiny (Obr. 17).



Obr. 16. Management konzole

Pro instalaci MSI balíčku je nutné vytvořit si objekt zásad skupiny pomocí editoru zásad skupin a hned v prvním kroku je velmi nutné si určit, chcete-li balík instalovat se zapnutím (restartem) počítače, nebo s přihlášením uživatele. Obě varianty v sobě skrývají určité výhody i zdánlivé nevýhody.



Obr. 17. Editor zásad skupin

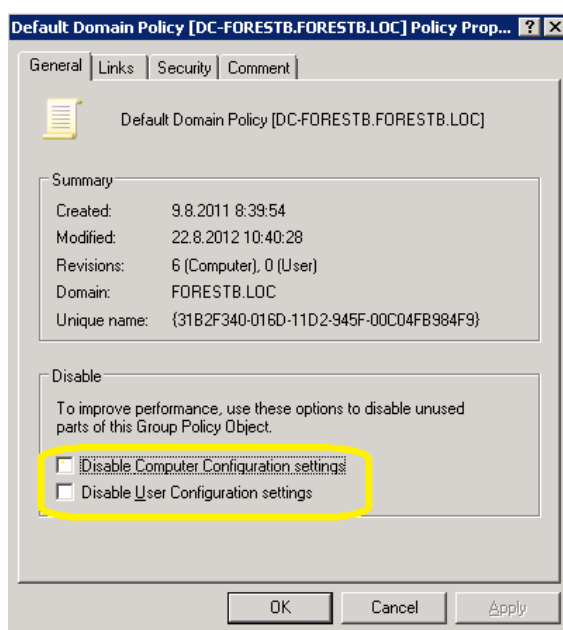
Instalace s počítačem:

- Objekty v organizační jednotce v AD, kde je objekt zásad skupiny linkován, musí být typu počítače.
- Instalace samotná začne se startem počítače, ještě před výzvou k ověření uživatele. Může tak dojít k časové prodlevě.
- Balíček je nařízen, nelze tedy jakkoliv ovlivnit jeho instalaci.

Instalace s uživatelem:

- Objekty v organizační jednotce v Active Directory, kde je objekt zásad skupiny linkován, musí být typu uživatel.
- Instalace se může spustit s přihlášením uživatele.
- Balíček může být přiřazen, tedy instalace se spustí bez možnosti zásahu uživatele, nebo může být pouze publikován, tedy uživatel má právo pomocí nástroje „programy a funkce“ v systému Windows ovlivnit, zda si přeje balíček instalovat nebo nikoliv.

Pokud jste se rozhodli, který z modelů instalace zvolíte, doporučuje se přes kontextovou nabídku v editoru zásad skupiny vypnout druhou možnost instalace. Zrychlí se tím samotné načítání objektu zásad skupiny cílovým počítačem (Obr. 18).



Obr. 18. Vypnutí části GPO

Přidáním balíčku pomocí kontextového menu dojde ke spuštění prvního průvodce, kde se určuje model instalace (Přiřazené, Publikované, Rozšířené).

Na jednotlivých záložkách naleznete možnosti, které můžete v rámci daného balíčku použít. Jsou to zejména tyto:

- Záložka General - možnost upravit prezentované jméno aplikace a vyplnit kontakt na podporu
- Záložka Deployment – možnost nastavit ignorování jazykového nastavení při instalaci balíčku nebo umožnit instalaci 32bitového balíčku na 64bitové systémy.
- Upgrades – umožňuje provádět upgrade aplikace pomocí jiného balíčku.
- Záložka Modifications – umožňuje přidání „Windows Installer Transform package“, tedy .mst soubor, který umí modifikovat různá nastavení Windows Installer balíčku.
- Záložka Security – určuje, na které počítače v rámci organizační jednotky bude aplikace nainstalována.

Na tomto výpisu vidíte, že možnosti instalace pomocí Group Policy jsou sice omezené, ale rozhodně ne nikterak bezvýznamné. Je možné za velmi příznivé náklady, tím se rozumí pouze čas správce, ušetřit nemalé finanční prostředky hromadnou instalací aplikací.[15]

7 SPRÁVA HROMADNÉ DISTRIBUCE INSTALAČNÍCH BALÍČKŮ POMOCÍ NÁSTROJE SYSTEMCENTER CONFIGURATION MANAGER

Systém Center Configuration Manager (SCCM) funguje ve stejném prostředí jako v předchozí kapitole zmiňované Group Policy, tedy v prostředí doménovém s Active Directory. Tímto způsobem je i zajištěna jejich provázanost a veškeré informace, které SCCM používá, jsou poskytovány právě z tohoto prostředí.

SCCM je komplexní řešení umožňující IT správcům nasazovat operační systémy, aplikace a softwarové a hardwarové aktualizace do klientů, serverů a mobilních zařízení v rámci distribuované sítě. Správci mohou využívat stejné nástroje a procesy pro nasazení do všech fyzických i virtuálních zařízení bez ohledu na to, kde se nacházejí.[19]

7.1.1 Distribuce softwaru

Správci mohou distribuovat aplikace a aktualizace do všech svých IT systémů, včetně stolních počítačů, serverů, přenosných počítačů a kapesních zařízení. Pro mobilní a geograficky vzdálené uživatele, kteří se k síti připojují zřídka, zahrnuje produkt Configuration Manager funkci Internet Based Client Management (IBCM), která zajišťuje bezpečné doručování aktualizací softwaru a aplikací v rámci Internetu. [19]

7.1.2 Správa softwarových aktualizací

Služba Software Update Management v produktu Configuration Manager je založena na známé službě WSUS (Windows Server Update Services) a zajišťuje jednotnou infrastrukturu pro aktualizace všech produktů společnosti Microsoft, jiných dodavatelů nebo interně vyvíjených podnikových aplikací. [19]

7.1.3 Možnosti instalace

Pomocí SCCM je možné instalovat

1. Windows Installer - to znamená MSI balíček, který je i nejvhodnější volbou.
2. Microsoft Application Virtualization.
3. Windows Mobile Cabinet - pro mobilní zařízení Windows mobile.

4. Nokia SIS file - mobilní zařízení s Nokia Symbian.
5. Script Installer - umožní instalaci z EXE nebo speciálních skriptů. [19]

7.2 Instalace SCCM 2007 na Windows Server 2008 R2

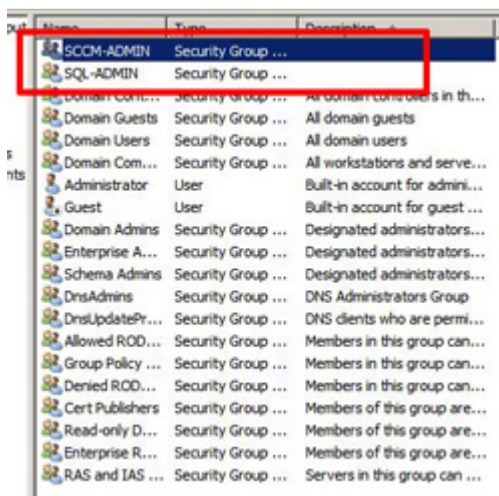
7.2.1 Krok 1: Nastavení SCCM domény

V první části instalace si vytvoříme skupiny a uživatele ve službě Active Directory, se kterými budeme dále pracovat. Je možné sice použít již nějaké dřívě, ale tato varianta je vhodnější.

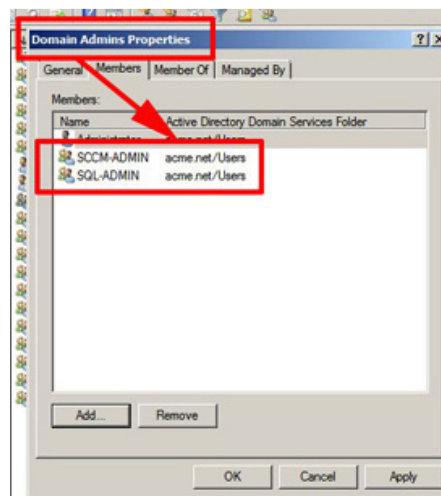
1) Vytvořte dvě skupiny ve službě Active Directory (Obr. 19).

- 1) SCCM skupina správců SCCM-ADMIN
- 2) SQL skupina správců SQL-ADMIN

2) Přidejte obě skupiny do skupiny „Domain Admins“ (Obr. 20).



Obr. 19. Active Directory Users



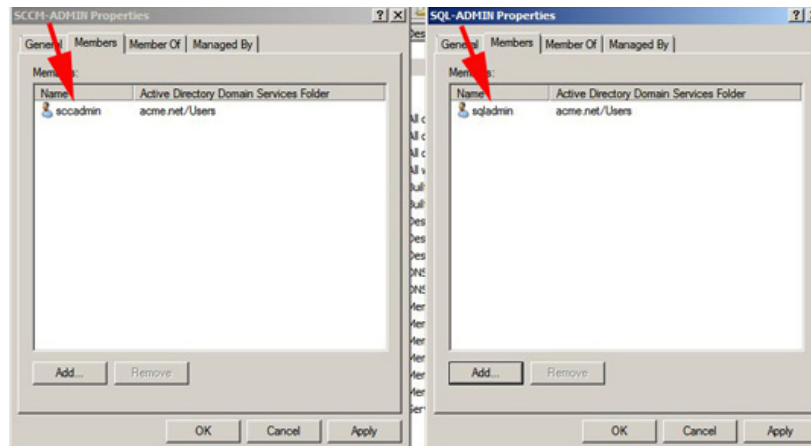
Obr. 20. Domain Admins settings

3) Vytvořte dva nové uživatele: sccmadmin a sqladmin (Obr. 21).



Obr. 21. Uživatelé ve skupině

- 4) Přidejte uživatele sccmadmin do skupiny SCCM-ADMIN a uživatele sqladmin do skupiny SQL-ADMIN (viz Obr. 22).

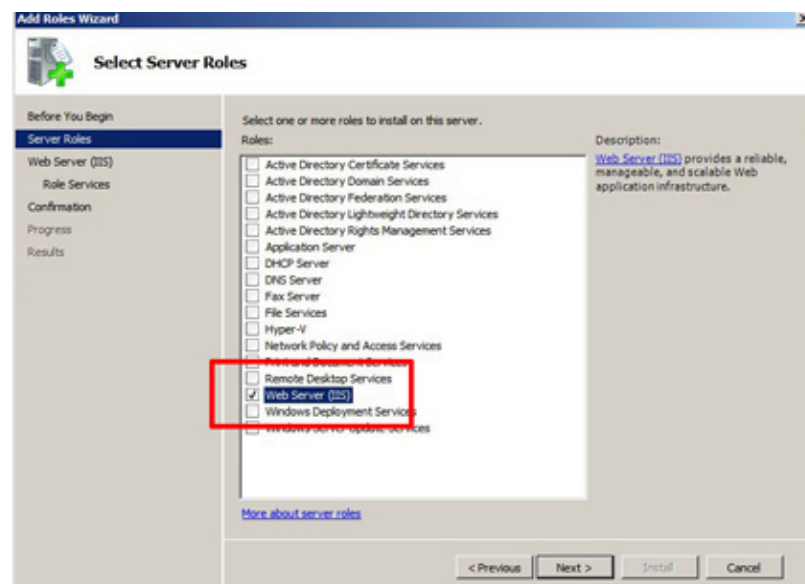


Obr. 22. Přidání uživatelů do skupin

7.2.2 Krok 2: SCCM prerequisites

V tomto kroku si připravíme aplikace a služby, které jsou k instalaci vyžadovány. Není potřeba nic dodatečně kupovat nebo stahovat, vše je součástí serverového operačního systému Windows. Jen je potřeba dané funkce zapnout a nakonfigurovat.

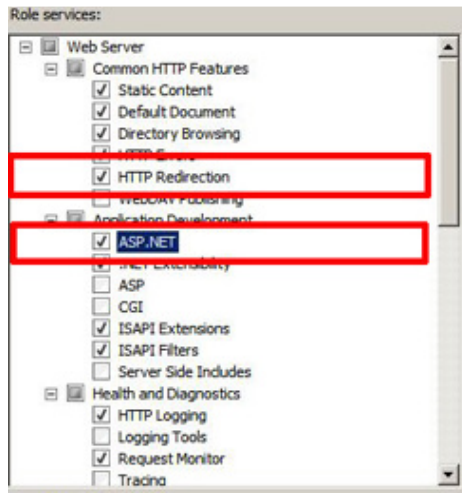
- 1) Přidejte Web Server IIS pomocí Web Server Role (Obr. 23).



