

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Jamšek

**Namestitev programske opreme
s poudarkom na tehnologiji MSI**

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: izr. prof. dr. Marko Bajec

Ljubljana, 2011

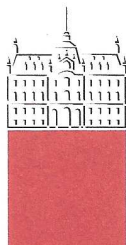
Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00058/2010

Datum: 06.12.2010



Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANDREJ JAMŠEK**

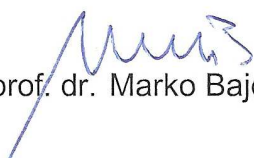
Naslov: **NAMESTITEV PROGRAMSKE OPREME S Poudarkom NA MSI
TEHNOLOGIJI
SOFTWARE DEPLOYMENT USING MSI TECHNOLOGY**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V okviru diplomske naloge preučite tehnologijo MSI, ki omogoča nameščanje programske opreme. Postopek namestitve najprej opišite v splošnem, potem pa obrazložite, kako korake splošnega postopka podpremo z MSI tehnologijo. Za korake, ki jih MSI tehnologija ne podpre, najdete ustrezne odprtokodne rešitve. Nalogo zaključite s primerom.

Mentor:


prof. dr. Marko Bajec

Dekan:


prof. dr. Nikolaj Zimic



Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Andrej Jamšek,

z vpisno številko 63080391,

sem avtor diplomskega dela z naslovom:

Namestitev programske opreme s poudarkom na
tehnologiji MSI

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Marko Bajec
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne

Podpis avtorja:

Zahvala:

Zahvaljujem se podjetju, kjer sem zaposlen, da mi je omogočilo študij, in svoji družini, ki me je pri tem vztrajno spodbujala.

Kazalo

1	Uvod.....	1
2	Programska oprema.....	2
2.1	Sestava programske opreme	2
2.1.1	Vrste datotek.....	2
2.1.2	Računalniške platforme in operacijski sistemi	3
2.2	Namestitev programske opreme	3
2.2.1	Načini namestitev	3
2.2.2	Internacionalizacija namestitvenega paketa	4
2.2.3	Odstranitev programske opreme.....	4
2.2.4	Vrste namestitvenih paketov	4
2.3	Namestitev kot proces.....	5
2.3.1	Skrbstvo nad datotečnim sistemom in pravice uporabe	5
2.3.2	Podatki za nadgraditev in odstranitev.....	5
2.3.3	Kaj če – obravnavanje napak pri instalaciji.....	6
2.3.4	Beleženje poteka dela	6
2.4	Orodja za izdelavo programskih paketov	6
3	Tehnologija MSI	8
3.1	Programska oprema tehnologije MSI	9
3.1.1	Namestitev kot storitev	9
3.1.2	Programske knjižnice in programski dodatki	9
3.2	MSIEXEC	10
3.2.1	Ravnanje ob napakah.....	10
3.2.2	Podatkovna baza namestitev.....	10
3.2.3	Načini delovanja MSIEXEC programa	11
3.2.4	Beleženje poteka namestitve	13
3.3	Potek namestitve	13
3.3.1	Zbiranje podatkov za namestitev – zaporedje akcij " <i>InstallUT</i> "	14
3.3.2	Graditev skripte za transakcijo - zaporedje akcij " <i>InstallExecute</i> " 1. etapa	15
3.3.3	Izvajanje transakcije - zaporedje akcij " <i>InstallExecute</i> " 2. etapa.....	15
3.3.4	Izvajanje zaporedja akcij " <i>InstallExecute</i> " nadaljevanje.....	15
3.4	Namestitveni paketi	15

3.4.1	Izdelava namestitvenega paketa.....	16
3.4.2	Orodja za izdelavo paketa .msi	16
3.4.3	Organizacija paketa.....	17
3.4.4	Tabele, namenjene shranjevanju datotek	18
3.4.5	Tabele za krmiljenje namestitve	20
3.4.6	Validacija – ugotavljanje pravilnosti namestitvenega paketa.....	22
3.4.7	Posebne oblike namestitvenih datotek	22
3.4.8	Lokalizacija paketa	22
3.4.9	Obravnavanje napak.....	23
3.4.10	Uporabna orodja.....	23
4	WiX – odprtokodna rešitev tehnologije MSI.....	26
4.1	Način graditve paketa.....	26
4.1.1	Prevajanje (v vmesno obliko) – orodje Candle.exe	27
4.1.2	Povezovanje	27
4.1.3	Ostala orodja WiX-a	28
4.2	Izvorna koda za graditev namestitvenega paketa.....	28
4.2.1	Format XML	28
4.2.2	Vrste datotek izvorne kode WiX.....	29
4.2.3	Direktive za predprocesno graditev	30
4.2.4	Lokalizacija.....	31
4.3	Dodatki (angl. Extensions).....	32
4.3.1	Uporabniški vmesnik	32
4.3.2	WiX utils.....	33
4.3.3	Namestitev gonilnikov	33
4.3.4	Netfx Shema.....	34
4.4	Orodja za izdelavo instalcijskega paketa MSI s pomočjo WiX-a	35
4.4.1	MS Visual Studio	35
4.4.2	WiXEdit	35
4.4.3	Ostala orodja	35
4.5	Primer uporabe	36
4.5.1	Graditev namestitvenega paketa	37
4.5.2	Opis posebnosti	37
4.6	Primerjava WiX z ostalimi orodji	39

5	Zaključek.....	41
	Reference.....	42
	Seznam slik.....	42
	Seznam tabel.....	42
	Literatura.....	42
	Viri.....	43

Seznam uporabljenih kratic in simbolov

Kratica, simbol	Pomen
MSI	M icrosoft I nstaller – namestitvena tehnologija podjetja Microsoft
MSIEXEC	Programski namestitveni program msiexec.exe
GUID	Angl. G lobally U nique I dentifier: globalno edinstvena identifikacijska številka
WiX	W indows I nstaller X ML
XML	Angl. e Xtensible M arkup L anguage: razširljiv označevalni jezik
VBS	V isual B asic S cripting: skriptni jezik podjetja Microsoft.
SDK	Angl.: S oftware D evelopment K it: zbirka orodij in programskih knjižnic za razvoj aplikacij operacijskega sistema.
WYSIWYG	Angl.: W hat Y ou S ee I s W hat Y ou G et: dobiš kar vidiš
HTML	Angl. H yper T ext M arkup L anguage: označevalni jezik za oblikovanje večpredstavnostnih dokumentov
IDE	Angl. I ntegrated D evelopment E nvironment: integrirano razvojno okolje

Povzetek

Diplomska naloga opisuje celoten proces namestitve izdelane programske opreme na ciljni računalnik. Še posebej se dotakne namestitev na operacijskih sistemih podjetja Microsoft, ki je razvil sorazmerno specifično vrsto namestitev, imenovano tehnologija MSI. Ta določa namestitve, odstranitve, spremembe namestitev in nadgradnje programske opreme vseh vrst. Tehnologija MSI določa le način zapakiranja namestitve, vse ostalo pa kaže na odprtost tehnologije MSI, saj se lahko celotno namestitev sprogramira na svoj način.

Kot primer je v nalogi navedena uporaba odprtokodnega orodja WiX s kratkim primerom namestitve.

Ključne besede:

Namestitev, MSI, WiX, XML, izvršne datoteke, računalniško okolje, operacijski sistem, prevajanje, graditev.

Abstract

A dissertation is describing entire deployment process of developed software on a target computer. It particular touches installations on operating systems of Microsoft Inc. company, that developed some kind of specific deployment technology called MSI technology.

The technology determines installation, removals, changes of installation mode and upgrades of software of all kinds. MSI technology determines only how packages needs to be packed, which shows that MSI technology is entirely opened to developers.

For use case in this dissertation exposes one little example which uses an open-source tool WiX.

Key words:

Deployment, installation, MSI, WiX, XML, executable files, computer environmet, operating system, compiling, building.

1 Uvod

Opravila, ki se izvajajo v računalniških sistemih, zahtevajo različno programsko opremo. Da programska oprema pravilno deluje, mora biti prilagojena strojni opremi računalnika, ki pa je lahko od sistema do sistema zelo različna. Ker bi bilo praktično nemogoče proizvajati programsko opremo za vsak računalniški sistem posebej, se izdeluje programska oprema, ki se jo do neke mere sorazmerno enostavno prilagodi računalniškemu sistemu. Programska oprema mora biti uporabniku kar se da prijazna, kar v osnovi pomeni, da bo uporabnik – kupec z njo zadovoljen, če mu bo nudila tisto, kar potrebuje. Da se programsko opremo lahko začne uporabljati, je potrebna:

- namestitev programske opreme na ciljni računalniški sistem in
- prilagoditev programske opreme ciljnemu računalniškemu sistemu.

Ta dva koraka uporaba programske opreme ne potrebuje, ampak sta nujna koraka le na začetku, da se jo sploh lahko začne uporabljati. Torej predstavljata neke vrste dodatno delo. Za čim večjo poenostavitve in skrajšanje te uvodne faze uporabe programske opreme se programska oprema spravi – zapakira v tako imenovane namestitvene pakete. Ti znajo namestiti programsko opremo z najmanjšo potrebo po računalniškem znanju in tako postane programska oprema na voljo večjemu krogu uporabnikov.

Diplomska naloga v nadaljevanju opisuje začetno fazo uporabe programske opreme – namestitev programske opreme, da lahko uporabnik začne uporabljati vse njene funkcije.

V prvem delu naloga opisuje namestitev na splošno, kot potrebo po nameščanju datotek programske opreme in prilagajanja programske opreme ciljnemu računalniku. Drugi del naloge je osredotočen na eno vrsto namestitvene tehnologije, ki se izvaja na operacijskih sistemih podjetja Microsoft Inc.[®], tako imenovano tehnologijo MSI. V zadnjem delu je opisan primer izdelave namestitvenega paketa v tehnologiji MSI s pomočjo odprtokodnega orodja WiX s praktičnim primerom uporabe.

2 Programska oprema

2.1 Sestava programske opreme

Programsko opremo običajno sestavlja več različnih datotek in nastavitvev, kar je odvisno od arhitekture programske opreme, od skupine, ki je to opremo razvila in naredila, ter od uporabniškega računalniškega sistema, na katerem se uporablja ta programska oprema.

2.1.1 Vrste datotek

Razvojna skupina lahko izdelava različne vrste datotek, ki so potrebne za delovanje programske opreme. Glavne skupine vrst datotek:

- izvršne datoteke: kamor spadajo navadne izvršne datoteke, sistemske izvršne datoteke in storitvene izvršne datoteke;
- programske knjižnice: bodisi enoupodobne (uporabljajo se izključno za eno programsko opremo) in knjižnice, namenjene skupni rabi;
- programski dodatki – vtičniki: to so razni programski bloki, ki se vključujejo v obstoječe izvršne datoteke ali programske knjižnice;
- podatkovne datoteke, v katerih se shranjujejo podatki. To so lahko čisto navadne tekstovne datoteke, datoteke slikovne in multimedijske vsebine, datoteke podatkovnih baz ipd.;
- nastavitvene in ostale datoteke, kamor spadajo vse datoteke, v katerih se shranjujejo nastavitve programske opreme, razne informacijske datoteke (npr. datoteke pomoči, ipd.).

Izvršne datoteke v večji meri proizvajalec programske opreme sam zgradi iz izvorne kode s pomočjo orodij za graditev teh vrst datotek (prevajalnik, povezovalnik ipd.).

Glede na vrsto uporabe lahko razdelimo izvršne datoteke v tri skupine:

- navadne izvršne datoteke – uporabniški programi;
- storitveni programi, ki se izvajajo v ozadju in so namenjeni upravljanju uporabniških programov (primer je storitev, ki skrbi, da je računalniška ura sinhronizirana z omrežjem);
- sistemske izvršne datoteke so datoteke, ki so potrebne za pravilno delovanje sistema oziroma za nadgraditev jedra računalniškega sistema (primer je programski filter datotečnega sistema, ki upravlja z velikostjo prostega prostora na disku).

Proizvajalec lahko pri izdelavi svoje programske opreme uporabi tudi vse vrste datotek drugih izdelovalcev. Običajno so to razne programske knjižnice, programski dodatki, posebne informacijske datoteke ipd. Tudi te datoteke se morajo nahajati na uporabnikovem računalniškem sistemu, da programska oprema lahko deluje.

Ko se soočimo s prenosom programske opreme na drug računalnik, naletimo na problem prilagajanja programske opreme ciljnemu računalniškemu sistemu. Vsak uporabnik programske opreme ima svoj specifični računalniški sistem, ki se lahko popolnoma razlikuje od sistemov drugih uporabnikov.

V ta namen razvijalci programske opreme uporabijo nastavitvene datoteke, ki vsebujejo informacije namenjene prilagoditvi računalniške opreme na uporabnikov računalniški sistem.

Poleg informacij o strojni opremi uporabnikovega računalniškega sistema lahko računalniški program potrebuje tudi druge informacije namenjene prilagoditvi uporabniških lastnosti in navad programski opremi.

2.1.2 Računalniške platforme in operacijski sistemi

Na tržišču so različne platforme, na katerih se poganja različne operacijske sisteme: na Intelovih platformah se lahko poganja operacijske sisteme podjetja Microsoft, razne odprtokodne operacijske sisteme Linux ipd., na platformah podjetja Apple operacijski sistem MAC ipd.