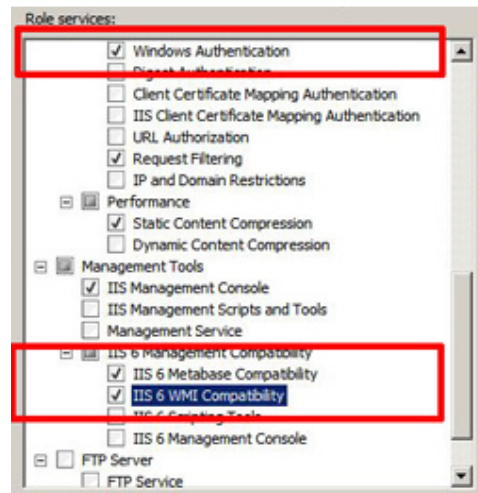
Obr. 23. Přidejte IIS

- 2) Přidejte do Web Server IIS následující Role (Obr. 24 a Obr. 25):
- HTTP redirection

- b) ASP.Net
- c) Windows Authentication
- d) IIS6 Metabase compatibility
- e) IIS6 WMI compatibility



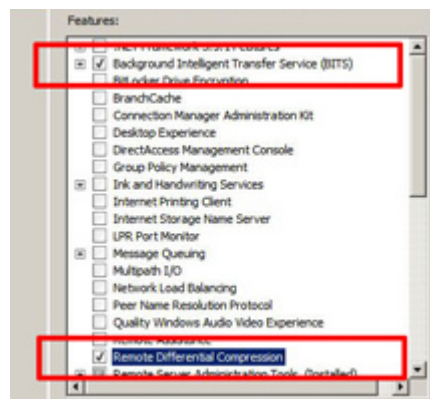
Obr. 24. Přidání rolí 1/2



Obr. 25. Přidání rolí do 2/2

3) Přidejte do serverových funkcí „server Features“ následující funkce (viz Obr. 26):

- a) Background Intelligent Transfer Service.
- b) Remote Differential Compression.



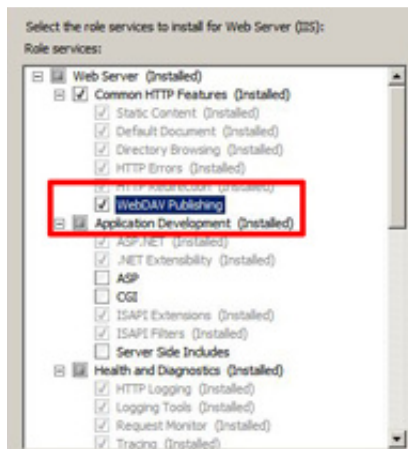
Obr. 26. Přidání serverových funkcí

4) Nainstalujte WebDAV

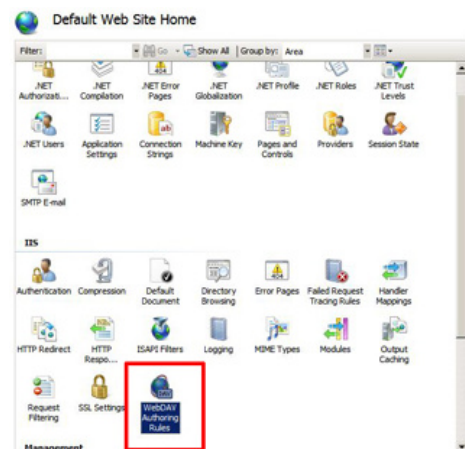
Server Manager > Roles > Web Server (IIS) > Add Role Services > WebDAV Publishing > Next > Close. (Obr. 27)

5) Povolte WebDAV

Start > Administrative Tools > Internet Information Services (IIS) Manager > Expand {server name} > Sites > Default Web Site > WebDav Authoring Rules. (Obr. 28)



Obr. 27. Nainstalování WebDAV



Obr. 28. Povolení WebDAV

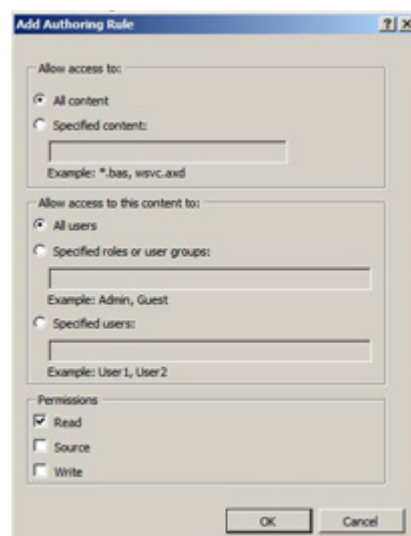
6) Vyberte Enable WebDav (Obr. 29)



Obr. 29. Enable WebDAV

7) Vyberte Add Authoring Rule a nastavte oprávnění

All Content > All Users > Permissions > Read > OK. (Obr. 30)



Obr. 30. WebDAV Authoring Rule

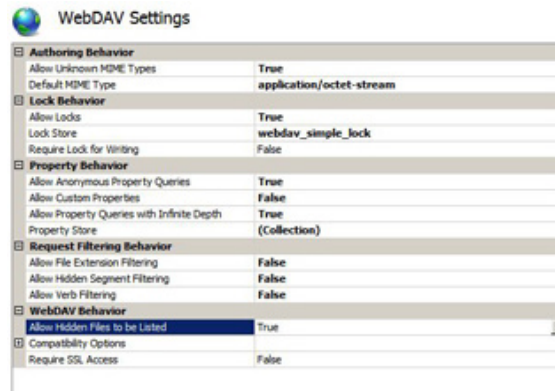
8) Vyberte pravidlo, které jste právě vytvořili > WebDav Settings. (Obr. 31)



Obr. 31. Odkaz WebDAV settings

9) Změňte hodnoty (Obr. 32):

- a) „Allow Anonymous Properties Queries" na True
- b) „Allow Custom Properties" na False
- c) „Allow Property Query with Infinite Depth" na True
- d) „Allow hidden files to be listed" na True



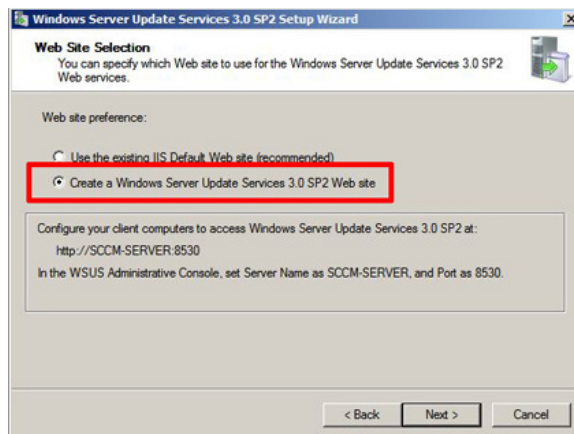
Obr. 32. WebDAV settings

10) Další podmínkou pro spuštění SCCM instalátoru je nainstalovaný WSUS 3.0 SP2 (Obr. 33)

11) Při instalaci WSUS vyberte "Full server installation" > a vyberte "Create a Windows Server Update Services 3.0 SP2 Web site" (Obr. 34).



Obr. 33. WSUS 3.0

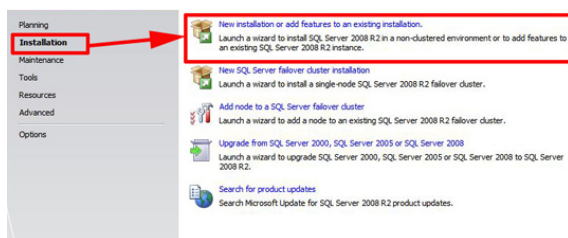


Obr. 34. WSUS Server Update

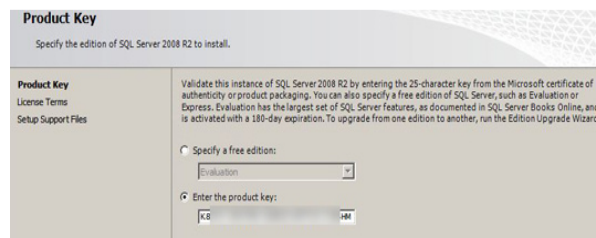
7.2.3 Krok 3: Instalace SQL Server

Nyní je potřeba nainstalovat Microsoft SQL Server. Jedná se o relační databázový a analytický systém, který bude sloužit jako úložiště informací pro SCCM. V našem případě bude stačit verze Express, kterou je možné stáhnout na stránkách www.microsoft.com.

- 1) Z instalačního media SQL Serveru spusíte setup.exe > Installation > "New Installation or add features to an existing Installation" > OK. (Obr. 35)
- 2) Vložte Product key > Next > "I accept ..." > Next > Install > Next > Next. (Obr. 36)

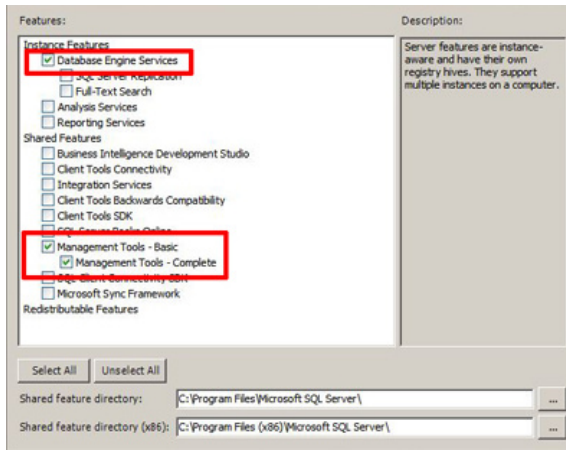


Obr. 35. Instalace SQL Server

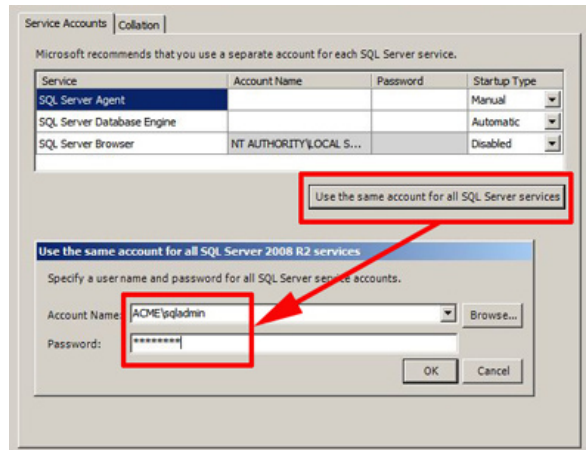


Obr. 36. Product key

- 3) Zaškrtněte Database Engine Services a Management Tools > Next. Next > Accept the defaults > Next > Next. (Obr. 37)
- 4) Na konfigurační stránce serveru > vyberte "Use the same account for all SQL Server services" > dále vyberte uživatele (sqladmin) > a nastavte SQL Server Agent and SQL Server Database Engine Startup type na "Automatic" > Next. (Obr. 38)



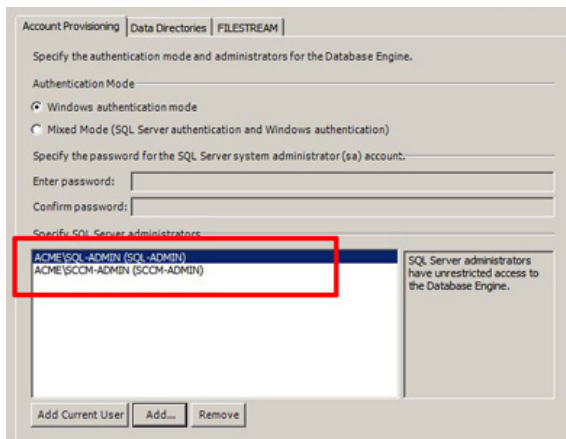
Obr. 37. Database Management



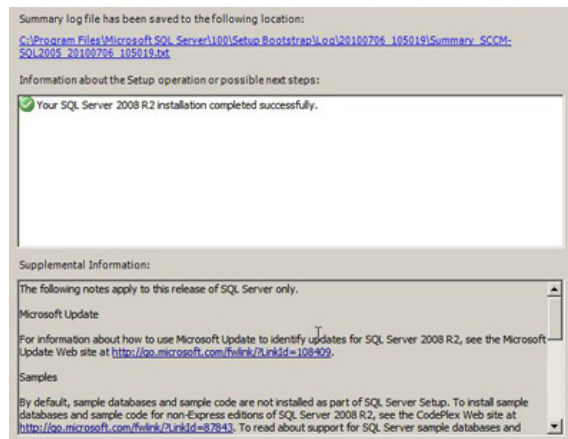
Obr. 38. Use the same account

5) Nastavte "Windows Authentication" > přidejte SCCM-ADMIN skupinu a SQL-ADMIN skupinu > Next > Next > Next > Install. (Obr. 39)

6) Po dokončení klikněte na close. (Obr. 40)



Obr. 39. Windows Authentication



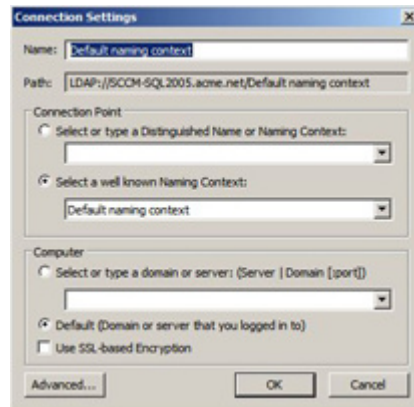
Obr. 40. Konec instalace

7.2.4 Krok 4: Připravte Active Directory pro SCCM

V tomto kroku si připravíme službu Active Directory tak, aby byla připravena na instalaci SCCM.

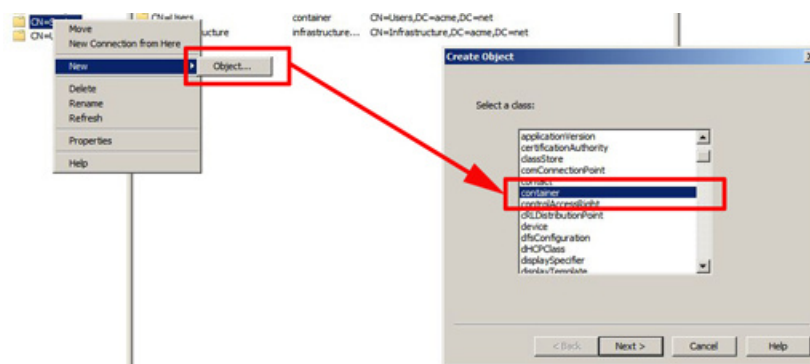
1) Z instalačního média spusťte > SMSSETUP > BIN > 1386 > extadsch.exe

- 2) Překontrolujte úspěšné rozšíření schématu, otevřete soubor c:\extADsch.txt a vyhledejte „successfully extended the Active Directory Schema“.
- 3) Nyní je třeba vytvořit objekty ve službě Active Directory > Start > Administrative tools > ADSI Edit > Action > connect to > OK. (Obr. 41)



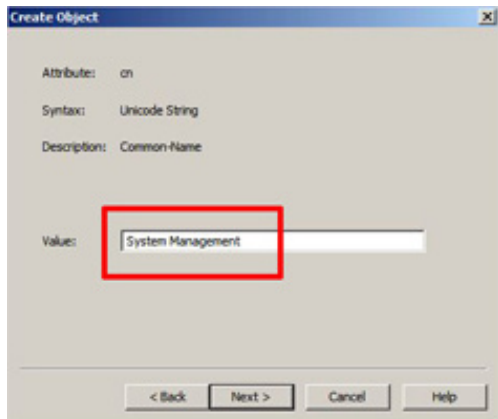
Obr. 41. ADSI Edit

- 4) Rozbalte Default naming context > rozbalte doménové jméno > Klikněte pravým tlačítkem na „system“ > New > Object > Container > Next. (Obr. 42)

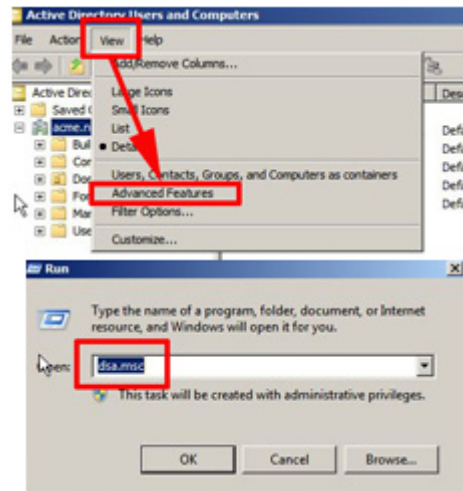


Obr. 42. ADSI Edit nový objekt

- 5) Do pole Value vyplňte „System Management“ a pokračujte na> Next > Finish (Obr. 43)
- 6) Spusťte Active Directory Users and Computers > Start > dsa.msc {enter} > a vyberte View > Advanced. (Obr. 44)

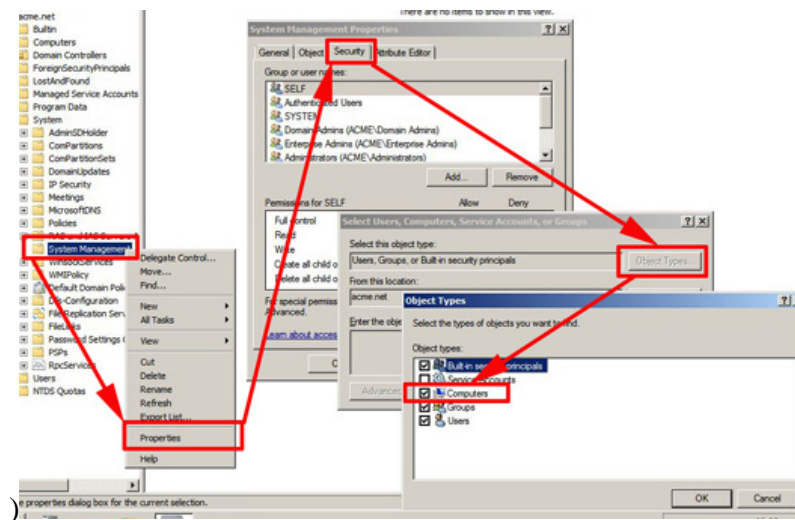


Obr. 43. System Management



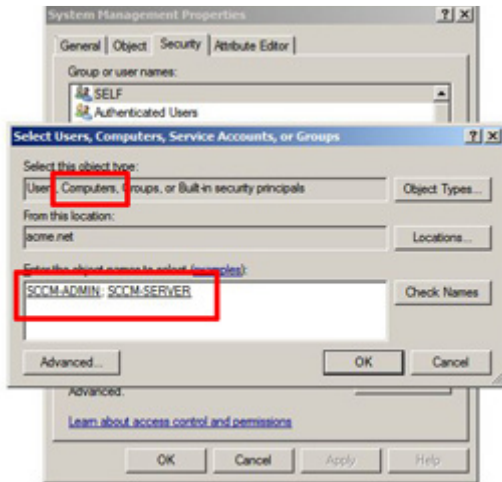
Obr. 44. Advanced view

- 7) Rozbalte „system“ > vyhledejte „System Management“ > klikněte pravým tlačítkem myši a vyberte properties > Security > Add > Object Types > zaškrtněte Computers > OK. (Obr. 45)

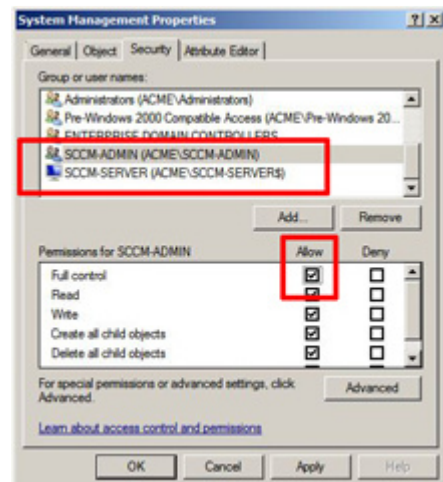


Obr. 45. Security

- 8) Klikněte na Advanced > Find Now > vyhledejte a přidejte SCCM-ADMIN skupinu a SCCM Server > OK. (Obr. 46)
- 9) Udělte oprávnění „Full Control“ na SCCM-ADMIN skupinu a na SCCMserver. (Obr. 47)

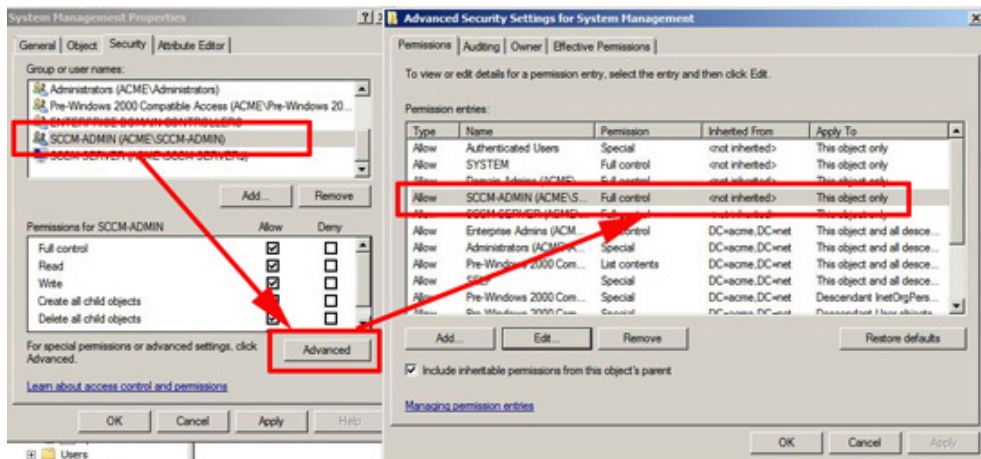


Obr. 46. Přidání skupin



Obr. 47. Oprávnění

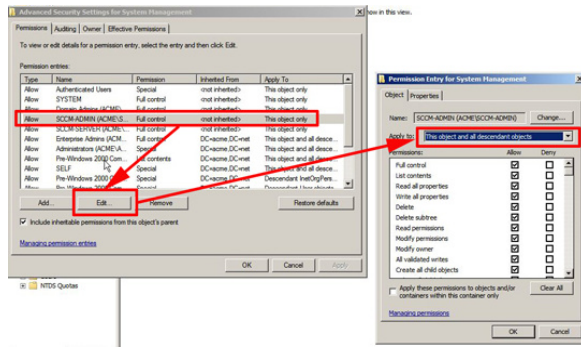
10) Klikněte na Advanced > vyberte SCCM-ADMIN skupinu > Edit. (Obr. 48)



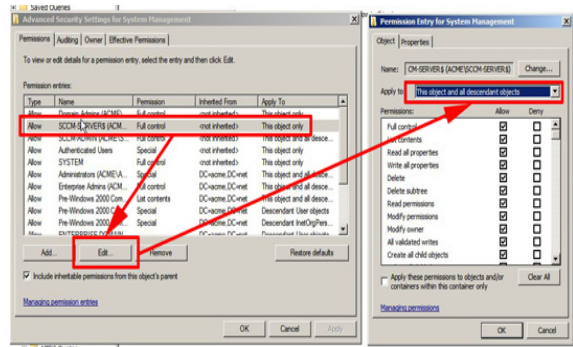
Obr. 48. Editace skupiny SCCM-ADMIN

11) V „Apply to“ změňte hodnotu na „This object and all descendant objects“ > OK > Apply > OK. (Obr. 49)

12) Stejným způsobem nastavte hodnoty i pro SCCM-Server. (Obr. 50)



Obr. 49. This object



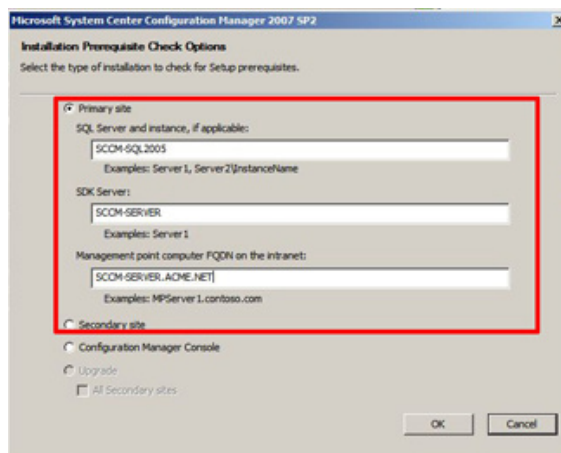
Obr. 50. SCCM-Server

7.2.5 Krok 5: Instalace SCCM

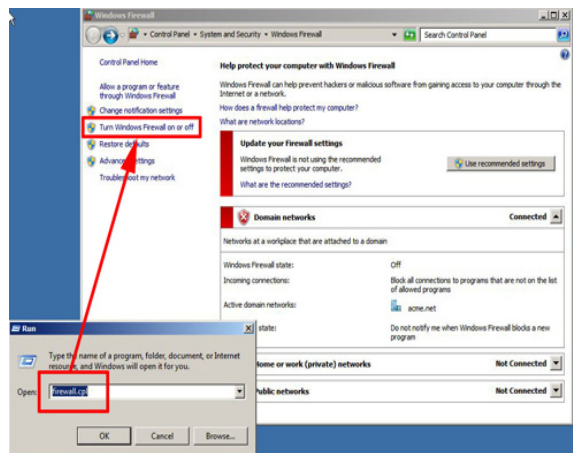
Nyní již máme vše připraveno a můžeme spustit samotnou instalaci SCCM.