Vsak sistem zahteva izvršne datoteke, ki se izvajajo izključno na njem. Zato na primer program, ki je grajen za operacijski sistem Linux, ne moremo pognati na Microsoftovih operacijskih sistemih.

Pomembna je tudi velikost osnovnega vodila procesorja. Trenutno računalniški sistemi uporabljajo predvsem procesorje, ki podpirajo 32- in 64-bitna vodila, 16-bitna pa počasi tonejo v pozabo. Običajno operacijski sistemi podpirajo izvršne datoteke narejene za manjša vodila, vendar to ni vedno tako.

2.2 Namestitev programske opreme

Namestitev računalniške programske opreme je prenos oziroma postavitve proizvajalčeve programske opreme na uporabnikov računalnik. Običajno je to uporabnikov prvi stik s programsko opremo, zato mora ta pustiti uporabniku čim boljši vtis.

Namestitev kot proces je lahko le enostavno kopiranje datotek ali bolj zapletene aktivnosti, ki nastanejo zaradi prilagajanja proizvajalčeve računalniške opreme uporabnikovemu računalniškemu sistemu.

Zaradi lažje dostave v računalniški sistem se celotno programsko opremo "zapakira" v tako imenovani namestitveni paket, ki postori vse potrebno za začetek uporabe programske opreme. Namestitveni paket se zgradi za eno platformo in en (v nekaterih primerih skupino) operacijski sistem.

2.2.1 Načini namestitev

K namestitvi programske opreme se lahko šteje:

- namestitev uporabniške programske opreme. Poznamo tako imenovano tiho namestitev, kjer ni potrebno posredovanje uporabnika za namestitev, in običajno namestitev;
- nadgraditev, če je računalniška oprema že nameščena in se jo želi le posodobiti z novimi funkcionalnostmi oziroma izboljšavami oziroma s popravki;
- namestitev operacijskega sistema. To je posebna vrsta namestitvenega paketa, ki se uporablja izključno za namestitev operacijskih sistemov;
- namestitev sistemskih programov. Običajno je namestitev sistemskih programov vključena kar v običajni paket, lahko pa nastopa tudi posamezno.

Glede na lokacijo namestitvenega paketa je lahko lokalna namestitev za uporabnika, ki namešča programsko opremo, ali lokalna namestitev za vse uporabnike računalnika. Namestitev se lahko izvaja iz omrežja oziroma na drug računalnik v omrežju.

2.2.2 Internacionalizacija namestitvenega paketa

V času globalizacije ima internacionalizacija programske opreme vse večji pomen, zato se programska oprema izdeluje tako, da se z nekaj nastavitvami spremeni jezik prikazovanja sporočil. S tem je programska oprema prijaznejša in dostopna širšemu krogu uporabnikov.

2.2.3 Odstranitev programske opreme

Pomemben del namestitve je tudi odstranitev programske opreme. Pravilna odstranitev se mora začeti že v času nameščanja. Takrat se zabeleži, kaj se je namestilo. Poleg nameščenih datotek, je treba odstraniti tudi datoteke, ki so morebiti nastale ob namestitvi ali med uporabo programske opreme. Odstranitev mora povrniti vse nastavitve, ki so se spremenile v času nameščanja.

Tehnologije namestitev po različnih platformah in operacijskih sistemih ne omogočajo vedno tudi odstranitve.

2.2.4 Vrste namestitvenih paketov

Namestitev kot postopek je enaka za vse platforme, ampak se kljub temu namestitveni paketi razlikujejo predvsem po tem. Tehnologije namestitev se razvijajo povsem neodvisno med seboj, kar je posledica dejstva, da je programska oprema narejena za en operacijski sistem.

Primeri:

- Linux: ker je več ponudnikov operacijskega sistema Linux, poznamo tudi različne vrste paketnih datotek za namestitev. Loči se jih po končnici imena datoteke: .rpm, .deb, ...;
- podjetje Microsoft v svojih operacijskih sistemih koristi predvsem končnico .msi. Možno je tudi zapakirati namestitev v izvršno datoteko tipa .exe;
- na operacijskih sistemih MAC se koristi predvsem končnica .pkg;
- na sistemih Unix se uporablja .depot.

2.3 Namestitvev kot proces

Namestitvenim procesom je skupno to, da so to posebni programi, ki znajo namestitvene pakete prebrati, jih ustrezno nastaviti in namestiti programsko opremo na uporabnikov računalnik.

2.3.1 Skrbstvo nad datotečnim sistemom in pravice uporabe

Namestitev pogosto potrebuje skrbniške pravice pri svojem izvajanju. V ta namen mora biti namestitveni proces zmožen izvajati opravila s povišanimi pravicami uporabe.

Primer namestitve na računalnik, ki je v rabi med več uporabniki: namestitveni program mora poleg skrbniških pravic znati namestiti program tako za enega uporabnika in s tem omogočiti diskretnost temu uporabniku kot tudi znati namestiti program v večuporabniškem okolju.

Pri tem mora imeti operacijski sistem nadzor nad namestitvami, da ne pride do nedovoljenih akcij.

2.3.2 Podatki za nadgraditev in odstranitev

Za ustrezno odstranitev oziroma nadgradnjo je treba vedeti kaj se nadgradi in ne oziroma kaj se lahko odstrani in ne. Večina teh podatkov je v domeni razvijalca namestitvenega paketa, ki mora vedeti, kaj je potrebno odstraniti oziroma nadgraditi.

Nadgraditev :

- nedostopnost oziroma nezmožnost zamenjave izvršnih datotek, če so te v stanju delovanja. Operacijski sistemi največkrat onemogočajo odstranitev oziroma kakršnekoli spremembe na izvršnih datotekah v času njihovega delovanja. Zato je pri izdelavi namestitvenega paketa potrebno upoštevati:
 - zaustavljanje/zagon izvršne datoteke oziroma storitve,
 - ponastavitev storitve,
 - nadgradnja izvršnih datotek storitev.
- nastavitve se morajo ohraniti: odvisno od vrste namestitvene tehnologije. Nekatere tehnologije delujejo le na principu odstrani–ponovno namesti, kar je treba upoštevati pri nadgradnji starih nastavitvev, da se jih predhodno shrani na začasno lokacijo in nato obnovi.

Odstranitev:

- zaustavljanje storitvenih in izvršnih datotek v času delovanja: za pravilno odstranitev je potrebno zagotoviti primerno zaustavitev storitve ali programov pred odstranitvijo;
- odstranitev programskih knjižnic, gradnikov, programskih dodatkov, ... kjer je potrebno odstraniti tudi vse nastavitve, ki so vezane na te dele programskih datotek;
- brisanje uporabniških, kasneje dodanih datotek: v večini primerov mora to zagotoviti razvijalec namestitvenega paketa;
- brisanje nastavitvev.

2.3.3 Kaj če – obravnavanje napak pri instalaciji

Med namestitvijo se lahko pojavi napaka. Kaj v tem primeru? Po načelu dobre prakse mora vsak program ob napaki pustiti za seboj prejšnje stanje – pobrisati nameščene datoteke, nastavitve ipd.

Namestitev:

- če pride do napake, mora namestitev pobrisati za sabo. V primeru nadgradnje mora namestitev ohraniti prejšnjo, že nameščeno verzijo. Najtežje napake se zgodijo ravno med nadgraditvijo, ker zaradi nepravilne izdelave paketa lahko ostane programska oprema v nekem vmesnem stanju med staro in novo verzijo.

Odstranitev:

- mora postaviti nazaj vse datoteke, ki so bile pred odstranitvijo, in jih zopet pravilno namestiti;
- postaviti nazaj vse nastavitve (če stanje ni pravilno povrnjeno, se program morda sploh ne požene).

2.3.4 Beleženje poteka dela

Beleženje poteka dela v datoteko je uporabno predvsem za razvijalce, ker dnevniške datoteke omogočajo boljši pregled nad delovanjem namestitve in enostavnejši pregled nad napakami.

Vse namestitve ne omogočajo beleženja poteka namestitve.

2.4 Orodja za izdelavo programskih paketov

Orodja se med seboj ločijo po računalniških operacijskih sistemih, za katere se namestitveni paket gradi. Nekatera orodja znajo graditi namestitveni paket za več operacijskih sistemov. S takimi orodji se naredi skupen projekt, iz katerega se zgradi namestitvene pakete za vsak operacijski sistem posebej.

Orodja po dostopnosti lahko razdelimo na komercialna in odprtokodna.

Tabela 1: Nekaj orodij za graditev namestitvenih paketov

Ime orodja	Proizvajalec	Podprt operacijski sistem	Dostopnost na tržišču
InstallAnywhere	Flexera Software	Microsoft Windows, Linux, Mac OS X, Solaris, AIX, HP-UX, in IBM iSeries	Komercialno orodje
JExpress	DeNova	Microsoft Windows, Linux, Mac OS X, Solaris	Komercialno orodje
InstallBuilder	BitRock Inc.	Microsoft Windows, Linux, Mac OS X, Solaris	Komercialno orodje
WISE	MindVision Software	Mac OS in Microsoft Windows	Komercialno orodje
WiX		Microsoft Windows	Odprtokodno orodje

NSIS		Microsoft Windows	Odprihodno orodje
IzPack		vse platforme, ki podpirajo Javo	Odprihodno orodje
InstallShield	Flexera Software	Microsoft Windows	Komercialno orodje
Wise Installation Studio	Wise Solutions, Inc.	Microsoft Windows	Komercialno orodje
InstallerMaker	StuffIt	Mac OS	Komercialno orodje

3 Tehnologija MSI

Podjetje Microsoft je razvilo svojo programsko opremo za nameščanje programske opreme – tako imenovano Microsoft Installer ali kratko MSI. Uporablja se jo izključno v Microsoftovih operacijskih sistemih. Tehnologija MSI je bila prvič predstavljena leta 2000 ob izidu operacijskega sistema Windows 2000.

Podjetje Microsoft je razvilo to tehnologijo za lažjo distribucijo programske opreme v podjetjih, ki potrebujejo programsko opremo na več posameznih računalnikih. Namen izdelave te tehnologije je namestitev programske opreme na daljavo s pomočjo omrežja oziroma s pomočjo tako imenovane omrežne programske opreme "aktivni imenik".

Razvoj tehnologije MSI je omogočil namestitev v dveh načinih:

- z dodelitvijo programske opreme. Ta se namesti kasneje, in sicer:
 - če se namestitev dodeli uporabniku, se običajno programska oprema namesti v celoti šele, ko želi uporabnik prvič pognati aplikacijo;
 - če se namestitev dodeli računalniku. Namestitev se izvede, ob prvem naslednjem zagonu računalnika;
- z objavo namestitve programske opreme. Ker je namenjena le uporabniku, se je ne more objaviti računalniškemu sistemu. Z objavo programska oprema dejansko ni nameščena, ampak dobi uporabnik možnost namestitve v seznamu za namestitev v *Nadzorni plošči* operacijskega sistema, kjer jo mora uporabnik sam sprožiti.

Tehnologija MSI je na lokalnem računalniku sestavljena iz dveh delov:

- namestitveni program s programskimi knjižnicami, ki se uporabljajo za namestitev kateregakoli programskega paketa. Ta se nahaja na računalniku, kjer se namešča programsko opremo;
- namestitveni paket, ki vsebuje vse podatke in datoteke potrebne za namestitev programske opreme. Ta se zgradi na proizvajalčevem sistemu.

MSI je zelo široko zastavljena tehnologija in praktično ni možnosti, ki je namestitvena tehnologija ne podpira.

MSI je opredeljena v priporočilih skladnosti *Microsoft Logo*, ki opisuje, kako, kam in na kakšne načine naj se namešča programska oprema in kako naj ta deluje.

V večji meri razvijalci namestitvenega paketa skupaj z razvojno skupino programa samega določijo, kako se bo program namestil na ciljni računalnik. V primeru namestitev storitvenih in sistemskih programov Microsoft le priporoča način nastavitve, v nekaterih primerih pa je način, kako naj se namestitev storitve izvede, celo obveza.

Tehnologija MSI omogoča:

- nameščanje izvršnih datotek s programskimi knjižnicami ter ostalih binarnih datotek,
- nameščanje sistemskih in storitvenih izvršnih datotek, programskih knjižnic v skupni rabi ipd.,
- nameščanje programskih dodatkov – vtičnikov in gradnikov (razne knjižnice Active-X ipd.),
- digitalno podpisovanje datotek¹ in namestitvenega paketa,
- nastavitve operacijskega sistema in programske opreme, ki se namešča (nastavitve v registru, v nastavitvenih in ostalih uporabniško določenih datotekah),
- ponovni zagon operacijskega sistema,
- iskanje vnaprej določenih datotek, nastavitvev ipd. na računalniku, kjer se namešča paket,
- beleženje poteka namestitve, ...

3.1 Programska oprema tehnologije MSI

Srce programske opreme je program *msiexec.exe* (v nadaljevanju MSIEEXEC), ki je del vsakega Microsoftovega operacijskega sistema. Z njim se požene vsako namestitev, nadgradnjo ali odstranitev programskih paketov. Program deluje v dveh načinih: kot storitev in kot samostojni izvršni program. Omogoča tudi sočasno delovanje v obeh načinih.