- 1) Přihlaste se jako uživatel sccmadmin.
- 2) Z SCCM média spusťte splash.hta > vyberte Pre requisite checker > do Primary site doplňte SQL Server name > SCCM server name a FQDN > OK. (Obr. 51)

Poznámka: Pokud máte problém s komunikací se SQL serverem, zkontrolujte, zda ji neblokuje Windows firewall. (Obr. 52)

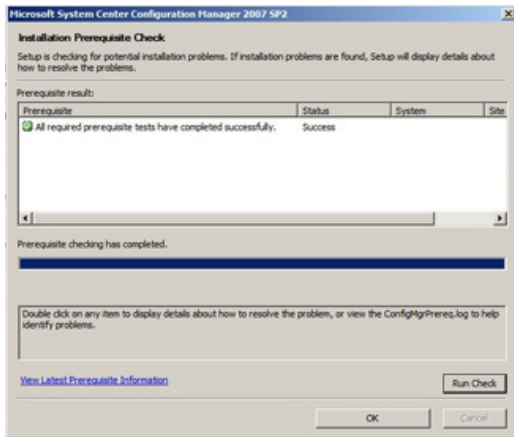


Obr. 51. SCCM instalace

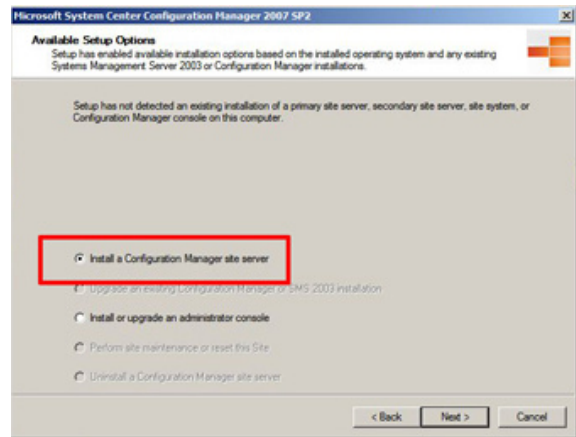


Obr. 52. Vypnutí firewallu

- 3) Nyní by mělo být vše připravené a test by měl skončit s hláškou „All required pre requisite tests have completed successfully" > OK. (Obr. 53)
- 4) Nyní znovu spusťte Splash.hta > vyberte > Install configuration manager 2007 > Next > "Install Configuration Manager site Server > Next. (Obr. 54)



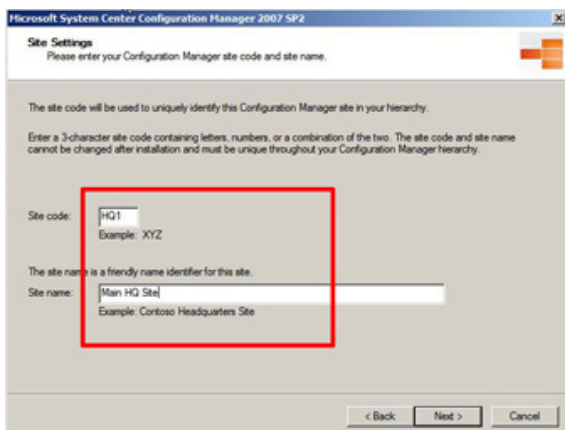
Obr. 53. Správné dokončení



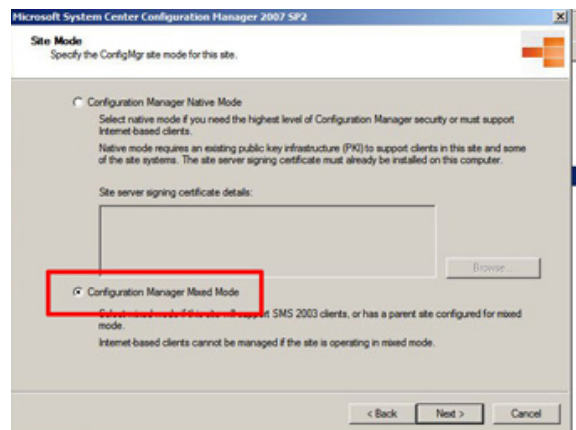
Obr. 54. Install Manager site

5) Zaškrtněte „I accept these License terms“ > Next > Custom Install > Next > Primary Site > Next > Next > zadejte sériový klíč > Next > potvrďte instalační adresář > Next > vložte Site code a Site name > Next. (Obr. 55)

6) Změňte Site mode na "Configuration Manager Mixed Mode" > Next. (Obr. 56)



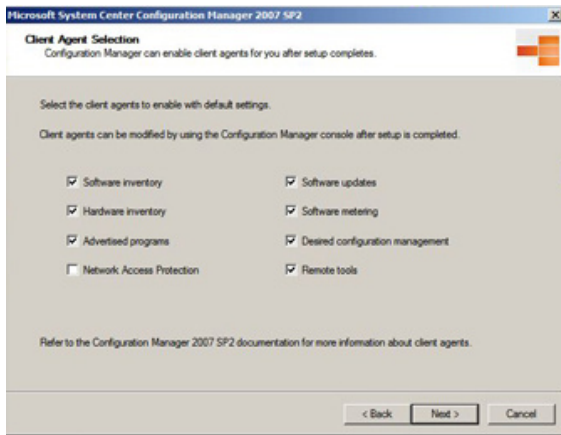
Obr. 55. Site code a Site name



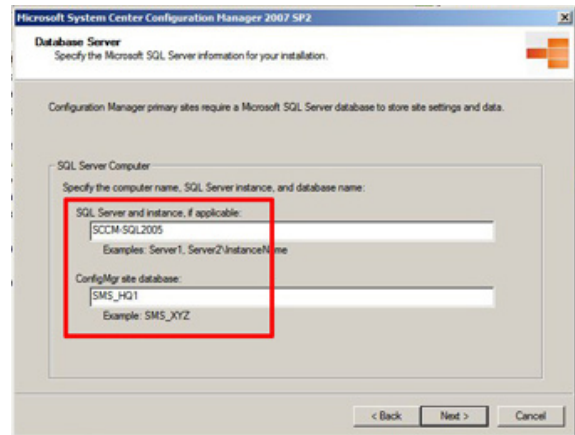
Obr. 56. Conf. Mixed Mode

7) Vyberte vše kromě Network Access Protection > Next. (Obr. 57)

8) Vložte SQL server name > Next > Next > Next > Next > Next. (Obr. 58)



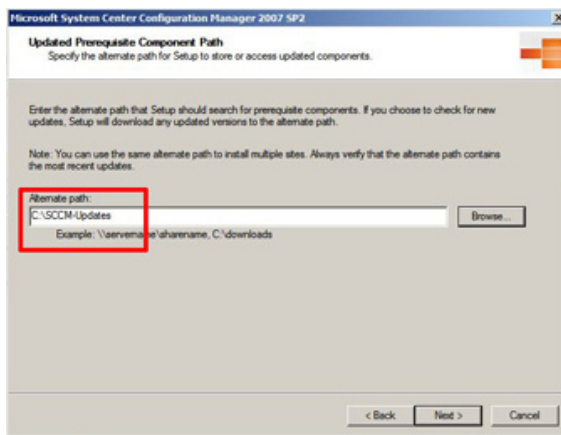
Obr. 57. Client Agent Selection



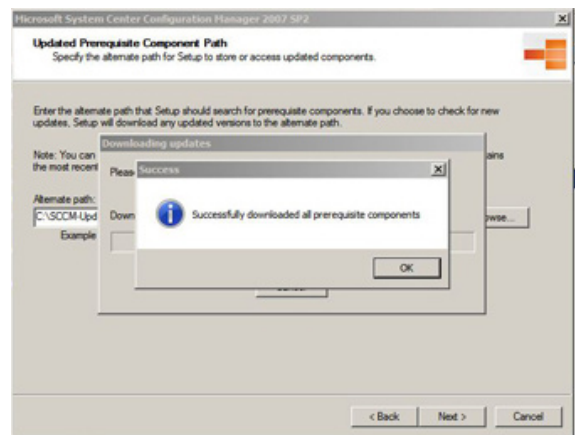
Obr. 58. SQL server name

9) Vyberte adresář, kam se budou stahovat updaty > Next. (Obr. 59)

10) Stáhněte aktualizace > po úspěšném dokončení potvrďte > OK > Next. (Obr. 60)



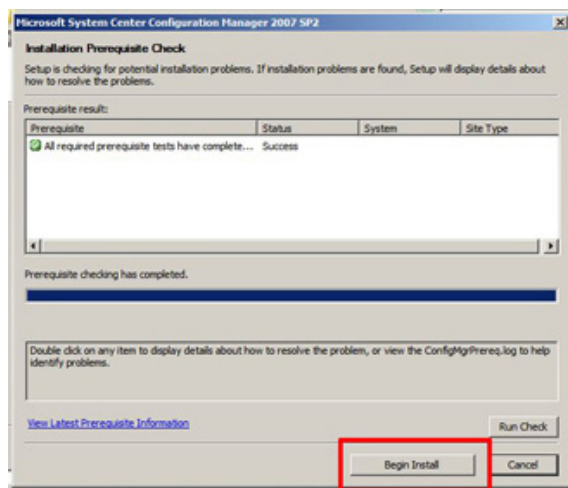
Obr. 59. Update path



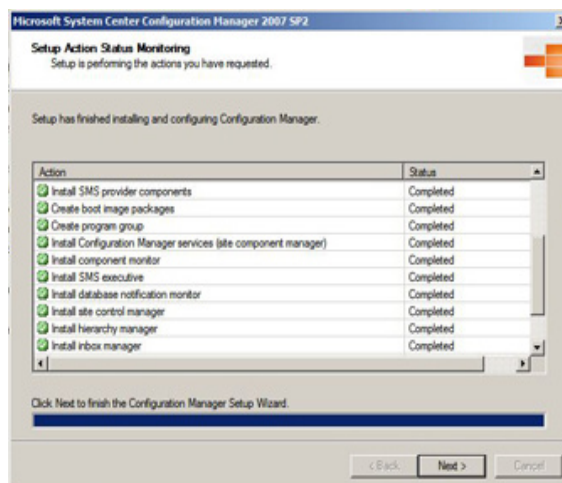
Obr. 60. Úspěšné dokončení akt.

11) Spustí se prerequisite check > po dokončení klikněte na Begin Install. (Obr. 61)

12) Po dokončení instalace klikněte na Next > Finish. (Obr. 62)



Obr. 61. Prerequisite check



Obr. 62. Dokončení inst. SCCM

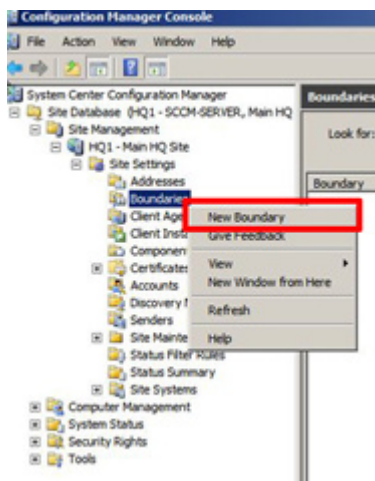
7.3 SCCM 2007 první nastavení a konfigurace

Nyní již máme nainstalovaný SCCM a je potřeba ho nakonfigurovat.

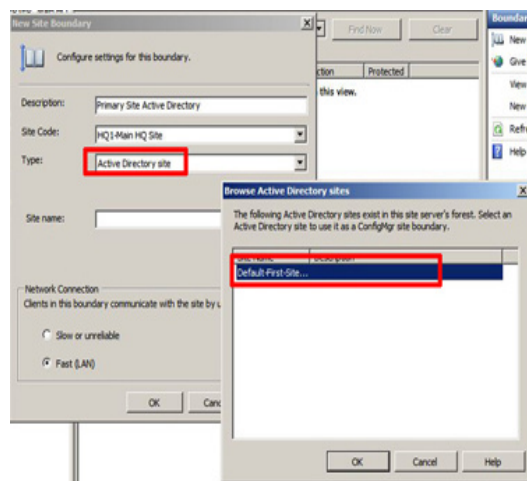
7.3.1 Krok 1: SCCM nastavení Boundary

V první fázi je potřeba nakonfigurovat tzv. Boundary neboli hranice, ve kterých se bude SCCM pohybovat. V našem případě použijeme jako hranici samotnou Active Directory.

- 1) Spusťte Configuration Manager Console > Rozbalte Site Database > Site Management > {Site} > Boundaries > klikněte na > New Boundary. (Obr. 63)
- 2) Dopíšte Descriptive Name > změňte Type na "Active Directory Site" > Browse > Vyberte vaši Active Directory > OK. (Obr. 64)



Obr. 63. Vytvoření Boundary

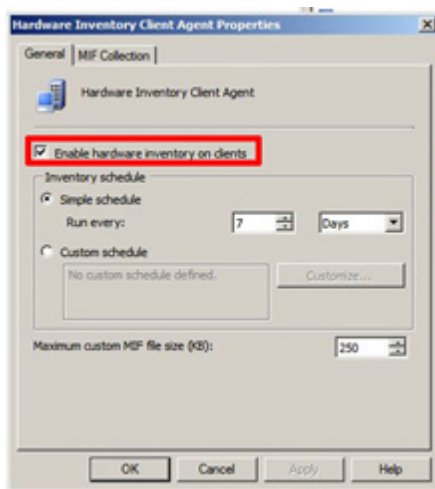


Obr. 64. Vybrání Active Directory

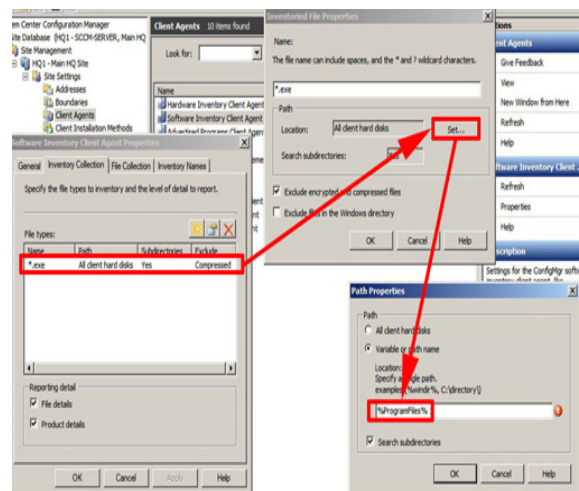
7.3.2 Krok 2: SCCM konfigurace a nasazení agenta

Nyní je potřeba nastavit tzv. agenty. Nastavením těchto agentů se začnou shromažďovat informace o hardware a software vybavení.

- 1) Spusťte Configuration Manager Console > rozklikněte Site Database > Site Management > { Site } > Client Agents > Hardware Inventory Client Agent. (Obr. 65)
- 2) Otevřete Software Inventory Client Agent > Properties > Inventory collection > nastavte vyhledávání *.exe pro všechny klienty. (Obr. 66)

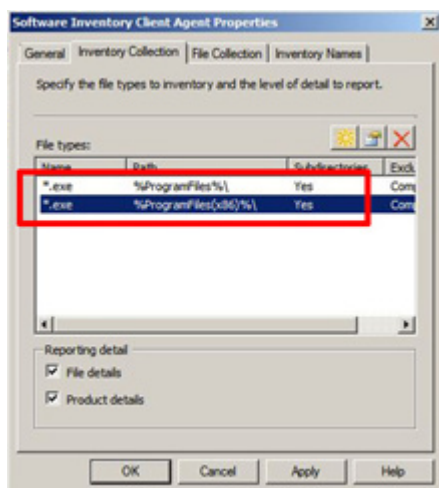


Obr. 65. Hardware Inventory

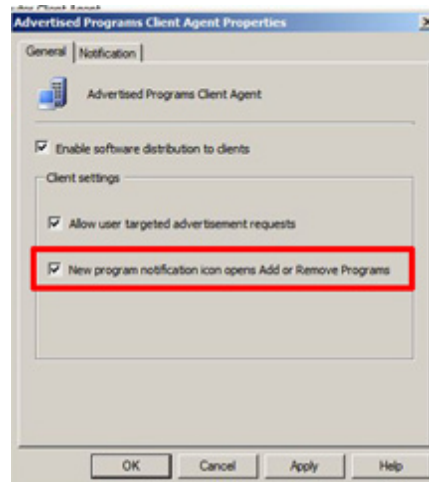


Obr. 66. Nastavení vyhledávání

- 3) Otevřete properties ve "File Types" dialogu > u Location > klikněte na Set > Variable or Path Name > enter %ProgramFiles% > OK > OK > Apply > OK (poznámka na x64 bit nastavte taky %ProgramFiles(x86)%). (Obr. 67)
- 4) Otevřete Advertised Programs Client Agent Properties a zaškrtněte "New program notification icon opens Add Remove Programs" > Apply > OK. (Obr. 68)

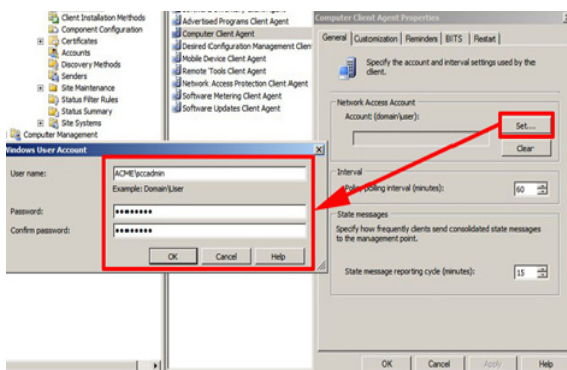


Obr. 67. Typy souborů

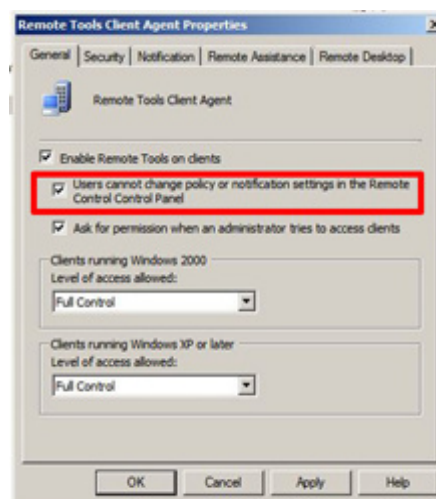


Obr. 68. New program notification

- 5) Otevřete Computer Client Agent Properties > General > Set > zadejte scmadmin hodnoty > OK > Apply > OK. (Obr. 69)
- 6) Otevřete Remote Control Tools Properties > zaškrtněte "Users cannot change policy or notification settings in the Remote Control Panel" > Apply > OK. (Obr. 70)

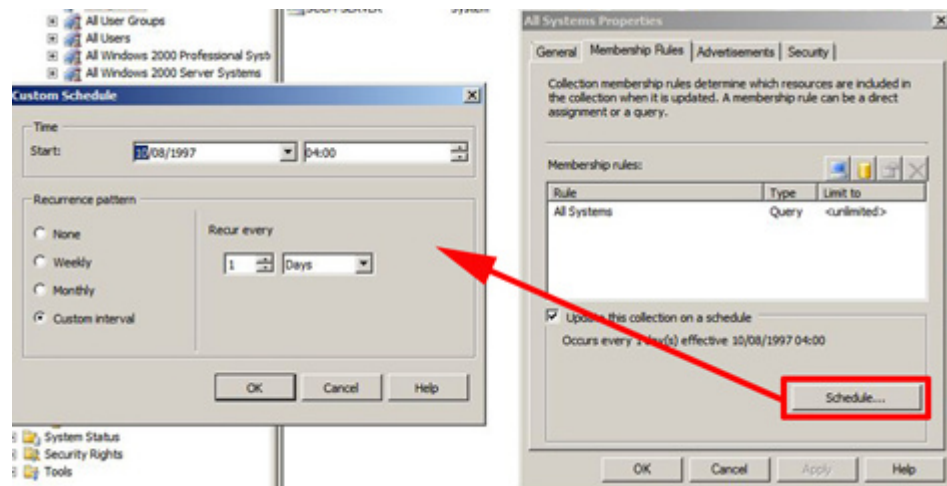


Obr. 69. User Account



Obr. 70. Remote Control Tools

- 7) Nyní je potřeba naplánovat spuštění připravené kolekce, například pokud chceme, aby se vše spustilo ve 4:00 ráno dalšího dne, vybereme All systems > properties > Membership rules > Schedule a nastavíme vlastní interval spuštění. (Obr. 71)



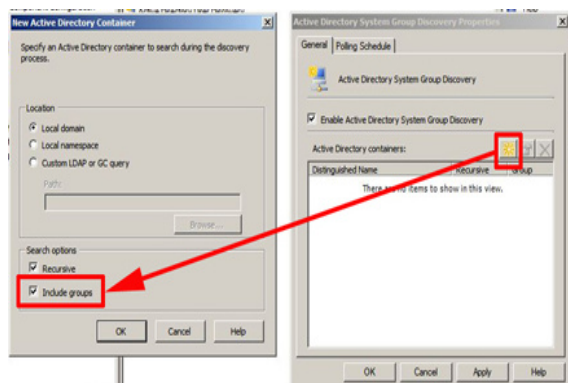
Obr. 71. Plánování agentů

Počkejte, až systém dokončí tvorbu seznamu > Spusťte Configuration Manager Console > Rozklikněte Site Database > Computer Management > Collections > All Systems.

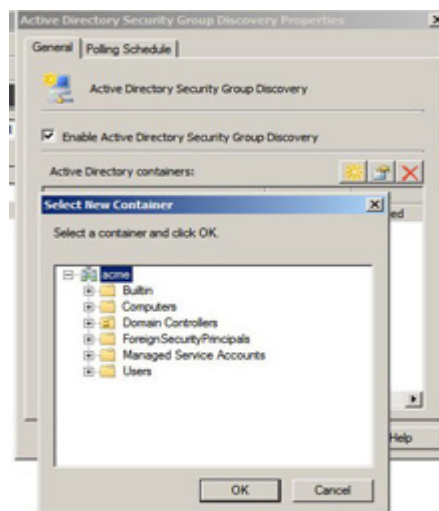
7.3.3 Krok 4: SCCM natavení Discovery methods

SCCM používá Discovery methods k nalezení a identifikování nových i stávajících klientů a uživatelů. Výsledky je poté možné použít při vytváření kolekcí či SQL dotazů za účelem získání určité sady informací.