3.1.1 Namestitev kot storitev

MSIEEXEC se samodejno namesti ob namestitvi operacijskega sistema. Čeprav je to samostojen program, se ga lahko šteje kot del operacijskega sistema.

Ob namestitvi operacijskega sistema se postavi tudi nova storitev, ki je namenjena izključno namestitvam. Prednost tega je, da se lahko namešča programsko opremo, ki zahteva skrbniške pravice. Sistemske storitve imajo vedno večje privilegije kot skrbnik.

Prednost namestitvene storitve je tudi v tem, da se jo lahko koristi za nameščanje programske opreme na daljavo, kar skrbnikom računalniških sistemov zelo olajša delo: iz enega mesta lahko zaženejo, nadzorujejo, nameščajo popravke in odstranjujejo nameščeno programsko opremo.

3.1.2 Programske knjižnice in programski dodatki

Program uporablja svojo programsko knjižnico *msi.dll*, ki ima tudi javno dostopni vmesnik do svojih funkcij. Poleg namestitve omogočajo funkcije programske knjižnice *msi.dll* še vpogled v dele namestitvenega paketa.

¹ Digitalno podpisovanje datotek je uvedeno kot rešitev za odpravo problemov v zvezi z namestitvijo dinamičnih programskih knjižnic, ker je prihajalo do nenadzorovanih nadgradenj med različnimi verzijami iste datoteke dinamične knjižnice. S pomočjo digitalnega podpisa je to izključeno, ker program pred nalaganjem knjižnice najprej preveri, če digitalni podpis ustreza.

MSIEXEC zna po potrebi programskega paketa dinamično naložiti tudi druge programske knjižnice in dodatke.

3.2 MSIEXEC

V Microsoftovih operacijskih sistemih imajo nekatere mape določena virtualna imena, ki so dostopna vsem delujočim programom. Take mape so na primer uporabniška domača mapa, systemska mapa datotek operacijskega sistema ipd. Poleg teh map pozna MSIEXEC še več virtualnih imen map. Take mape so dostopne preko spremenljivk, ki razvijalcem olajša nastavitve ciljnih map datotek v namestitvenem paketu – razvijalec uporabi virtualna imena, ki se med namestitvijo nadomestijo s pravimi vrednostmi iz ciljnega računalnika. Nekaj takih spremenljivk prikazuje Tabela 4 na strani 20.

MSIEXEC ima tudi vmesnik, da lahko izvaja ukaze iz zunanjih izvršnih datotek, ki so bodisi del operacijskega sistema bodisi del namestitvenega paketa. Omogoča tudi izvajanje ukazov nekaterih skriptnih jezikov (Visual Basic Scripting in JavaScript), da ni potrebe po večjem znanju programiranja.

3.2.1 Ravnanje ob napakah

Bistvena lastnost tehnologije MSI je povrnitev računalnika v prejšnje stanje ob morebitni napaki med namestitvijo. MSIEXEC koristi tako imenovani transakcijski način namestitve: vsako akcijo si beleži, da lahko ob napaki izvede njeno obratno akcijo (več o tem v nadaljevanju Potek namestitve, stran 13).

3.2.2 Podatkovna baza namestitev

Da se lahko nek programski paket odstrani, je potrebno imeti podatke o tem, kaj se je pri namestitvi dejansko naredilo. Tehnologija MSI koristi sistemski register za hrambo vseh teh podatkov. V tej podatkovni bazi se hranijo zapisi o nameščenih datotekah, vključno z različico, in zapisi o med nameščanjem narejenih mapah.

Za pravilno odstranitev tehnologija MSI potrebuje tudi namestitveni paket, iz katerega se je program namestil.

Datoteke za nadgraditev ali odstranitev aplikacije so shranjene v posebni mapi *Install*, ki je del systemske mape operacijskega sistema *Windows*.

Pri opisu podatkovne baze namestitve ne moremo mimo tako imenovanih *GUID-ov* (globalni edinstven niz znakov, dolžine 36 znakov, ki predstavlja število v šestnajstiški obliki oziroma v binarni obliki število velikosti 128 bitov). MSI označuje z GUID-i skoraj vsak podatek.

Najvažnejše oznake so identifikacijske številke za označevanje produkta, paketa in identifikacijska številka za označevanje nadgradenj (glej Nadgradnja, stran 11). Tudi vsaka komponenta namestitvenega paketa ima svoj GUID (glej Tabele, namenjene shranjevanju datotek, stran 18).

V registru obstajajo zapisi najprej o nameščenih produktih in njihovih nadgradnjah ter posebej še o komponentah. Obstajajo tudi zapisi o povezavah med produkti in njihovimi komponentami.

Ob namestitvi MSIEXEC najprej preveri podatkovno bazo, če je tak paket že nameščen.

3.2.3 Načini delovanja MSIEXEC programa

Poleg namestitve zna MSIEXEC izvesti:

- nadgradnjo in popravek namestitve,
- spremembo namestitve,
- odstranitev,
- administrativno namestitev,
- tiho namestitev.

Za ugotavljanje načina delovanja ima tehnologija MSI vgrajen poseben mehanizem na osnovi treh identifikacijskih števil, ki morajo biti zapisane v vsakem namestitvenem paketu (glej Namestitveni paketi, stran 15). Te so:

- identifikacijska številka projekta,
- identifikacijska številka namestitvenega paketa in
- identifikacijska številka nadgradnje.

MSIEXEC najprej preveri identifikacijske številke namestitvenega paketa s tistimi iz podatkovne baze in ukrepa tako, kot prikazuje Tabela 2: Odvisnosti med posameznimi ID-številkami namestitve:

Tabela 2: Odvisnosti med posameznimi ID-številkami namestitve

Kaj se zgodi	ID-produkta	ID-paketa	ID-nadgraditve
Namestitev	Ni v bazi	Ni v bazi	Ni v bazi
Nadgraditev	Ni v bazi	Ni v bazi	Najdena v bazi
Mala nadgraditev	Najdena v bazi	Ni v bazi	Najdena v bazi
Popravek namestitve	Najdena v bazi	Najdena v bazi	Najdena v bazi
Odstranitev	Najdena v bazi	Najdena v bazi	Najdena v bazi

Nadgradnja

V primeru popravkov programske opreme ali zaradi vgrajenih novih funkcionalnosti se pri nadgradnji zamenja datoteke in nastavitve prejšnje verzije nameščenega produkta z novejšo različico.

Tehnologija MSI pozna dva primera nadgraditev:

- mala nadgraditev, kjer se nadgradijo oz. zamenjajo le datoteke, struktura namestitvenega paketa (datoteke, način, kako se izvajajo akcije ipd.) pa ostaja enaka;

- večja nadgraditev, pri kateri je lahko tudi struktura namestitvenih paketov popolnoma različna.

Pri večji nadgradnji je potreben namestitveni paket starejše verzije, to je verzije, ki se nadgrajuje. MSIEXEC ga dobi v mapi, kjer se hranijo paketi za odstranitev. Večjo nadgradnjo se lahko vgradi na štiri načine (za razlago posameznih faz namestitve glej Potek namestitve, stran 13):

- najprej se odstrani stara verzija izven transakcije, nato namesti nova.
Slabost: če pride do napake med namestitvijo nove verzije, lahko po povrnitvi v prejšnje stanje ostane produkt odstranjen;
- nova verzija se namesti, nato pa stara odstrani.
Je sicer bolj učinkovita od prejšnje, vendar v primeru napake lahko ostane namestitveni program v nekem vmesnem stanju;
- najprej se odstrani stara verzija, vendar se v okviru transakcije nato namesti nova.
Je najvarnejši način namestitve, ker v nobenem primeru ne ostane namestitev na pol ali celo odstranjena, vendar najmanj učinkovita;
- nova verzija se namesti, nato pa stara odstrani v okviru transakcije.
Je načelno najučinkovitejša nadgraditev, ker se izvaja vse pod transakcijo – v primeru napake se povrne prejšnje stanje.

Del nadgradnje, kjer se odstranjuje "stara" namestitev, je skoraj enak, kot če bi se pognala normalna odstranitev, zato mora razvijalec predvideti, da se ob koncu nadgradnje zopet vzpostavi prejšnje stanje. Razvijalec namestitvenega paketa se mora zavedati, da nadgraditev najprej odstrani staro verzijo s "starim" paketom, zato mora biti paket narejen tako, da je na nek način korak pred trenutno verzijo.

MSIEXEC napredno ugotavlja, katere datoteke je treba zamenjati z novimi. Večinoma to ugotavlja na osnovi datuma nastanka datoteke, razen pri izvršnih datotekah in programskih knjižnicah, kjer lahko preveri različico produkta s posebnimi podatki o verziji, ki so vgrajeni v samo datoteko. Če ti podatki ne obstajajo, preverja enako kot pri ostalih datotekah. Vsebine datotek ne preverja.

Sprememba namestitve

Do spremembe namestitve pride takrat, ko obstajajo v registru vsa tri identifikacijska števila enaka kot v namestitvenem paketu. Takrat namestitev ponudi možnost popravka ali spremembe instalacije.

Spremembe namestitve se lahko izvedejo pri razdelanem projektu po posameznih funkcionalnostih, kjer se želi naknadno dodati/odstraniti le del funkcionalnosti projekta.

Ta vrsta namestitve je primerna tudi za popravek namestitve v primeru izgube nekaterih prej nameščenih datotek.

Razvijalec namestitvenega paketa mora predvideti, kaj lahko taka namestitev popravi: običajno ni v redu, da popravi kakšne uporabnikove nastavitve.

Administrativna namestitvev

To je posebna vrsta namestitvev, kjer se namesti paket na neko omrežno mesto, da lahko uporabniki omrežja lažje namestijo paket iz enega mesta. Če je paket narejen tako, da ga je možno namestiti v omrežje, imajo uporabniki možnost uporabe programa neposredno iz omrežja.

Namestitev dejansko le razpakira vsebino namestitvenega paketa tako kot je določeno v paketu.

Tiha namestitvev

To je običajna namestitvev, kjer se ne prikazuje grafični vmesnik, ampak se celotna namestitvev izvaja brez posredovanja uporabnika. Če namestitev zahteva, da uporabnik vnese določene podatke, je potrebno pri tihi namestitvi to vnaprej predvideti. Lahko se vgradi v namestitveni paket privzete vrednosti ali pa zahteva namestitveni paket vnos parametrov iz ukazne vrstice.

3.2.4 Beleženje poteka namestitvev

Za namene razhroščevanja in iskanja napak namestitvenega paketa omogoča MSIEXEC beleženje poteka namestitvev. V odvisnosti od potreb se lahko beleži od enostavnih akcij do detajlnega poteka namestitvev.

Beleženje je lahko že vnaprej vklopljeno na nivoju računalnika v skupinskih politikah računalnika. Če tega ni, pa se s posebnim stikalom pove MSIEXEC-u, da vklopi beleženje.

3.3 Potek namestitvev

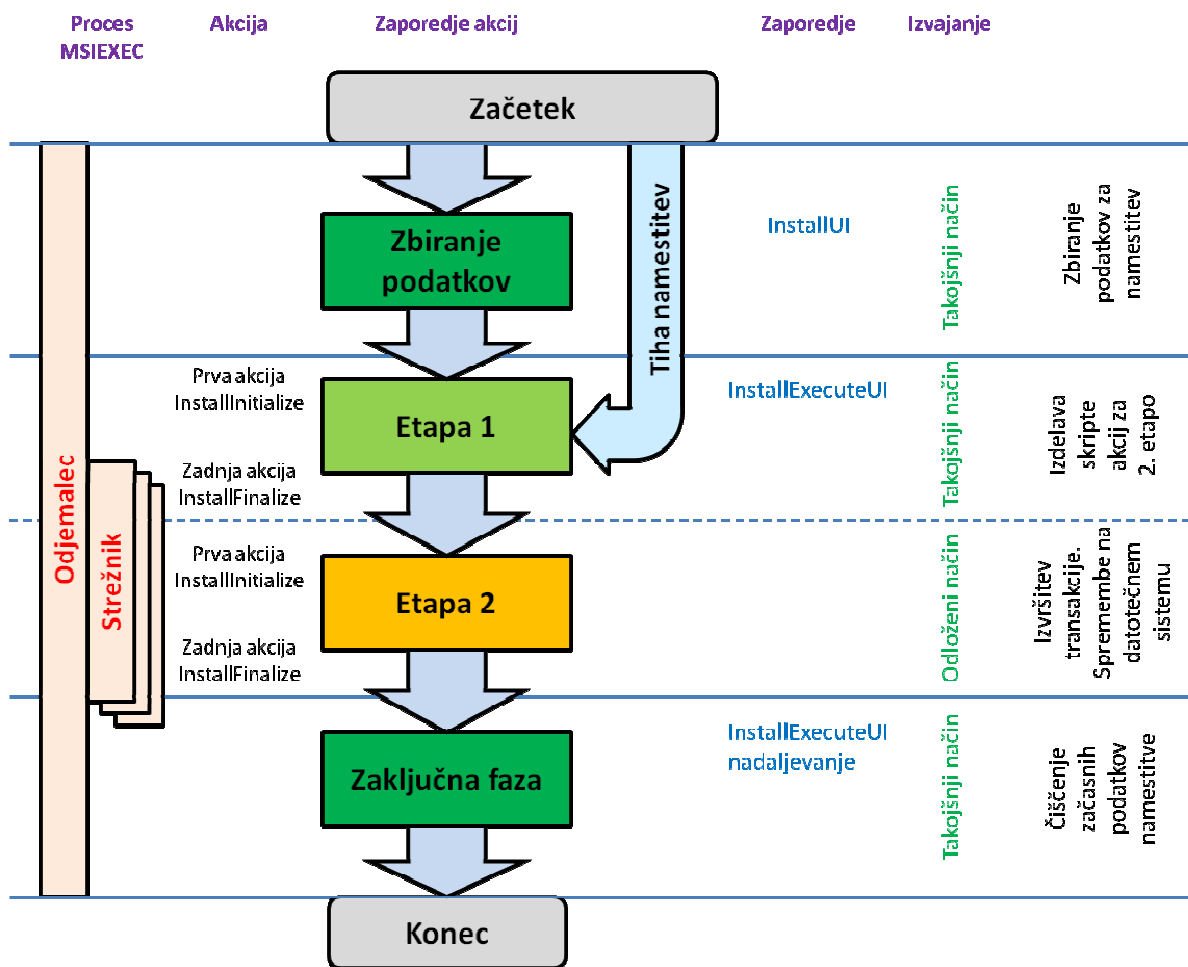
Namestitev se odvija v več procesih, ki jih lahko razdelimo na odjemalce in strežnike. Z odjemalcem so označeni procesi, ki se uporabljajo za komunikacijo z uporabnikom (zbiranje podatkov in prikaz poteka namestitvev), s strežnikom pa tiste procese, ki dejansko izvajajo namestitev.

Tehnologija MSI določa zaporedje namestitvev (glej Slika 1: Namestitev po posameznih fazah):

- zaporedje akcij "*InstallUI*" – zbiranje podatkov za namestitev,
- zaporedje akcij "*InstallExecute*",
 - etapa 1, kjer se tvori skripta za izvršitev transakcije,
 - etapa 2 – izvršitev transakcije,
- zaporedje akcij "*InstallExecute*" nadaljevanje – zaključna faza.

Zaporedje akcij "*InstallExecute*" (do nadaljevanja) izvaja akcije v dveh etapah: v prvi etapi se gradi transakcijska skripta z akcijami, v drugi pa se te akcije izvedejo. Akcije v prvi etapi se izvajajo v takojšnjem načinu, v drugi pa v tako imenovanem odloženem načinu (angl. "*deferred execution*").

Druga etapa je namenjena spreminjanju podatkov na uporabnikovem disku, zato se v primeru napake lahko povrne v stanje, kot je bilo pred izvršitvijo druge etape.



Slika 1: Namestitev po posameznih fazah

3.3.1 Zbiranje podatkov za namestitev – zaporedje akcij "InstallUI"

Ta uvodni del je namenjen izključno zbiranju podatkov za namestitev, kot je na primer prikaz licenčne pogodbe, izbira lokacije namestitve ipd. Ta proces je označen kot odjemalec.

Bistven del teh akcij sestavlja grafični vmesnik, s pomočjo katerega se določa vse potrebne podatke in lastnosti, ki jih nastavitveni paket potrebuje.

Pri tihi namestitvi se ta faza izpusti. Na razvijalčevi strani je, da morebitne manjkajoče podatke nadomesti z vnaprej določenimi vrednostmi.

Zaporedje akcij "InstallUI" se izvaja v tako imenovanem takojšnjem izvajanju (angl. "immediate execution") kar pomeni, da se vsaka akcija (glej Akcije, stran 21) izvede, ko se prejšnja konča v zaporedju, kot je določeno v tabeli akcij "InstallUI".

Ta etapa se začne z akcijo "InstallInitialize" in konča z akcijo "InstallFinalize".

3.3.2 Graditev skripte za transakcijo – zaporedje akcij "*InstallExecute*" 1. etapa

Namen te etape je zgraditi zaporedje ukazov za transakcijsko izvršitev namestitve.

Namestitev obdela vsako akcijo iz tega zaporedja, in če je akcija označena kot odložena, jo potem postavi v transakcijo z vsemi potrebnimi parametri.

Ta etapa ima na voljo vse funkcije in spremenljivke namestitvenega procesa in spremenljivke okolja operacijskega sistema, vključno z oznakami nekaterih vnaprej znanih map.

Proces te faze lahko imenujemo strežnik, ker se izvaja v ozadju.

3.3.3 Izvajanje transakcije - zaporedje akcij "*InstallExecute*" 2. etapa

Zaporedje akcij se izvaja v odloženem načinu. Akcije se sicer poženejo v prvi etapi v določenem zaporedju, vendar lahko tečejo vse akcije vzporedno, ker se poženejo asinhrono. Ker konec akcije ni točno določen, razvijalec namestitvenega paketa ne more računati na zaporedje, podano v zaporedju akcij "*InstallExecute*".

Značilnost druge etape je med drugim tudi ta, da nima na voljo vseh informacij kot v ostalih fazah namestitve (na primer vnaprej določena imena map, okoljske spremenljivke ipd.).

Ob napaki se izvajanje ustavi in transakcija se prekine, kar povzroči povrnitev stanja pred izvršitvijo te etape. Povrne se le stanje datotek in map, ki so se kopirale, stanje storitev in posegov v sistemski register. Za vse ostale akcije, ki se izvajajo v tej etapi, mora razvijalec po potrebi sam določiti protiakcijo – akcijo, ki se izvede v primeru napake.

Tudi ta etapa se začne z akcijo "*InstallInitialize*" in konča z akcijo "*InstallFinalize*".

3.3.4 Izvajanje zaporedja akcij "*InstallExecute*" nadaljevanje

To je zaključna faza namestitve, ki se nadaljuje po akciji "*InstallFinalize*". Namenjena je brisanju vseh začasnih podatkov, nastalih med namestitvijo, morebitnem prvem zaganjanju programa ipd., ker je bistveni del namestitve končan. Izvaja se v takojšnjem načinu.

3.4 Namestitveni paketi

Tehnologija MSI pozna predvsem datoteke tipa .msi, .msm in .mst, v nekaterih primerih pa tudi .msp.

Datoteka s končnico .msi je običajni namestitveni paket. Datoteka s končnico .msm predstavlja poseben modul, ki se vključuje v datoteko tipa .msi med graditvijo, datoteke s končnico .mst predstavljajo posebne module, ki se vključujejo v datoteko tipa .msi med samo namestitvijo. Datoteka s končnico .msp predstavlja programski popravek namestitve ali tako imenovano malo nadgradnjo.

3.4.1 Izdelava namestitvenega paketa

Namestitveni paket se lahko zgradi le s posebnimi programskimi orodji za graditev takih paketov. Nekaj orodij je že naštetih v prejšnjem poglavju (glej Orodja za izdelavo programskih paketov, stran 6, in Tabela 1: Nekaj orodij za graditev namestitvenih paketov).

Da bi se varčevalo s prostorom, se lahko namestitveni paket tudi programsko stisne. To je ugodno predvsem za pakete, ki se nameščajo preko omrežja, ker se želi čim krajši in s tem hitrejši prenos.

Nekatera orodja vgradijo namestitveni paket kar v izvršno datoteko s končnico .exe. Prednosti takega paketa so, da se lahko požene kakšno dodatno akcijo, ki jo dejanski namestitveni paket potrebuje še pred namestitvijo. Primer take akcije je namestitev programskega dodatka za paket, ki je narejen z orodjem "*InstallShield*".

3.4.2 Orodja za izdelavo paketa .msi

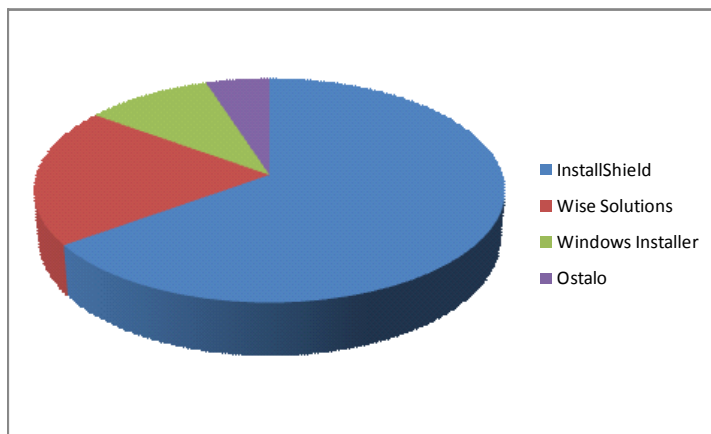
V računalniškem prostoru, omejenem na operacijske sisteme podjetja Microsoft, obstaja več vrst orodij za graditev namestitvenih paketov.

Razvijalcem najprijaznejša orodja so običajno komercialne narave, ker imajo dodelan grafični vmesnik. Na drugi strani je nekaj odprtokodnih rešitev, od katerih se najbolj uveljavlja orodje WiX (glej WiX – odprtokodna rešitev tehnologije MSI, stran 26).

Tabela 3: Orodja za graditev namestitvenih paketov tehnologije MSI

Ime programskega orodja	Proizvajalec	Verzija	Omogoča
MS Visual Studio	Microsoft Inc.	Komercialno orodje – dodatek osnovnemu orodju	Osnovna namestitev: ena funkcijska enota, ne omogoča nadgraditev. Slab pregled nad referenčnimi datotekami, ker jih projekt Visual Studio samodejno vključi.
Advanced Installer Pro	Caphyon Ltd.	Komercialna različica	Napredne rešitve namestitve: na primer nastavitev lastnosti, instalacija ogrodja .NET, nastavitev in namestitev gonilnikov, ...
Advanced Installer	Caphyon Ltd.	Osnovna verzija odprtokodna rešitev	Le osnovne možnosti tehnologije MSI.
WinINSTALL Pro	Scalable Software Limited	Osnovna verzija odprto kodna rešitev, Pro komercialna	Napredne rešitve namestitve.
Wise for Windows Installer	Symantec Inc. (ki ga kupilo od Wise Solutions Inc.)	Komercialna različica	Tako rekoč vse, kar omogoča tehnologija MSI.
InstallShield	Flexera Software Inc.	Komercialna različica	Tako rekoč vse, kar omogoča tehnologija MSI. InstallShield dodaja osnovni tehnologiji MSI še nekaj dodatnih funkcionalnosti, vendar paket še vedno potrebuje tehnologijo MSI za namestitev.
Windows	Izvorni avtor	Odprtokodna	Omogoča napredne lastnosti namestitve:

Installer XML – "WiX"	Rob Mensching, izdeluje se v skupnosti SourceForge.net	rešitev	namestitve gonilnikov, storitev ipd. Posebnost tega orodja je izvorna koda v formatu XML. Graditev paketa je po načelu graditve programske opreme: prevajanje in povezovanje.
--------------------------	---	---------	---



vir: <http://www.installsite.net>, oktober 2010

Slika 2: Zastopanost orodij za graditev namestitvenega paketa v tehnologiji MSI

3.4.3 Organizacija paketa

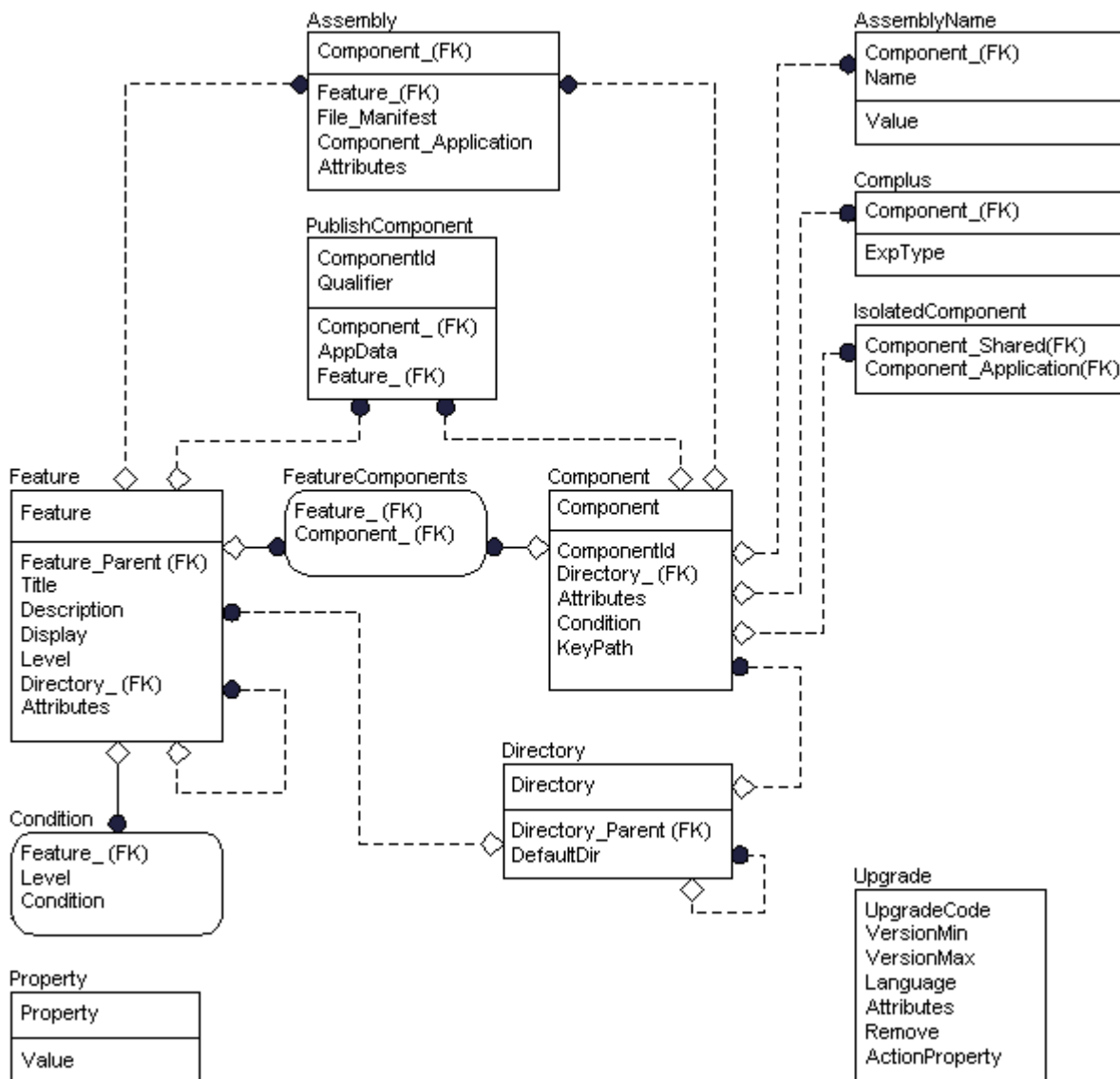
Namestitveni paket tehnologije MSI lahko primerjamo s podatkovno bazo, kar v bistvu tudi je: posamezni elementi so organizirani po tabelah, zapisi v tabelah pa predstavljajo podatek tega elementa.