- 1) Rozklikněte Site management > Site Name > Site Settings > Discovery Methods > klikněte pravým tlačítkem na "Active Directory System Group Discovery" > Properties > Zaškrtněte "Enable Active Directory Group Discovery" > Klikněte na žlutou hvězdičku > Zaškrtněte Include Groups > OK > Vyberte doménu > Klikněte na Polling Schedule > Zaškrtněte "Run Discovery as soon as Possible" > OK > Apply > OK. (Obr. 72)
- 2) Klikněte pravým tlačítkem na "Active Directory Security Group Discovery" > Properties > Zaškrtněte "Enable Active Security Group Discovery" > Klikněte na žlutou hvězdičku > Zaškrtněte Include nested Groups > OK > Vyberte doménu > Klikněte na Polling Schedule > Zaškrtněte "Run Discovery as soon as Possible" > OK > Apply > OK. (Obr. 73)

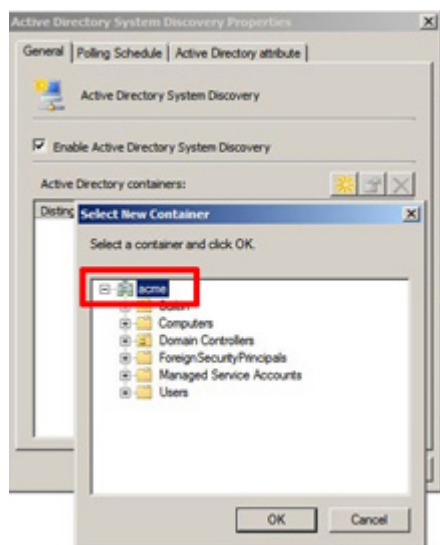


Obr. 72. Povolení Discovery

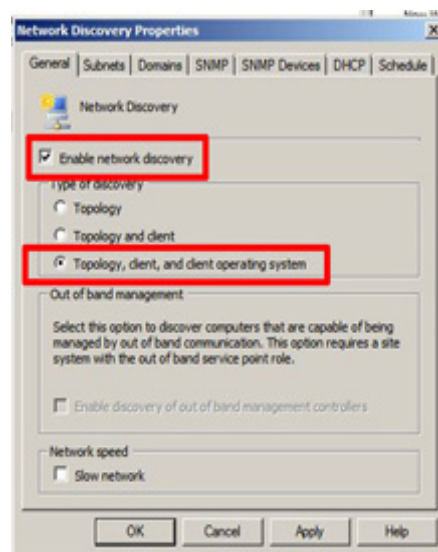


Obr. 73. Security Discovery

- 3) Klikněte pravým tlačítkem na "Active Directory System Discovery" > Properties > Zaškrtněte "Enable Active System Discovery" > Klikněte na žlutou hvězdičku > Zaškrtněte Include Groups > OK > Vyberte doménu > Klikněte na Polling Schedule > Zaškrtněte "Run Discovery as soon as Possible" > OK > Apply > OK. (Obr. 74)
- 4) Opakujte stejný postup jako u bodu 3 i pro "Active Directory User Discovery".
- 5) Klikněte pravým tlačítkem myši na "Network Discovery" Properties, zaškrtněte "Enable net work discovery" > Vyberte Topology, client and client operating system. (Obr. 75)



Obr. 74. System Discovery

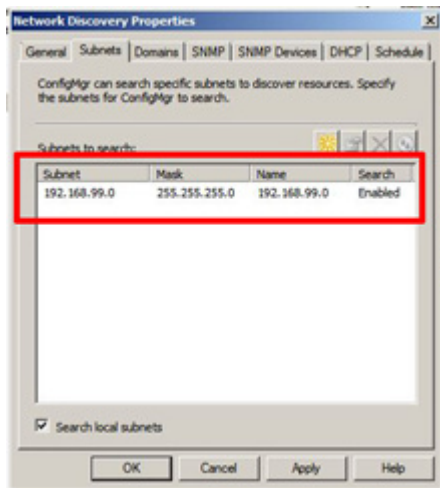


Obr. 75. Enable network disc.

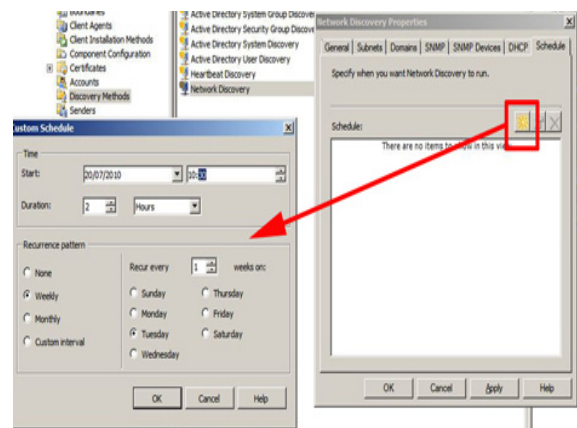
- 6) Vyberte Subnet záložku > zde můžete nastavit vlastní podsítě, ve kterých má probíhat hledání. (Obr. 76).

Poznámka: Můžete také přidat další domény a SNMP zařízení.

- 7) Vyberte záložku Schedule > Klikněte na žlutou hvězdičku > Nastavte plán > OK > Apply > OK. (Obr. 77)



Obr. 76. Subnet

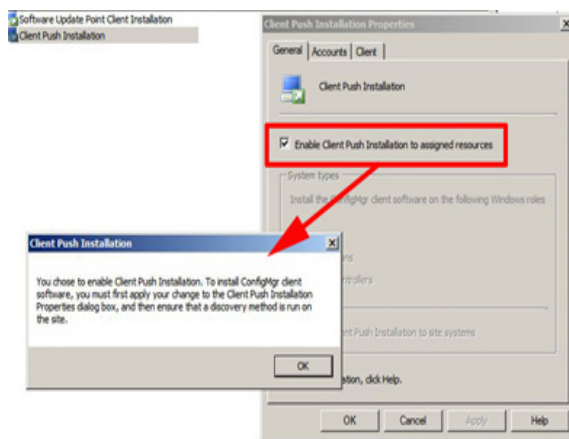


Obr. 77. Nastavení plánování

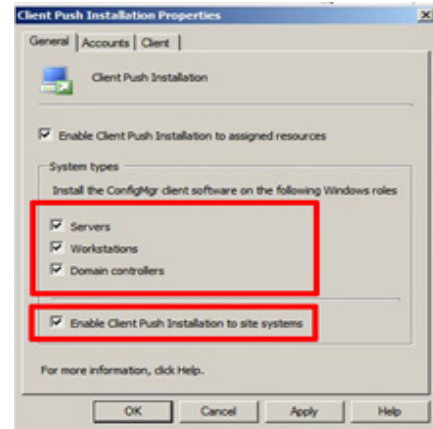
7.3.4 Krok 5: SCCM nastavení instalace klientů

Před samotnou správou klientských stanic je potřeba na ně roz distribuovat klientskou aplikaci. SCCM nabízí dvě varianty. První metoda se nazývá Push Installation, druhá pak probíhá pomocí WSUS role. V tomto případě jsem použil metodu Push Instalation, protože chci nainstalovat klienty na všechny stroje v nadefinované hranici Boundary.

- 1) Rozklikněte Site Database > Site Management > Site name > Site Settings > Client Installation Methods > Client Push Installation > Zaškrtněte "Enable Client Push Installation to Assigned resources" Odklikněte varovnou hlášku. (Obr. 78)
- 2) Zaškrtněte volby podle potřeby. (Obr. 79)

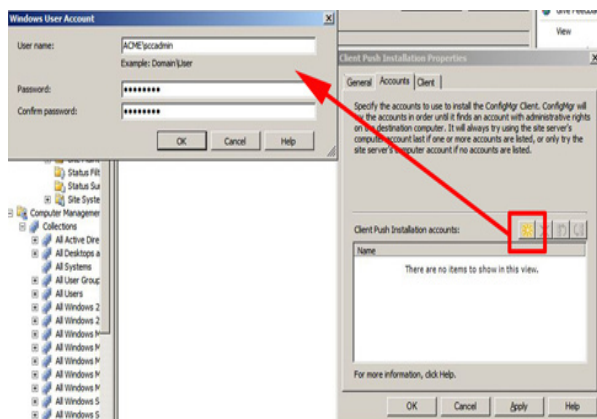


Obr. 78. Client Push Inst. warning

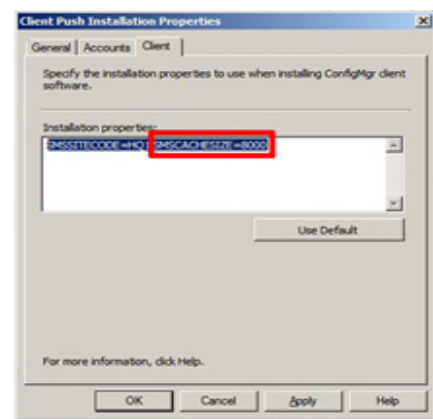


Obr. 79. Volby Client Push Inst.

- 3) Vyberte záložku Accounts a zadejte kontaktní údaje k účtu scmadmin. (Obr. 80)
- 4) Vyberte záložku Client a na konec řetězce přidejte hodnotu SMSCACHESIZE=8000 > Apply > OK. (Obr. 81)

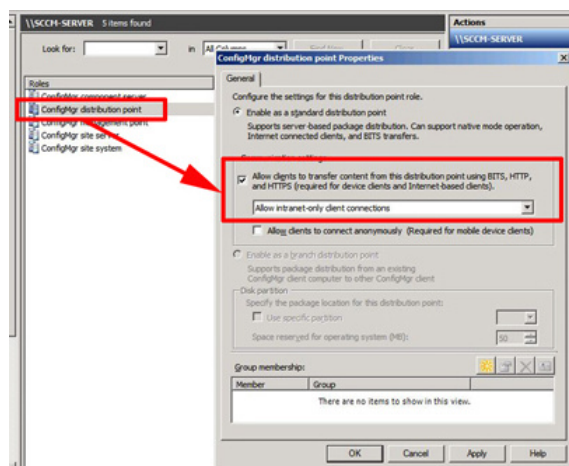


Obr. 80. Kontaktní údaje k účtu

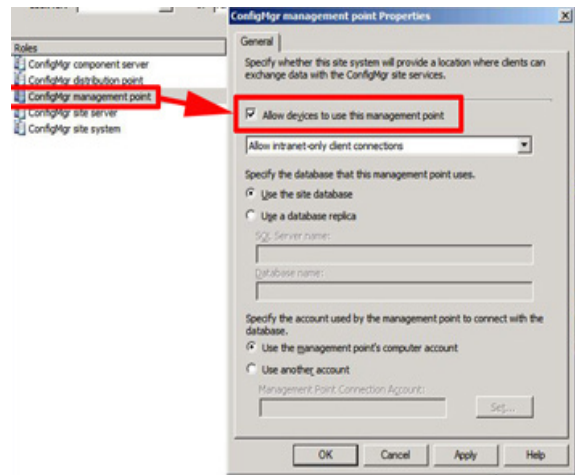


Obr. 81. SMSCACHESIZE

- 5) Rozklikněte Site management > Site Name > Site Settings > Site systems > Vyberte SCCM Server > pravým tlačítkem klikněte na "ConfigMgr distribution point" > Properties > Zaškrtněte "Allow clients to transfer content from this distribution point using BITS, HTTP and HTTPS) > Apply OK. (Obr. 82)
- 6) Klikněte na properties u položky "Configmgr management point" > Zaškrtněte "Allow devices to use this management point" > Apply > OK. (Obr. 83)



Obr. 82. Distribution point

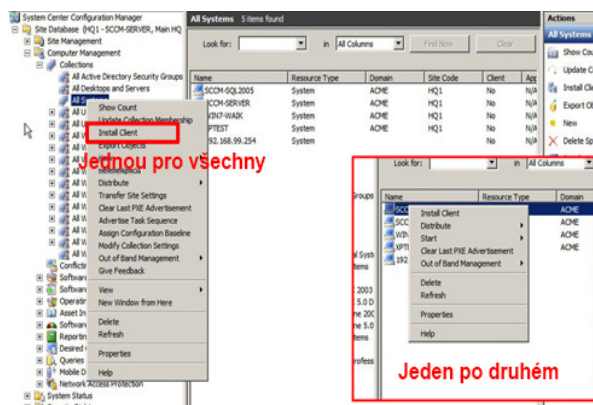


Obr. 83. Configmgr management

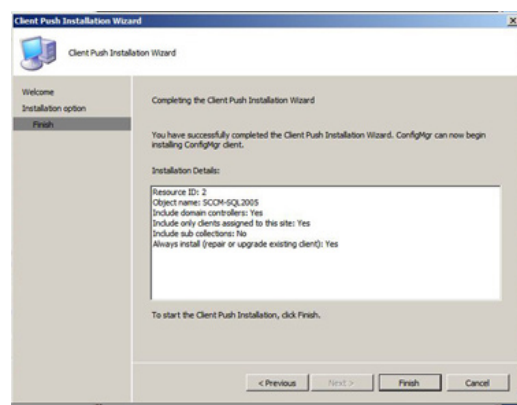
7.3.5 Krok 6: SCCM Instalace klienta

V tomto kroku již spustíme samotnou instalaci klientů na cílové stanice.