Na primer: datoteke za namestitev predstavljajo enega izmed elementov paketa. Ta element lahko poimenujemo tabela datotek, posamezen zapis v tej tabeli pa predstavlja datoteko. Podobno velja tudi za razne spremenljivke, ki so v svoji tabeli, akcije ipd.

Tabele namestitvenega paketa se lahko razdeli na dva dela:

- tabele z datotekami,
- tabele za krmiljenje namestitve.

3.4.4 Tabele, namenjene shranjevanju datotek



vir: <http://msdn.microsoft.com/en-us/library/aa368047%28v=VS.85%29.aspx>, november 2010
 Slika 3: Glavne tabele datotek paketa .msi z medsebojnimi relacijami.

Komponente (angl. Components)

Osnovni namen namestitvenega paketa je namestiti program v ciljni računalnik. Program je sestavljen iz ene ali več datotek, zato je tudi datoteka osnovni gradnik paketa. Tehnologija MSI imenuje ta osnovni gradnik "*komponenta*" (angl. component) in je sestavni del tabele komponent. Komponento opisuje več lastnosti:

- identifikacijska številka, ki mora biti enovit podatek (glej tudi Podatkovna baza namestitev, stran 10),
- razvijalcu prijazno ime komponente,
- izvorna datoteka na graditvenem sistemu,
- datoteka na ciljnim računalniku,
- namestitvena mapa te komponente v ciljnim računalniku,

- pogoj, pri katerem se komponenta namesti,
- pravice nad uporabo datoteke na ciljnem računalniku,
- ostale lastnosti (med drugim: ali je to 64-bitna datoteka, ali je npr. programska knjižnica v skupni rabi ipd.)

Komponenta je namenjena datoteki, vendar ni nujno, da jo tudi vsebuje. Vsaki komponenti je treba določiti datoteko (če jo vsebuje), ki je nosilec komponente. Ta datoteka privzame lastnosti, s katerimi se ugotavlja potrebo po zamenjavi v primeru nadgraditve ali popravljanja namestitve.

Komponenta lahko vsebuje tudi več datotek, vendar mora biti glavna le ena (vnaprej določena). Pri nadgradnji, če se glavna datoteka nadgradi, se vse ostale datoteke te komponente brezkompromisno nadgradijo.

Funkcijske enote (angl. Features)

Predstavljajo komponente, združene v posamezne skupine. Vsak namestitveni paket mora vsebovati vsaj eno funkcijsko enoto. Funkcijska enota poleg komponent lahko vsebuje tudi druge funkcijske enote. Tako so funkcijske enote postavljene v drevesno strukturo: ena enota vsebuje eno ali več drugih ipd.

Funkcijske enote so pomembne v primeru porazdeljenega produkta po posameznih programskih enotah, kjer se vsako enoto posebej lahko namesti ali pa ne.

Funkcijska enota lahko vsebuje tudi pogoj za namestitvev. Primer: paket vsebuje dve funkcijski enoti. V prvi so komponente z datotekami za 32-bitno platformo, v drugi pa za 64-bitno platformo. Katera funkcijska enota se instalira, je določeno s pogojem glede na vrsto platforme na ciljnem računalniku.

Če so funkcijske enote postavljene v drevesno strukturo, se lahko vsaka veja označi z lastnostjo "nivo namestitve" (angl. "*Install level*"). Princip uporabe te lastnosti je, da MSIEXEC namesti le tiste funkcijske enote, ki imajo številko nivoja manjšo ali enako skupni lastnosti nivoja.

Mape (angl. Directory)

Ta tabela vsebuje vsaj en zapis, in sicer zapis za namestitveno mapo. Pomembna je za akcijo izdelave map na ciljnem računalniku.

Nekatere mape se na ciljnem računalniku ob namestitvi naredijo, čeprav jih v tej tabeli ni. Take mape so določene v komponentah – MSIEXEC prekopira datoteko v ciljno mapo, ki jo prej naredi, če še ne obstaja.

V tej tabeli se nahajajo tudi mape, ki so določene iz virtualnih imen operacijskega sistema. Posamezni zapisi teh map se samodejno določijo ob namestitvi.

3.4.5 Tabele za krmiljenje namestitve

Namestitev je sekvenčni in točno določen proces, zato ga je treba v celoti opisati. V ta namen ima namestitveni paket kar nekaj tabel. Najvažnejše tabele med njimi:

- spremenljivke,
- pogoji,
- zaporedja izvajanja po posameznih fazah,
- akcije:
 - tabele posameznih aktivnosti (premikanje, kopiranje, odstranjevanje, popravljanje datotek),
 - tabele posameznih aktivnosti za nastavitve (register, nastavitvene datoteke ...),
- tabela sistemskih nastavitvev,
- tabela iskanja datotek, nastavitvev ipd.,
- podatki grafičnega vmesnika,
- večjezična podpora, ...

Spremenljivke (angl. Properties)

Spremenljivke so osnovni gradniki, namenjeni krmiljenju namestitve. Spremenljivke imajo enako vlogo kot spremenljivke v izvorni kodi klasičnega programskega jezika.

Za spremenljivke, ki jih določi razvijalec, velja, da sta dostop in področje veljave spremenljivke določena z velikostjo posameznih znakov v imenu spremenljivke.

Primer: če ima spremenljivka zapisane vse znake z velikimi črkami abecede, ima spremenljivka globalni dostop. To velja za vse faze namestitve, razen za drugo etapo izvršne faze namestitve, kjer to ne velja – ta etapa ima zelo omejen dostop do teh in sistemskih spremenljivk.

Spremenljivke se poleg uporabe za razna imena in podatke pri nastavitvah uporablja tudi pri pogojih.

Tabela 4: Nekaj vnaprej določenih spremenljivk tehnologije MSI

Ime spremenljivke	Opis
CommonFiles64Folder	Celotna pot do skupne mape programov za 64-bitne aplikacije
CommonFilesFolder	Celotna pot do skupne mape programov za trenutnega uporabnika
DesktopFolder	Celotna pot do mape Namizja
PersonalFolder	Celotna pot do mape Dokumentov za trenutnega uporabnika
ProgramFiles64Folder	Celotna pot do mape za programske 64-bitne aplikacije
ProgramFilesFolder	Celotna pot do mape za programske 32-bitne aplikacije. Običajno je ta mapa "C:\Program Files" na 32-bitnih operacijskih sistemih in "C:\Program Files(x86)" na 64-bitnih operacijskih sistemih
ProgramMenuFolder	Celotna pot do mape startnega menija
System16Folder	Celotna pot do sistemske mape za 16-bitne programske knjižnice
System64Folder	Celotna pot do sistemske mape za 64-bitne programske knjižnice
SystemFolder	Celotna pot do sistemske mape za 32-bitne programske knjižnice
TempFolder	Celotna pot do mape začasnih datotek

WindowsFolder	Celotna pot do sistemske mape Windows
WindowsVolume	Ime diska, kjer se nahaja sistemska mapa Windows
ComputerName	Ime računalnika, kjer se namešča paket
VersionNT	Verzija operacijskega sistema (ne velja za operacijske sisteme Windows98 in starejše)
ProductLanguage	Pove, s katero jezikovno skupino se prikazuje sporočila
LogonUser	Uporabniško ime, pod katerim se namešča paket

Akcije (angl. CustomActions)

Vsaka aktivnost, ki se izvaja med namestitvijo, je predstavljena z akcijo. Te se delijo na vgrajene akcije, ki se jih ne da izpustiti (na primer *InstallInitialize*), vgrajene akcije, ki jih lahko poljubno vgrajujemo in celo izpustimo (na primer *ServiceInstall*) in akcije po meri, ki jih vgradi razvijalec namestitvenega paketa.

Akcije po meri lahko izvajajo različne aktivnosti. Med drugim:

- nastavijo novo vrednost določeni spremenljivki namestitve;
- izvedejo klic funkcije skripte VBS²;
- izvedejo klic funkcije iz zunanje ali programske knjižnice MSI;
- ostale aktivnosti.

Vsaka akcija ima več svojih značilnosti, ki jo opisujejo:

- vrsta akcije,
- označba načina izvajanja (takojšnji, odloženi način, vpis ali preklic transakcije, s povišanimi privilegiji ipd.),
- pogoj za izvajanje,
- izvajanje v 32- ali 64-bitnem načinu,
- način izvajanja – sinhrono/asinhrono, preverjanje vrnjene vrednosti,
- ostale lastnosti.

Zaporedje izvajanja akcij (angl. Sequences)

Zaporedje je razdeljeno na dva dela: na zaporedje za zbiranje podatkov in zaporedje namestitve, ki se izvaja v dveh etapah (glej Potek namestitve, stran 13, in Slika 1: Namestitev po posameznih fazah).

Zaporedja so sestavljena iz akcij, ki so označena s svojo zaporedno številko, po kateri se akcije izvajajo. Negativna zaporedna števila imajo poseben pomen in se jih ne da določati po meri.

² VBS - Visual Basic Scripting je aktivni skriptni jezik na osnovi uspešnega programskega jezika Visual Basic podjetja Microsoft.

Pogojno izvajanje akcij (angl. Conditions)

Pogoji so pomemben del namestitvenega paketa. Z njimi se krmili izvajanje namestitve. Pripada jim posebna tabela.

Primer takega pogoja: namestitev se lahko izvede le na določenih operacijskih sistemih.

3.4.6 Validacija – ugotavljanje pravilnosti namestitvenega paketa

Validacija je mehanizem za preverjanje pravilnosti razvitega in pravkar zgrajenega namestitvenega paketa.

Preverja se pravilnost relacij med povezanimi zapisi, pravilnost tipov lastnosti ipd.

Pravila za preverjanje so določena s posebnimi tabelami, kar pripomore k preverjanju namestitvenega paketa po različnih namenih uporabe.

3.4.7 Posebne oblike namestitvenih datotek

Paketi, namenjeni spajanju k paketu v izdelavi

Ti moduli so namenjeni poenostavitvi graditve namestitvenega paketa. Vsebujejo lahko vse, kar vsebuje običajni namestitveni paket v fazi "*InstallExecuteUI*" (glej Slika 1: Namestitev po posameznih fazah). Iz tega sledi, da tak modul ne sme vsebovati grafičnega vmesnika.

Datoteke teh modulov imajo praviloma končnico `.msm`.

Transformacijski – diferenčni paketi

Transformacijski paketi so posebni moduli, ki se spajajo k osnovnemu paketu v času namestitve, z razliko od modulov, ki se spajajo v času graditve. Datoteke se praviloma končajo na `.mst`. Podatki in lastnosti iz transformacijskega paketa prekrijejo stare podatke ali pa dodajo nove.

Večji delež teh paketov predstavljajo podatki za lokalizacijo: v takem modulu so shranjene le informacije za eno jezikovno skupino. Glede na nastavitve ciljnega računalnika se izvede med namestitvijo transformacija namestitvenega paketa z novimi informacijami.

3.4.8 Lokalizacija paketa

Za lokalizacijo ima tehnologija MSI vgrajen mehanizem, da lahko namestitev prikazuje sporočila tudi v drugih jezikih ob predpostavki, da so ti dodatni jeziki vgrajeni v paket. MSIEXEC najprej preveri jezikovno področje računalnika, na katerega se program namešča. Če jezikovna skupina ustreza eni izmed vgrajenih v namestitveni paket, jo uporabi za prikazovanje, sicer uporabi prvo oziroma privzeto iz namestitvenega paketa.

Dodatni jeziki se običajno izberejo med namestitvijo, zato zahteva tak paket transformacijske datoteke, ki so lahko vključene že kar v posebni del paketa ali pa so samostojne. V drugem primeru mora uporabnik, ki namešča programsko opremo, podati MSIEXEC-u pravilno pot in ime do te transformacijske datoteke.

3.4.9 Obravnavanje napak

Tako kot vsaka programska oprema je tudi tehnologija MSI podvržena napakam:

- med izdelavo namestitvenega paketa,
- med namestitvijo.

Napake pri izdelavi namestitvenega paketa

Do teh napak pride v večji meri zaradi nepoznavanja tehnologije MSI in redkeje zaradi nepoznavanja sistemskega okolja.

Napake sintakse spadajo tudi v to poglavje. Z današnjimi programerskimi orodji se težje pripetijo in še te se odražajo že takoj na začetku – v času graditve, tako da se jih lahko že takoj odpravi.

Primeri napak, ki se vgradijo med izvedbo:

- napačna vgradnja in izvedba dodatnih akcij,
- napačna uporaba in inicializacija spremenljivk,
- napačna postavitvev akcij v zaporedje,
- nepoznavanje sintakse, ...

Napake med namestitvijo paketa

Teh napak je v večji meri deležno nepoznavanje raznolikosti računalniškega okolja. Kaj hitro se lahko zgodi, da katera izmed akcij ni podprta v računalniku, ki se namešča ipd.

Te napake je težje odkriti, ker razvijalec običajno nima dostopa do teh računalnikov. Za odkrivanje teh napak se uporabljajo predvsem dnevniške datoteke, v katere se zapisuje celoten postopek namestitve, ki ga razvijalec lahko analizira.

3.4.10 Uporabna orodja

Orodja Microsoftovega razvojnega paketa

Z Microsoftovim razvojnim paketom, imenovanem Microsoft SDK, pride tudi nekaj orodij, namenjenih razvoju namestitvenih paketov. Sem spadajo predvsem:

- Orca: z grafičnim vmesnikom podkrepljeno orodje za pregledovanje in popravljanje že narejenega paketa .msi;
- WiLogUtl.exe: za pregledovanje dnevniških datotek (glej WiLogUtl.exe, stran 24);
- MSIDB: orodje za obdelavo paketa .msi. Lahko se uvaža ali izvaža posamezne tabele ipd.
- MSITRAN: orodje za izdelavo transformacij. Na vhodu se orodju poda dva paketa .msi, izhod pa je transformacijska datoteka;
- MSIVAL2: orodje za preverjanje paketa .msi. Podati mu je treba datoteko s pravili preverjanja;

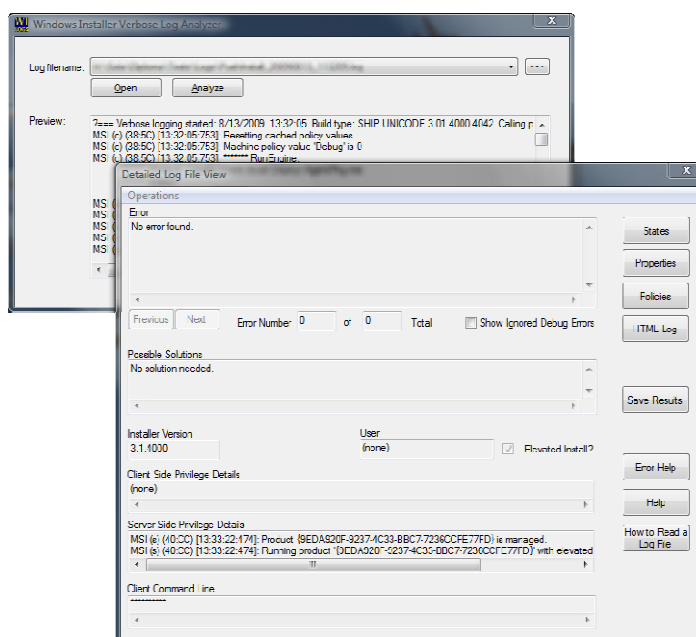
- MSIDiff: orodje za ugotavljanje razlik med posameznimi paketi.msi. Primerno je predvsem za ugotavljanje razlik med istimi paketi različnih verzij ipd.;
- skripte VBS za razna preverjanja in obdelavo že narejenih namestitvenih paketov. Najpomembnejša:
 - WiSubStg.vbs: skripta za popravljanje nekaterih informacij o namestitvenem paketu;
 - WiDiffDb.vbs: skripta za ugotavljanje razlik med posameznimi paketi;
 - WiLangId.vbs: skripta za nastavitve pravih oznak lokalizacijskih podatkov;
 - WiMakCab.vbs: skripta, ki naredi kompresirane datoteke iz osnovnega paketa .msi;
 - ...

WiLogUtl.exe

Je primerno orodje za analiziranje poteka namestitve. Analizira namestitvene dnevniške datoteke:

- potek namestitve po posameznih akcijah,
- vrednosti posameznih spremenljivk po procesih (odjemalec, strežnik, vmesni proces),
- stanje računalnika, na katerem se namešča paket,
- namestitvene politike na računalniku, na katerem se namešča paket,
- izvozi dnevniško datoteko v prijaznejši in bolj pregleden format HTML, ...

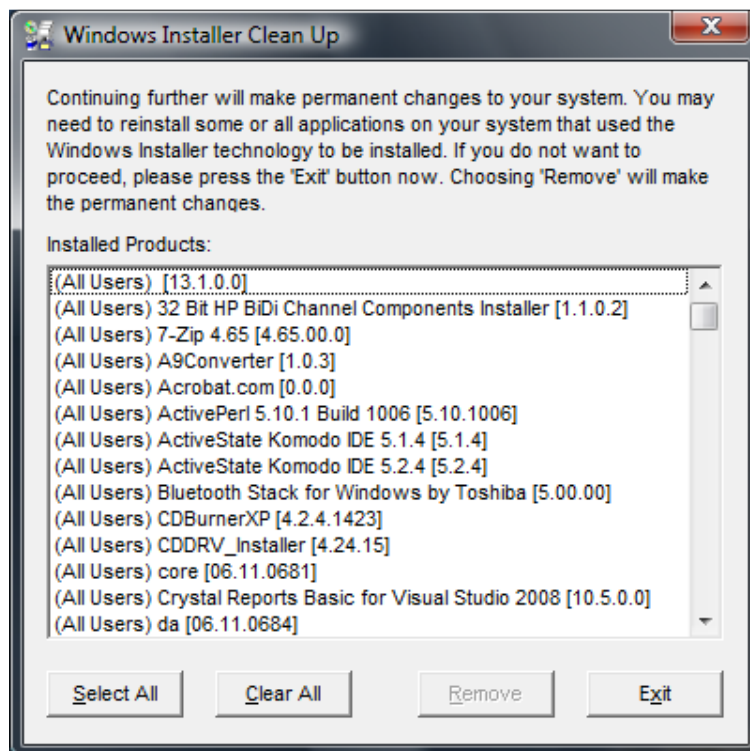
Orodje lahko izvozi dnevniško datoteko v preglednejšo datoteko za vpogled v "Pregledovalniku internetnih strani". Prikaz je v barvah in možno se je sprehajati med posameznimi napakami.



Slika 4: Orodje za analizo dnevniških datotek namestitve

Windows Installer Cleanup Utility

Na Microsoftovih straneh se nahaja še orodje za odstranjevanje namestitvev, imenovano "Windows Installer Cleanup Utility". Ta programček odstrani le podatke o namestitvi iz namestitvene podatkovne baze, kar se je dejansko naredilo ob namestitvi, pa ne. To mora uporabnik sam odstraniti. Primerno je predvsem za razvijalce namestitvev, ker točno vedo, kaj je namestitev naredila na sistemu.



Slika 5: Windows Clean Up Utility

4 WiX – odprtokodna rešitev tehnologije MSI

WiX oziroma s pravim imenom "Windows Installer XML" je odprtokodna programska oprema, s pomočjo katere se lahko zgradi namestitveni paket v tehnologiji MSI. Orodja WiX so prvotno razvili v podjetju Microsoft in potem objavili kot odprtokodno rešitev.

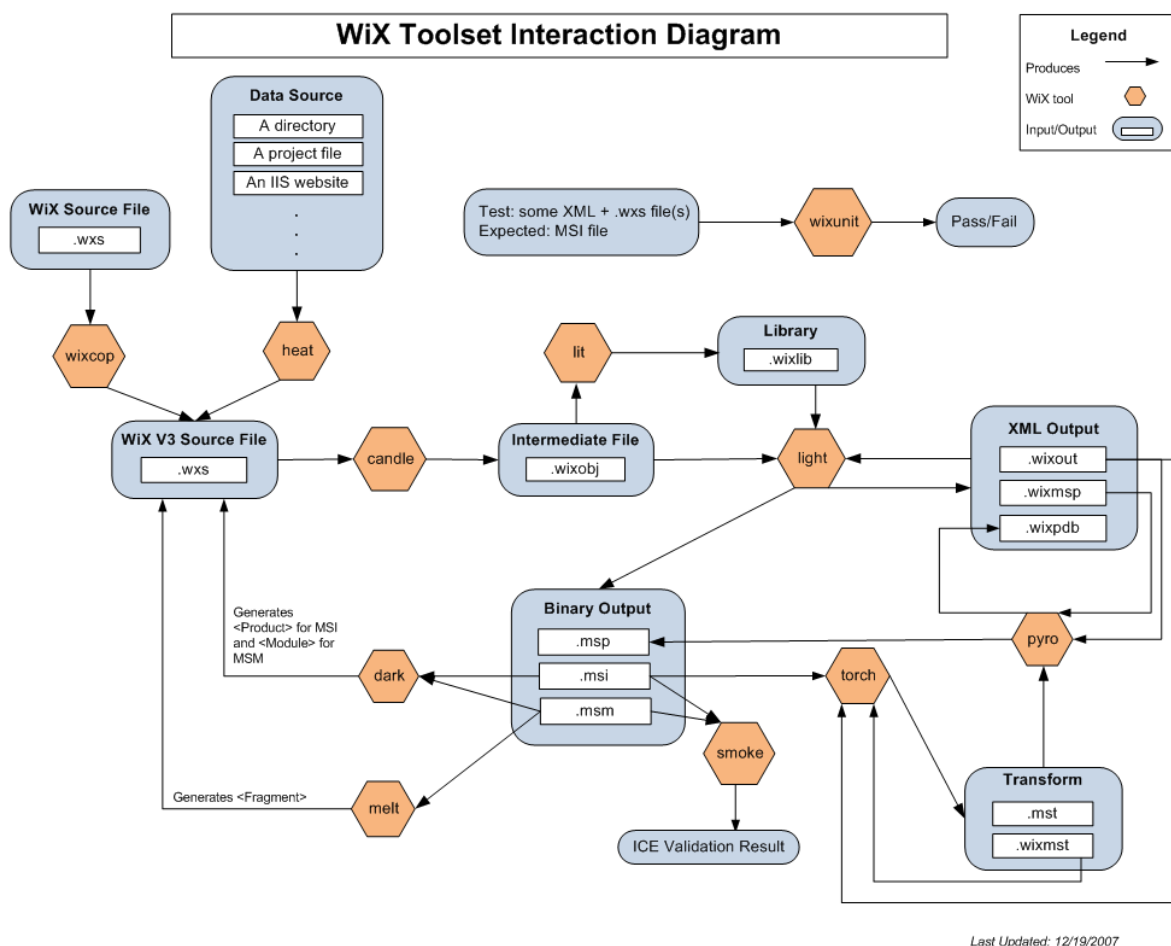
Oprema temelji na graditvi paketa iz izvorne kode, ki je sestavljena iz enega ali več dokumentov XML. Programsko opremo WiX sestavlja več orodij, ki jih lahko enostavno vgradimo v proces graditve celotne programske opreme, ki se izdeluje. Microsoft je razvil tudi poseben dodatek za IDE v Microsoftovem okolju *Visual Studio*. Ta omogoča direktno izdelavo datotek XML kar iz osnovnega razvijalčevega okolja.

Dodatek doda v Microsoftovo okolje *Visual Studio* novo vrsto projekta –projekt WiX. S tem se pridobi barvanje izvorne kode, preverjanje pravilnosti sintakse, tako imenovani *IntelliSense*³ ipd.

4.1 Način graditve paketa

Graditev paketa .msi s pomočjo opreme WiX je zelo podobno graditvi običajne programske opreme, pisane recimo v programskem jeziku C++. Tudi tu je potrebno najprej prevesti dokumente XML v neko vmesno obliko, sledi pa še povezovanje datotek v vmesni obliki v končni namestitveni paket ali v posebne knjižnice WiX, ki se jih lahko kasneje vključi pri povezovanju.

³ IntelliSense je Microsoftova implementacija samodokončevanja znanih izrazov, uporabljena predvsem v razvojnem okolju Microsoft Visual Studio.