- 1) Rozklikněte Site Database > Computer Management > Collections > All systems > klikněte pravým tlačítkem na "All systems" a vyberte Install Client > Nebo můžete instalovat klienty samostatně jeden po druhém. (Obr. 84)
- 2) Na úvodní stránce klikněte na Next > vyberte všechny instalační možnosti > Next > Finish. (Obr. 85)

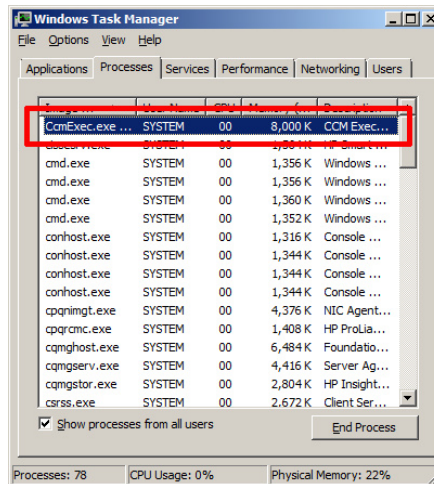


Obr. 84. Instalace klienta



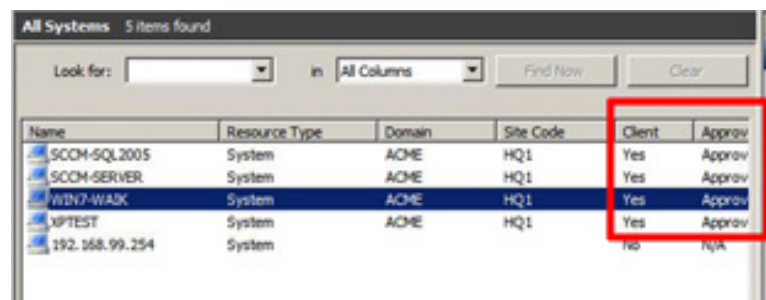
Obr. 85. Dokončení inst. klientů

Během instalačního procesu budete ve správci úloh vidět CcmExec.exe. (Obr. 86)



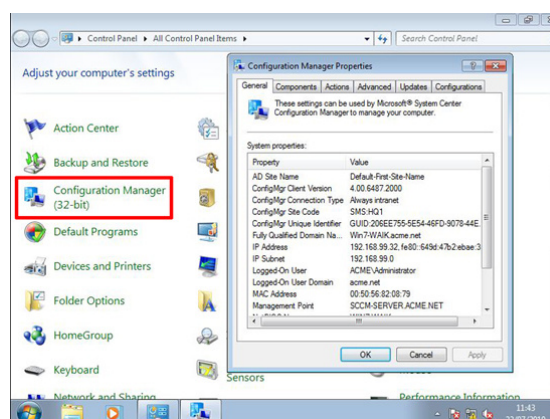
Obr. 86. Instalace CcmExec.exe

- 3) Jakmile se dokončí instalace a „klient“ se přihlásí, bude ve sloupečku Client hodnota „Yes“ (Obr. 87)



Obr. 87. Změna hodnot u klientů

- 4) Na klientských počítačích bude vidět ikona v ovládacích panelech "Configuration Manager". (Obr. 88)



Obr. 88. Klient na stanici v ovládacích panelech

Nyní je vše připraveno k distribuci a správě prvních instalačních balíčků. Což už není zas až tak těžké, protože máme vše již připraveno a nakonfigurováno. V sekci Software

Distribution si vytvoříme nový balíček obdobně jako u instalace klienta tedy kliknutím na New Package a vyplněním informací o balíčku, jako je název, verze a podobně. Ne vše je nutné vyplňovat. Pokračujeme v průvodci, ve kterém zvolíme umístění instalačního balíčku a oprávnění uživatele určeného ke spouštění instalace, většinou Administrátor. Nyní je vytvořen tzv. Package, který je potřeba ještě blíže specifikovat a naplánovat jeho spuštění. Rozbalíme tedy nově vytvořený package a zvolíme položku „Program“ a pod pravým tlačítkem vybereme „New Program“, spustí se průvodce, ve kterém postupně zvolíme parametry spouštění aplikace, jako je například typ spuštění, nebo akce, které se mají provést po úspěšné instalaci. Dále můžeme zvolit omezení, na které operační systémy se má aplikace nainstalovat a zda se má instalace provést v době, kdy je uživatel přihlášen, a další závislosti. Nyní již stačí jen zvolit čas instalace, a to provedeme pomocí průvodce v části „Advertisements“.[17]

ZÁVĚR

Klasická instalace na platformě Windows je relativně křehká záležitost, která vyvolává řadu otázek, jako třeba jestli je možno instalovat více verzí dané aplikace najednou, jestli instalace jedné aplikace nepoškodí instalaci aplikace druhé, jestli mezi nimi neexistuje nějaká závislost, například na úrovni knihoven nebo registrů atd.

Windows jsou díky svému návrhu postiženy nepříjemným problémem zvaným DLL hell. Ten je způsoben tím, že sdílené knihovny aplikací jsou instalovány do společného adresáře, resp. adresářů, takže se může stát, že instalace úplně jiné aplikace může knihovnu přepsat její novější verzí, která nemusí být zpětně kompatibilní. Tyto a další problémy je možné vyřešit právě správně zvolenou technologií tvorby instalačních balíčků.

Většinou každý, kdo se podílí na vývoji, distribuci nebo implementaci software, slyšel o systému Microsoft Windows Installer. Ale proč používat tuto službu? Za prvé Instalační služba systému Windows je součástí operačního systému Windows. Za druhé poskytuje standardní základ pro instalaci a odinstalování softwaru včetně možnosti rollback a za třetí řeší problémy jako DLL hell pomocí správy verzí sdílených knihoven DLL a podporuje aktualizace či opravy.

Ale nejen DLL knihovny jsou problémem klasického způsobu nasazení aplikací. Dalším vážným nedostatkem je náročnost instalace na mnoho klientů. Pokud vezmeme v úvahu korporátní sféru, kde má počítače na starosti IT oddělení, potom instalace nebo jakákoliv aktualizace aplikace bez použití nějakého sofistikovaného softwaru na správu nasazení znamená nutnost fyzicky obejít veškeré počítače a nějaký čas se jim věnovat. To může být u některých aplikací relativně často. Z těchto, ale i dalších důvodů, je vhodné používat pro nasazování aplikací nástroje, jako je například Systém Center Configuration Manager, nebo minimálně distribuovat aplikace například pomocí Group Policy.

Přestože se postupně přechází na možnost instalace například pomocí Windows Store, kde se žádné instalační balíčky manuálně nespouští a vše probíhá automaticky. V korporátní sféře stále není možné nahradit úlohu standardních typů instalací a Windows Installer stále zůstává hlavní technologií umožňující instalaci balíčků na platformě Microsoft Windows.

ZÁVĚR V ANGLIČTINĚ

Classic installation on Windows is a relatively delicate issue that raises many questions, such as whether it is possible to install multiple versions of the application at the same time, whether the installation of one application does not damage the installation of another application, whether there is some dependence between them, for example on the level of libraries or registers, etc.

Windows are due to their design affected by an unpleasant problem called DLL Hell. It is caused by the fact that shared libraries applications are installed into common directory, or directories. Therefore it can happen that an installation of completely different application can overwrite the library with a new version that may not be compatible backwards. These and other problems can be solved only by properly selected technology for production of installation packages.

Most of the people who are involved in the development distribution or implementation of software have heard of the Microsoft Windows Installer system. However, why use this service? Firstly, Windows Installer is part of the Windows operating system. Secondly, it provides a standard basis for installing and uninstalling software, including the possibility of a rollback. Thirdly, it solves the problems like DLL Hell using management versions of shared DLL libraries and it supports updates and patches.

DLL libraries are not the only problem of classical way of applications deployment. Another serious drawback is the installation difficulty for many clients. If we consider the corporate sphere, where the IT department is in charge of the computers, the installation or any update application without using some sophisticated software for managing the deployment means that it is necessary to go to all the computers. It is necessary to devote some time to them which can be with some applications relatively often. For this and other reasons it is appropriate to use tools like System Center Configuration Manager for applications deployment or at least distribute applications, for example via Group Policy.

Therefore as the possibility for installation for example Windows Store is gradually being more used because there are no setup packages started manually and everything happens automatically. In the corporate sector it is still not possible to replace the standard type of installation and Windows Installer still remains a main technology for package installation on Microsoft Windows.

SEZNAM POUŽITÉ LITERATURY

- [1] KADLEC, Václav. Umíme to s Delphi: 84. díl. In: KADLEC, Václav. *Umíme to s Delphi* [online]. 2003 [cit. 2012-08-21]. Dostupné z: <http://www.zive.cz/clanky/umime-to-s-delphi-84-dil--tvorba-instalacnich-baliku/sc-3-a-111500/default.aspx>
- [2] KADLEC, Václav. Umíme to s Delphi: 85. díl. In: KADLEC, Václav. *Umíme to s Delphi* [online]. 2003 [cit. 2012-08-21]. Dostupné z: <http://www.zive.cz/clanky/umime-to-s-delphi-85-dil--tvorba-instalacnich-baliku-prakticky/sc-3-a-111632/default.aspx>
- [3] HOLAN, Jan. WiX. In: HOLAN, Jan. *WiX* [online]. 2011 [cit. 2012-08-21]. Dostupné z: <http://blog.imp.cz/post/2011/04/26/WiX-Jak-na-instalace-cast-1-Uvod>
- [4] Tvoříme instalace k programům v Inno setup compiler. In: [Http://www.devbook.cz](http://www.devbook.cz) [online]. 2010 [cit. 2012-08-21]. Dostupné z: <http://www.devbook.cz/tvorime-instalace-k-programum-v-inno-setup-compiler>
- [5] [Http://installsite.org/](http://installsite.org/) [online]. 1997 [cit. 2012-08-27]. Dostupné z: <http://installsite.org/>
- [6] Microsoft Windows Installer. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012-2-18 [cit. 2012-08-21]. Dostupné z: http://cs.wikipedia.org/wiki/Microsoft_Windows_Installer
- [7] [Http://searchcio-midmarket.techtarget.com](http://searchcio-midmarket.techtarget.com). ROUSE, Margaret. *Http://searchcio-midmarket.techtarget.com* [online]. 2005 [cit. 2012-08-27]. Dostupné z: <http://searchcio-midmarket.techtarget.com/definition/GUID>
- [8] BORUP, Rick. *Understanding Windows Installer*. Great Lakes, 2003, 16 s.
- [9] WILSON, Phil. *The definitive guide to Windows Installer*. New York: Distributed to the Book trade in the United States by Springer-Verlag, c2004, xvii, 301 p. ISBN 15-905-9297-2.

- [10] Hašovací funkce. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012-7-12 [cit. 2012-08-21]. Dostupné z: http://cs.wikipedia.org/wiki/Ha%C5%A1ovac%C3%AD_funkce
- [11] KANISOVÁ, Hana a Miroslav MÜLLER. *UML srozumitelně*. 2. aktualiz. vyd. Brno: Computer Press, 2006, 176 s. ISBN 80-251-1083-4.
- [12] VIRIUS, Miroslav. *C# 2010: hotová řešení*. 1. vyd. Brno: Computer Press, 2012, 424 s. K okamžitému použití (Computer Press). ISBN 978-80-251-3730-7.
- [13] FORD, Sara. *266 tipů a triků pro Microsoft Visual Studio*. Vyd. 1. Brno: Computer Press, 2009, 224 s. ISBN 978-80-251-2554-0.
- [14] YOUNG, Michael J. *XML: krok za krokem*. Vyd. 2. Brno: Computer Press, 2006, 471 s. ISBN 80-251-1070-2.
- [15] PILAŘ, Jan. Instalace Internet Explorer 8 pomocí Group Policy. In: *Zive.cz* [online]. 2010 [cit. 2012-08-22]. Dostupné z: <http://www.zive.cz/clanky/instalace-internet-explorer-8-pomoci-group-policy/sc-3-a-153305/default.aspx>
- [16] www.interval.cz. In: DOLEŽAL, Dušan. *Jak funguje digitální podpis* [online]. 2002 [cit. 2012-08-26]. Dostupné z: <http://interval.cz/clanky/jak-funguje-digitalni-podpis/>
- [17] [Http://www.ahmedgroup.co.uk](http://www.ahmedgroup.co.uk) [online]. 2009-12-29 [cit. 2012-08-22]. Dostupné z: <http://www.ahmedgroup.co.uk/categories/System-Center-Configuration-Manager-2007/>
- [18] [Http://www.it-bloguje.cz](http://www.it-bloguje.cz). In: SOUKUP, Ondřej. *Co jsou skupiny zásad (Group Policy)* [online]. 2009 [cit. 2012-08-24]. Dostupné z: <http://www.it-bloguje.cz/windows-server/active-directory/70-co-jsou-skupiny-zasad-group-policy.html>
- [19] [Http://www.microsoft.com](http://www.microsoft.com). In: [Http://www.microsoft.com](http://www.microsoft.com) [online]. 2012 [cit. 2012-08-24]. Dostupné z: <http://www.microsoft.com/cze/systemcenter/configuration-manager/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|--------|--|
| MSI | Soubor formátu tvoří instalační balíček pro systémy Microsoft Windows ve formátu Windows Installer |
| MSP | Soubor formátu tvoří instalační balíček ve formě patche pro systémy Microsoft Windows ve formátu Windows Installer |
| MST | Soubor formátu tvoří transformační balíček pro systémy Microsoft Windows ve formátu Windows Installer. |
| SCCM | Systém Center Configuration Manager |
| GUID | Globally Unique Identifier |
| WSUS | Microsoft Windows Server Update Services |
| WEBDAV | Web-based Distributed Authoring and Versioning |
| IIS | Internet Information Services |
| SQL | Standardizovaný dotazovací jazyk |
| ODBC | Open Database Connectivity |
| COM | Component Object Model |
| SDK | Software Development Kit |
| DLL | Dynamic-link library, dynamicky linkovaná knihovna |
| IT | Informační technologie |
| GPO | Group Policy |
| UI | user interface |
| ID | Identifikace, Identifikační číslo, označení |
| MSM | Soubor formátu merge modul pro Windows Installer |