Vir: <http://wix.sourceforge.net/manual-wix3/tools.htm>, november 2010

Slika 6: Orodja programskega paketa WiX

4.1.1 Prevajanje (v vmesno obliko) – orodje **Candle.exe**

Služi za prevajanje datotek izvorne kode namestitvenega paketa v vmesne datoteke, ki imajo praviloma končnico `.wixobj`. Take datoteke se lahko poveže v namestitveni paket ali v posebne knjižnice WiX za kasnejšo uporabo. Prevaja se lahko le datoteke, ki vsebujejo XML-element `<Wix>` (glej Format XML, stran 28).

4.1.2 Povezovanje

V `.msi` paket – orodje **Light.exe**

Vmesne datoteke, zgrajene z orodjem *Candle.exe*, je potrebno povezati v namestitveni paket. Pri tem se poleg predhodno prevedenih datotek poveže tudi knjižnica WiX z osnovnimi simboli in funkcijami. Zraven se lahko povezuje tudi knjižnice WiX drugih proizvajalcev ipd.

Za lokalizacijo se v ukazno vrstico povezovanja lahko poleg navede tudi datoteke, namenjene lokalizaciji namestitvenega paketa (glej tudi Vrste datotek izvorne kode WiX, stran 29).

V knjižnico WiX za kasnejšo uporabo pri graditvi drugih paketov – orodje **Lit.exe**

Orodje je namenjeno izdelavi knjižnic WiX iz vmesnih, že prevedenih datotek za kasnejšo uporabo. Datoteke knjižnice WiX imajo praviloma končnico .wixlib.

4.1.3 Ostala orodja WiX-a

Torch.exe

Služi izdelavi transformacijske datoteke med dvema namestitvenima paketoma.

Smoke.exe

S tem orodjem se preverja osnovno pravilnost namestitvenega paketa, če ustreza pogojem tehnologije MSI (relacije med tabelami, pravilnost podatkov ipd.)

Insignia.exe

S tem se lahko digitalno podpisuje posamezne dele namestitvenega paketa in sam paket (glej Tehnologija MSI omogoča, stran 9).

4.2 Izvorna koda za graditev namestitvenega paketa

Tudi pri izvorni kodi za WiX se lahko opremo na programski jezik C++: sestavljena je iz enega ali več datotek v formatu XML. Te lahko vključujejo tudi druge datoteke izvorne kode.

4.2.1 Format XML

Izvorna koda predstavlja zapis v formatu XML. Osnovne zakonitosti sheme – imenski prostor je podan s podatkom <http://schemas.microsoft.com/wix/2006/wi>.

Osnovni elementi sheme so.

Wix

kjer je treba naštetih lastnosti sheme XML oziroma imenskih prostorov. Primer je prikazan na sliki 7.

```

<Wix xmlns=http://schemas.microsoft.com/wix/2006/wi
  xmlns:util="http://schemas.microsoft.com/wix/UtilExtension">
  <Product Id="$(var.PRODUCT_GUID)"
    Name="$(var.PRODUCT_NAME)"
    Language="1060" Codepage="1250"
    Version="$(var.PRODUCT_VERSION)"
    Manufacturer="$(var.COMPANY_NAME)"
    UpgradeCode="$(var.UPGRADE_GUID)">
    <Package Id="*"
      InstallerVersion="300"
      Compressed="yes"
      InstallScope="perMachine"
      SummaryCodepage="1250"/>
    ...
  </Product>
</Wix>

```

Slika 7: Primer elementa Wix

Vsak projekt WiX, iz katerega se zgradi namestitveni paket, mora biti sestavljen iz vsaj ene datoteke, ki vsebuje element Wix. Ravno tako mora obstajati tudi točno ena značka "*Product*" v okviru enega izmed elementov Wix.

Include

Element, ki služi za označevanje datoteke izvorne kode za namene vključevanja v druge datoteke izvorne kode. Primer:

```
<Include xmlns="http://schemas.microsoft.com/wix/2006/wi">
```

WixLocalization

Element služi označevanju datoteke za namene lokalizacije – prevoda v drug jezik. Obvezna lastnost poleg sheme XML je še "*Culture*", ki pove, kateremu jezikovnemu področju je datoteka namenjena. Ta se upošteva pri namestitvi na uporabnikov sistem.

```
<WixLocalization Culture="de-de"
  xmlns="http://schemas.microsoft.com/wix/2006/localization">
```

4.2.2 Vrste datotek izvorne kode WiX

Osnovne datoteke izvorne kode (.wxs)

To so datoteke, ki imajo na najvišjem nivoju element XML <Wix>. Ta element določa, da se datoteka lahko prevede z orodji WiX.

Datoteke za vključevanje v druge dele izvorne kode (.wxi)

Datoteke vsebujejo vse običajne elemente izvorne kode WiX, ker se vsebina med prevajanjem dejansko vrine na mesto, kjer se tako datoteko vključuje.

Datoteke za vključevanje morajo imeti vsebino med elementom XML na najvišjem nivoju `<Include>` in `</Include>`.

Zaradi lažjega ločevanja z drugimi datotekami imajo posebno končnico `.wxi`.

Datoteke lokalizacije namestitvenega paketa (.wxl)

Število teh datotek je odvisno od števila podprtih jezikovnih skupin.

Nizi, ki so spravljani v teh datotekah, se običajno prikazujejo v uporabniškem vmesniku. Uporablja se jih lahko tudi pri raznih nastavitvah ipd.

Te vrste datotek se z razliko od vključitvenih datotek ne prevaja, pač pa se jih uporabi pri povezovanju v končni namestitveni paket ali pri povezovanju v knjižnico WiX.

Pri graditvi namestitvenega paketa s prevodi v več jezikovnih skupin izvorna koda ne vsebuje več vprogramiranih nizov, pač pa le referenco na niz – ime niza.

Vsaka datoteka lokalizacije mora vsebovati vsa imena – reference nizov in njegovo vrednost – prevod.

Datoteke te vrste morajo imeti vsebino med elementom XML na najvišjem nivoju `<Wixlocalization>` in `</Wixlocalization >`.

Vmesne datoteke (.wixobj)

So produkt prevajanja. Služijo za povezovanje v namestitveni paket ali v knjižnice Wix (glej Prevajanje (v vmesno obliko) – orodje `Candle.exe`, stran 27).

Wix knjižnice (.wixlib)

Datoteke te vrste vsebujejo dele prevedene kode WiX, namenjene lažji graditvi namestitvenega produkta. Te datoteke se lahko uporabi pri graditvi različnih namestitvenih paketov (glej Povezovanje, stran 27).

4.2.3 Direktive za predprocesno graditev

WiX omogoča uporabo tako imenovanih predprocesorskih direktiv, s pomočjo katerih se lahko krmili prevajanje iste izvorne kode na različne načine. Tako prevajanje se krmili bodisi s pomočjo posebnih stikal prevajalnika `Candle.exe`, s pomočjo različnih vsebin vključujočih datotek ali s pomočjo različnih nastavitvev systemskega okolja.

Predprocesorske direktive so označene s posebnim elementom `<? direktiva?>`

```

<?ifndef WixMajorMinor?>
  <?error WixMajorMinor spremenljivka je ze dolocena?>
<?endif?>
<?define WixMajorMinor = "3.0" ?>
<?ifndef WixBuildNum?>
  <?error WixBuildNum spremenljivka je ze dolocena?>
<?endif?>
<?define WixBuildNum = "5419" ?>
<?ifndef WixVersion?>
  <?error WixVersion spremenljivka je ze dolocena?>
<?endif?>
<?define WixVersion = "$(var.WixMajorMinor).$(var.WixBuildNum).0" ?>

```

Slika 8: Primer predprocesorskih direktiv

WiX omogoča predvsem tri vrste direktiv:

- za krmiljenje – prevajanje na več načinov (*ifdef*, *ifndef*, *else...*).
- direktive za nastavljanje vrednosti spremenljivk glede na okolje prevajanja (*define*).
- ostale direktive, kot so na primer direktive za brezpogojno zaustavitev prevajanja ipd. (*error*, *warning...*).

Na direktive, označene z značko XML `<?define ?>`, se lahko sklicuje tudi v izvorni kodi, ker ta direktiva definira posebno konstanto. Glede na primer na sliki 8, se lahko poda WiX verzijo kar iz lastnosti tekstovnega gradnika na pozdravnem oknu namestitvenega paketa:

```

<Publish Dialog="WelcomeDlg"
  Control="VersionText"
  Value="$(var.WixVersion)"
  Order="1">
</Publish>

```

4.2.4 Lokalizacija

Uporabniški vmesnik je lahko tudi lokaliziran. Grafični vmesnik je v osnovi v angleškem jeziku. Vsa sporočila, ki se prikazujejo, so vgrajena z referencami. Za prevod v drug jezik je dovolj, da se prevede osnovni jezik lokalizacije v angleškem jeziku ter spremeni lastnost "Culture" glavnega elementa "Wixlocalization" v ustrezni niz, ki predstavlja jezikovno skupino. Niz, ki predstavlja jezikovno skupino, mora biti zapisan po standardu [ISO-639-1](#).

Da se prikazujejo sporočila v pravilni jezikovni skupini, mora vsebovati element "Product" tudi lastnost "Language" in lastnost "Codepage". Podobno velja tudi za element "Package", ki mora vsebovati lastnost "SummaryCodepage".

Vrednost lastnosti "Language" predstavlja identifikacijsko številko jezikovne skupine (okrajšava LCID).

Vrednost "Codepage" in "SummaryCodepage" lastnosti mora vsebovati številko kodne tabele znakov za prikazovanje sporočil.

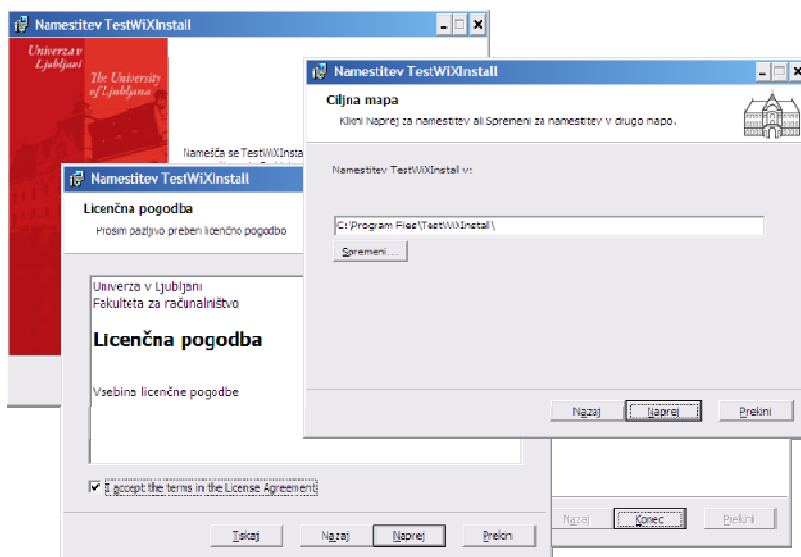
Primer teh nastavitvev prikazuje Slika 7: Primer elementa Wix na strani 29.

4.3 Dodatki (angl. Extensions)

Dodatki so posebne programske knjižnice, ki se vključujejo v namestitveni paket med graditvijo – v času povezovanja.

4.3.1 Uporabniški vmesnik

Uporabniški vmesnik namestitve je največkrat prvi stik programske opreme s kupcem – bodočim uporabnikom, zato je zelo pomembno, kako je ta narejen, da že ob prvem pogledu na programsko opremo dobi kupec dobro mnenje o nakupu.



Slika 9: Uporabniški vmesnik

Vsaka namestitev, ki se ne izvaja v tistem načinu, uporablja grafični vmesnik.

Osnovni del grafičnega vmesnika je knjižnica WiX. Ta se v fazi povezovanja vstavi v namestitveni paket.

Grafični vmesnik je sestavljen kot ostala okna in dialogi v okolju Windows iz programskih gradnikov, ki se lahko odzivajo na določene dogodke (npr. odziv na klik na gumb ipd.).

Če se želi uporabniški vmesnik prilagoditi posebnim potrebam projekta, je potrebno sestaviti ali prirediti datoteke izvorne kode grafičnega vmesnika. V praksi to izgleda običajno tako, da se priredi kopijo originalne izvorne kode recimo pozdravnega dialoga in to kopijo se vključi med ostalo izvorno kodo.

Največkrat so uporabljeni pozdravni dialog, dialog za izbiro ciljne mape namestitve, izbiro možnosti namestitve ipd.

WiX ima nekaj vrst grafičnih vmesnikov že vnaprej določenih:

- WixUI_Mondo vsebuje dialoge, ki omogočajo naprednejše načine namestitve; WixUI_Mondo je primeren za namestitve, ki so porazdeljene po funkcionalnostih, kjer

se tipično za samo delovanje programa ne potrebuje nameščenih vseh funkcionalnosti. Uporabnik lahko izbira med tipično in celotno namestitvijo;

- WixUI_Advanced omogoča namestitev z "enim klikom" ob možnosti izbire ciljne mape in funkcionalnosti;
- WixUI_FeatureTree je poenostavljena možnost WixUI_Mondo. Ta ne vsebuje tipa namestitve, ampak po potrditvi licenčne pogodbe nadaljuje z izbiro funkcionalnosti. Primeren je predvsem za namestitve, porazdeljene po funkcionalnostih, kjer se običajno nameščajo vse funkcionalnosti;
- WixUI_InstallDir omogoča le izbiro ciljne mape namestitve;
- WixUI_Minimal omogoča prikaz le licenčne pogodbe. Primeren je za namestitve z eno funkcionalnostjo in namestitve, kjer ni možnosti izbire ciljne mape.