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obr. 1. Tabulka Property..... | 20 |
| Obr. 2. Tabulka Feature | 21 |
| Obr. 3. Tabulka Component | 22 |
| Obr. 4. Tabulka File | 22 |
| Obr. 5. Diagram postupu vývoje..... | 34 |
| Obr. 6. Vygenerované xml..... | 34 |
| Obr. 7. Kontrola integrity..... | 35 |
| Obr. 8. Installation Information | 37 |
| Obr. 9. Features..... | 38 |
| Obr. 10. Wizard Destination | 39 |
| Obr. 11. Wizard files..... | 39 |
| Obr. 12. Registry | 40 |
| Obr. 13. Služby | 41 |
| Obr. 14. ODBC | 42 |
| Obr. 15. Shortcuts | 43 |
| Obr. 16. Management konzole..... | 50 |
| Obr. 17. Editor zásad skupin..... | 50 |
| Obr. 18. Vypnutí části GPO | 51 |
| Obr. 19. Active Directory Users | 54 |
| Obr. 20. Domain Admins settings | 54 |
| Obr. 21. Uživatelé ve skupině..... | 54 |
| Obr. 22. Přidání uživatelů do skupin | 55 |
| Obr. 23. Přidejte IIS | 55 |
| Obr. 24. Přidání rolí 1/2 | 56 |
| Obr. 25. Přidání rolí do 2/2 | 56 |
| Obr. 26. Přidání serverových funkcí | 56 |
| Obr. 27. Nainstalování WebDAV..... | 57 |
| Obr. 28. Povolení WebDAV | 57 |
| Obr. 29. Enable WebDAV | 57 |
| Obr. 30. WebDAV Authoring Rule | 57 |
| Obr. 31. Odkaz WebDAV settings | 58 |

| | |
|--|----|
| Obr. 32. WebDAV settings | 58 |
| Obr. 33. WSUS 3.0 | 59 |
| Obr. 34. WSUS Server Update | 59 |
| Obr. 35. Instalace SQL Server | 59 |
| Obr. 36. Product key | 59 |
| Obr. 37. Database Management..... | 60 |
| Obr. 38. Use the same account..... | 60 |
| Obr. 39. Windows Authentication | 60 |
| Obr. 40. Konec instalace | 60 |
| Obr. 41. ADSI Edit | 61 |
| Obr. 42. ADSI Edit nový objekt | 61 |
| Obr. 43. System Management..... | 62 |
| Obr. 44. Advanced view | 62 |
| Obr. 45. Security | 62 |
| Obr. 46. Přidání skupin | 63 |
| Obr. 47. Oprávnění | 63 |
| Obr. 48. Editace skupiny SCCM-ADMIN..... | 63 |
| Obr. 49. This object | 64 |
| Obr. 50. SCCM-Server | 64 |
| Obr. 51. SCCM instalace | 64 |
| Obr. 52. Vypnutí firewallu..... | 64 |
| Obr. 53. Správné dokončení..... | 65 |
| Obr. 54. Install Manager site..... | 65 |
| Obr. 55. Site code a Site name..... | 65 |
| Obr. 56. Conf. Mixed Mode | 65 |
| Obr. 57. Client Agent Selection..... | 66 |
| Obr. 58. SQL server name | 66 |
| Obr. 59. Update path..... | 66 |
| Obr. 60. Úspěšné dokončení akt. | 66 |
| Obr. 61. Prerequisite check | 67 |
| Obr. 62. Dokončení inst. SCCM..... | 67 |
| Obr. 63. Vytvoření Boundary..... | 67 |
| Obr. 64. Vybrání Active Directory | 67 |

| | |
|--|----|
| Obr. 65. Hardware Inventory | 68 |
| Obr. 66. Nastavení vyhledávání | 68 |
| Obr. 67. Typy souborů | 69 |
| Obr. 68. New program notification | 69 |
| Obr. 69. User Account | 69 |
| Obr. 70. Remote Control Tools | 69 |
| Obr. 71. Plánování agentů | 70 |
| Obr. 72. Povolení Discovery | 71 |
| Obr. 73. Security Discovery | 71 |
| Obr. 74. System Discovery | 71 |
| Obr. 75. Enable network disc | 71 |
| Obr. 76. Subnet | 72 |
| Obr. 77. Nastavení plánování | 72 |
| Obr. 78. Client Push Inst. warning | 73 |
| Obr. 79. Volby Client Push Inst. | 73 |
| Obr. 80. Kontaktní údaje k účtu | 73 |
| Obr. 81. SMSCACHESIZE | 73 |
| Obr. 82. Distribution point | 74 |
| Obr. 83. Configmgr management | 74 |
| Obr. 84. Instalace klienta | 74 |
| Obr. 85. Dokončení inst. klientů | 74 |
| Obr. 86. Instalace CcmExec.exe | 75 |
| Obr. 87. Změna hodnot u klientů | 75 |
| Obr. 88. Klient na stanici v ovládacích panelech | 75 |

SEZNAM TABULEK

| | |
|--|----|
| Tab. 1. Historie verzí 1/2 | 16 |
| Tab. 2. Historie verzí 2/2 | 17 |
| Tab. 3. Možnosti bezobslužné instalace | 27 |
| Tab. 4. Small, minor, major upgrades..... | 28 |

SEZNAM PŘÍLOH

PŘÍLOHA. P I. Program pro kontrolu integrity vstupních souborů

PŘÍLOHA P I: PROGRAM PRO KONTROLU INTEGRITY VSTUPNÍCH SOUBORŮ

```
using System;
using System.IO;
using System.Xml.Linq;
using System.Linq;

namespace IntegrityCheck
{
    /// <summary>
    /// Program ma dvě hlavní funkce
    /// 1. Vygenerovat z adresáře xml dokument s hashema pro následnou
    kontrolu integrity
    /// 2. Porovnat aktuální obsah adresáře s dříve vytvořeným xml
    dokumentem obsahujícím hashe souboru
    /// </summary>
    class Program
    {
        static int Main(string[] args)
        {
            if (args.Length == 0)
            {
                Console.WriteLine("Nástroj pro kontrolu integrity
adresáře.");
                Console.WriteLine("");
                Console.WriteLine("Vygenerování klíče pro pozdější
kontrolu integrity: ");
                Console.WriteLine("-generateKey [DirectoryPath]");
                Console.WriteLine("");
                Console.WriteLine("Kontrola itegrity adresáře s použitím
již vygenerovaného klíče: ");
                Console.WriteLine("-checkIntegrity [DirectoryPath]
[IntegrityKeyPath]");

                return 0;
            }

            var command = args[0];

            if (command == "-generateKey")
            {
                try
                {
                    return GenerateKey(args);
                }
                catch(Exception e)
                {
                    Console.WriteLine("Chyba - nastala chyba při
generování klíče");
                    Console.WriteLine(e);
                    return -1;
                }
            }

            if (command == "-checkIntegrity")
            {
                try
                {
                    return CheckIntegrity(args);
                }
                catch (Exception e)
                {

```

```

        Console.WriteLine("Chyba - nastala chyba při kontrole
integrity");
        Console.WriteLine(e);
        return -1;
    }
}

Console.WriteLine("Chyba ve vstupních argumentech.");
return -1;
}

/// <summary>
/// Zkontroluje integritu, porovná dříve vygenerovaný hash
dokument a aktuální stav adresářů.
/// </summary>
private static int CheckIntegrity(string[] args)
{
    //kontrola vstupních argumentů....
    if (args.Length != 3)
    {
        Console.WriteLine("Chyba - nesprávný počet parametrů.");
        return -1;
    }

    if (!CheckDirectoryExists(args[1]) ||
!CheckFileExist(args[2]))
        return -1;

    //nechám si vytvořit hash dokument pro aktuální obsah
adresáře
    var actualKeyDocument = new
IntegrityCheckService().CreateKeyDocument(args[1]);

    //načtu hash dokument s původním obsahem adresáře
    var oldKeyDocument = XDocument.Load(args[2]);

    if (actualKeyDocument.ToString() !=
oldKeyDocument.ToString())
    {
        actualKeyDocument.Save("ActualIntegrityKeyDocument.xml");

        Console.WriteLine("V adresáři došlo ke změně. ");
        Console.WriteLine("");
        Console.WriteLine("Pro pozdější analýzu konkrétních změn
porovnejte obsahy souborů {0} a nově vygenerovaného
ActualIntegrityKeyDocument.xml", args[2]);
        Console.WriteLine("");
        Console.WriteLine("Seznam změn");
        Console.WriteLine("");

        var intersectOfHashes =
oldKeyDocument.Descendants("fileHash").Intersect(actualKeyDocument.Descen
dants("fileHash"), new HashAndPathEqualityComparer()).ToList();

        //vypis aktuálních souboru, které se liší oproti původním
foreach (var actualFile in
actualKeyDocument.Descendants("fileHash").Where(f =>
!intersectOfHashes.Contains(f, new HashAndPathEqualityComparer())))
        {
            Console.WriteLine("Změna v souboru {0}.",
actualFile.Attribute("path").Value);

```

```

        var oldFile =
oldKeyDocument.Descendants("fileHash").Where(f =>
f.Attribute("path").Value == actualFile.Attribute("path").Value)
        .FirstOrDefault();
        if (oldFile != null &&
actualFile.Attributes("version").Any() &&
oldFile.Attributes("version").Any())
        {
            Console.WriteLine("Verze souborů si neodpovídají.
Původní: {0} Nová: {1}", oldFile.Attributes("version").First().Value,
actualFile.Attributes("version").First().Value);
        }
        //výpis původních souborů, které se liší cestou
        foreach (var oldFile in
oldKeyDocument.Descendants("fileHash").Where(f =>
!actualKeyDocument.Descendants("fileHash").Contains(f, new
PathEqualityComparer())))
        {
            Console.WriteLine("Změna v souboru {0}.",
oldFile.Attribute("path").Value);
        }
        var actualFile =
actualKeyDocument.Descendants("fileHash").Where(f =>
f.Attribute("path").Value == oldFile.Attribute("path").Value)
        .FirstOrDefault();
        if (actualFile != null &&
oldFile.Attributes("version").Any() &&
actualFile.Attributes("version").Any())
        {
            Console.WriteLine("Verze souborů si neodpovídají.
Původní: {0} Nová: {1}", oldFile.Attributes("version").First().Value,
actualFile.Attributes("version").First().Value);
        }
        return -1;
    }

    Console.WriteLine("V adresáři nedošlo k žádným změnám.");
    return 0;
}

/// <summary>
/// Vygeneruje dokument s klíčem (hashema)
/// </summary>
private static int GenerateKey(string[] args)
{
    if (args.Length != 2)
    {
        Console.WriteLine("Chyba - nesprávný počet parametrů.");
        return -1;
    }

    if (!CheckDirectoryExists(args[1]))
        return -1;

    var keyDocument = new
IntegrityCheckService().CreateKeyDocument(args[1]);
    keyDocument.Save("IntegrityCheckKey.xml");
}

```

```

        Console.WriteLine("Klíč byl vygenerován do souboru
IntegrityCheckKey.xml, ponechte si jej pro pozdější kontrolu
integrity");
        return 0;
    }

    /// <summary>
    /// Zkontroluje zda existuje adresar
    /// </summary>
    private static bool CheckDirectoryExists(string dirPath)
    {
        if (!Directory.Exists(dirPath))
        {
            Console.WriteLine("Chyba - adresář {0} nebyl nalezen.",
dirPath);
            return false;
        }

        return true;
    }

    /// <summary>
    /// Zkontroluje zda soubor existuje
    /// </summary>
    private static bool CheckFileExist(string filePath)
    {
        if (!File.Exists(filePath))
        {
            Console.WriteLine("Chyba - soubor {0} nebyl nalezen.",
filePath);
            return false;
        }

        return true;
    }
}
}
}

```