Izvorna koda grafičnega vmesnika vključuje element <UI>, ki mora imeti nastavljeno vrsto vmesnika.

```
<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi">
  <Fragment>
    <UI Id="TestInstallDialog">
      <Property Id="WixUI_Mode" Value="Mondo" />
      ...
    </UI>
  </Fragment>
</Wix>
```

Slika 10: Obvezni elementi datoteke izvorne kode vključitve grafičnega vmesnika

4.3.2 WiX utils

Dodatek omogoča predvsem nastavitve uporabniških pravic nad nameščenimi resursi v ciljnim računalniku, kreiranju novih uporabnikov ipd. Poleg tega se ga lahko koristi za zapiranje programov med nameščanjem in za spreminjanje nastavitve storitev.

Poseben del tega dodatka je vmesnik za datoteke v formatu XML, s pomočjo katerega lahko v času nameščanja popravljamo obstoječe datoteke XML ali izdelujemo nove.

Imenski prostor in vključitev dodatka je definiran:

```
<WixExtension Include="WixUtilExtension.wixext"
  xmlns:util="http://schemas.microsoft.com/wix/UtilExtension"/>
```

4.3.3 Namestitev gonilnikov

Gonilnike in posebne filtre datotečnega sistema ter drugih naprav se namešča s pomočjo tako imenovanih informacijskih namestitvenih datotek (končnica .inf).

Wix v ta namen se koristi Microsoftov programski modul [DIFxApp](#) (angl. kratica za *Driver Install Frameworks for Applications*), ki zna s pomočjo datoteke .inf namestiti gonilnik.

Povezava tega modula z WiX-om je vzpostavljena z vključitvijo imenskega prostora:

```
<Wix xmlns='http://schemas.microsoft.com/wix/2006/wi
      xmlns:difx='http://schemas.microsoft.com/wix/DifxAppExtension'>
```

in pri prevajanju in povezovanju je potrebno podati referenco na programsko knjižnico *WixDifxAppExtension.dll*.

Datoteke gonilnika je treba napovedati v svoji komponenti (glej sliko 11, na strani 34), ki mora vsebovati še informacijsko datoteko in katalog⁴.

```
<Directory Id='DirFilterDir' Name='DirFilter'>
  <Component Id='DirFilterDriver'
    Guid='8c64e674-5476-46e4-93cd-balae78622df'>
    <File Id='DirFilterSYS'
      Name='DirFilter.sys'
      DiskId='1'
      Source='DirFilter.sys'
      KeyPath='yes' />
    <File Id='DirFilterINF'
      Name='DirFilter.inf'
      DiskId='1'
      Source='DirFilter.inf' />
    <File Id='DirFilterCAT'
      Name='DirFilter.cat'
      DiskId='1'
      Source='DirFilter.cat' />
    <difx:Driver Legacy='yes' AddRemovePrograms='no' />
  </Component>
</Directory>
```

Slika 11: Vključitev datotek za namestitev gonilnika

4.3.4 Netfx Shema

Namenjena je upravljanju z ogrodjem NET. Omogoča ugotavljanje nameščenih verzij, posodobitev ipd. in tudi samo nameščanje ogrodja NET.

Shemo se vključi z nizom

```
xmlns:NetFx="http://schemas.microsoft.com/wix/NetFxExtension"
```

v elementu <Wix>.

Dejansko ogrodje se priključi datoteki, ki ga uporablja.

⁴ Katalog specificira skupino datotek, namenjeno namestitvi gonilnika. Več o tem na [Microsoft MSDN straneh](#).

4.4 Orodja za izdelavo instalcijskega paketa MSI s pomočjo WiX-a

4.4.1 MS Visual Studio

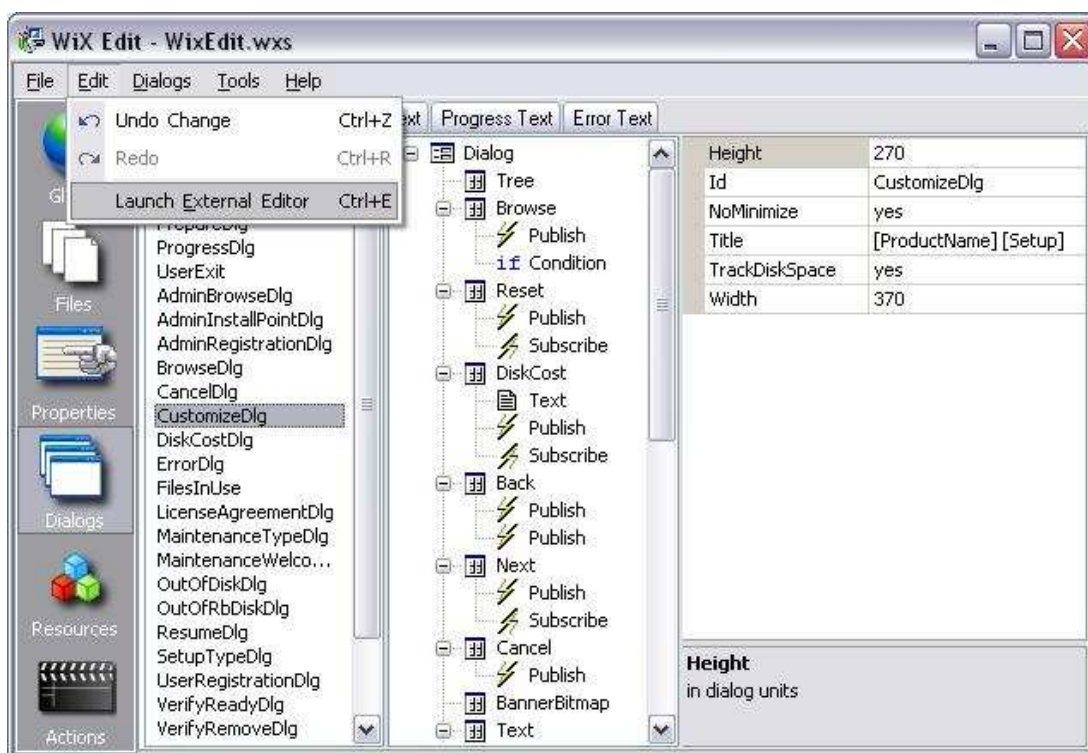
Microsoft *Visual Studio* ima programski dodatek, imenovan "*Votive*", s pomočjo katerega *Visual Studio* dobi novo vrsto projekta. S tem se pridobi vse dobre lastnosti *Visual Studio*: obarvanje izvorne kode, predlaganje lastnosti elementov XML, prevajanje in graditev z enim klikom miške ipd.

Dodatek, tako kot sama tehnologija WiX, ni komercialne narave.

4.4.2 WiXEdit

WiXEdit je orodje, namenjeno graditvi namestitvenih paketov in vključujočih modulov iz datotek XML z elementi WiX-a. Prednost je predvsem v izdelavi grafičnega vmesnika po principu "dobiš kar vidiš" (angl. kratica WYSIWYG – what you see is what you get).

Orodje ni komercialne narave in se ga še aktivno razvija.



Vir: <http://wixedit.sourceforge.net/screenshots/>, november 2010

Slika 12: WixEdit orodje

4.4.3 Ostala orodja

Na splošno se lahko uporablja vsako orodje za izdelavo tekstovnih datotek kot orodje za pisanje izvorne kode WiX, vendar je za razvijalca mnogo težje delo.

4.5 Primer uporabe



Slika 13: Primer izdelave namestitvenega paketa s pomočjo WiX-a

Na primeru je prikazan enostaven namestitveni paket, narejen z uporabo orodij WiX. Uporablja grafični vmesnik za vnos osnovnih informacij namestitve (lokacija – mapa namestitve). Poleg tega namesti navadno izvršno datoteko, datoteko pomoči in namesti eno enostavno storitev. Poleg tega zabeleži tudi nekaj nastavitvev v sistemskem registru.

Projekt je razdeljen v več datotek:

- osnovna datoteka `TestWixInstall.wxs`: vsebuje osnovne elemente, kot so Projekt, imena ciljnih map in akcij. Grafični vmesnik je tudi referenciran v tej datoteki;
- datoteka `TestWixKomponente.wxi`: vsebuje spisek datotek, ki se nameščajo. Nastavitve registra so dodeljene komponenti z imenom "*CompIzvrснаDatoteka*". Nastavitev storitve je vključena v komponento, ki vsebuje izvršno datoteko storitve (komponenta "*testService*").
- datoteke grafičnega vmesnika: grafični vmesnik je prirejen programu, ki se namešča. Uporabljena je datoteka izvirne kode WiX-a "*InstallDialog.wxs*" in je preimenovana v "*TestInstallDialog.wxs*". V tej datoteki je vključeno posebno okno grafičnega vmesnika "*TestWelcomeDlg.wxs*". Ker je to le testni primer, je tudi tu uporabljena preimenovana ter spremenjena datoteka originalne izvirne kode WiX-a;
- datoteka lokalizacije v slovenščino: prevod originalne angleške verzije in še datoteka posebnih sporočil.

4.5.1 Graditev namestitvenega paketa

Graditev paketa, ki ima vgrajeno nadgradnjo, je zahtevnejša od običajne graditve paketa. Potrebno je zagotoviti novo identifikacijsko številko paketa, novo številko verzije paketa ter nove verzije vseh izvršnih datotek v paketu. Ker se paket zgradi s pomočjo orodja *Visual Studio*, ki tega ne zna samodejno spremeniti, je potrebno pripraviti posebno okolje za graditev.

Zaradi poenostavitve graditve je najbolje vse parametre, ki so potrebni za uspešno nadgraditev paketa, vgraditi tako, da se vrednosti dodeljuje kar se da dinamično. Na tem primeru se ti parametri nastavijo v sistemsko okolje, v katerem se potem požene *Visual Studio*, ta pa med graditvijo zamenja reference z ustrežno vrednostjo. Na primeru je uporabljen tudi skriptni jezik *Python* za popravljanje oznake verzije izvršne datoteke.

Slika 14 prikazuje vse ukaze, ki nastavljajo okolje za graditev. Tako so potrebni minimalni posegi za graditev na drugem sistemu.

```

:: =====
:: File:          $URL: $
:: Author:       Andrej Jamsek
:: =====
setlocal
@echo off
echo Nastavitev parametrov...
set VS_BASE=c:\Program Files\Microsoft Visual Studio 8\Common7\IDE\
set UTILS_PATH=C:\sys\UxUtils\
set path=%path%;c:\Program Files\Microsoft Visual Studio 8\Common7\Tools\
set CWD=%CD%

set PRODUCT_VER=
if not %1 == __ set PRODUCT_VER=%1
if !_%PRODUCT_VER% == __ set PRODUCT_VER=1,0,0,0

echo %PRODUCT_VER% |%UTILS_PATH%sed "s/\"//g" > pkgver.txt
set /p PRODUCT_VER= < pkgver.txt
echo %PRODUCT_VER% |%UTILS_PATH%sed -e "s/,/./g" -e "s/\"//g" > pkgver.txt
set /p PRODUCT_WIXVER= < pkgver.txt
echo Generiranje GUID za paket. Uporablja se orodje Visual Studia UUIDGEN.EXE (mora biti v poti!)
uuidgen -c -opkgguid.txt
set /p PRODUCT_ID= < pkgguid.txt
title RunVS %PRODUCT_VER%

echo.
echo PRODUCT_VER    =%PRODUCT_VER%
echo PRODUCT_WIXVER=%PRODUCT_WIXVER%
echo PRODUCT_ID     =%PRODUCT_ID%

rem for %%i in (TestWinService\TestSvc IzvrснаDatoteka\izvrсна) do copy %%i.rc %%i.old
for %%i in (TestWinService\TestSvc IzvrснаDatoteka\izvrсна) do python setver.py "%PRODUCT_VER%" "%%i.rc" "%%i_rc.tmp"

echo Pogonja se Visual Studio...
if "%_VS_BASE%" == "" set VS_BASE=c:\Program Files\Microsoft Visual Studio 8\Common7\IDE\
"%VS_BASE%devenv.exe" IzvrснаDatoteka.sln
endlocal

```

Slika 14: Skripta za zaganjanje orodja za graditev paketa

4.5.2 Opis posebnosti

- Iskanje nastavitve (prikazuje slika 15) se določi spremenljivki: prvo iskanje je namenjeno iskanju stare lokacije namestitve v sistemskem registru. Če ta ne obstaja, vrednost spremenljivke "INSTALLDIR" ne dobi vrednosti oziroma postane prazna. Drugo iskanje pa preverja, če obstaja ključ v sistemskem registru, njegovo vrednost pa shrani v spremenljivko "TEST_REG_PROP".

```

<Property Id="INSTALLDIR" Secure="yes">
  <RegistrySearch Id="FindInstallLocation"
    Root="HKLM"
    Key="Software\Microsoft\Windows\CurrentVersion\
      Uninstall\[UPGRADEFOUND]"
    Name="InstallLocation"
    Type="raw" />
</Property>
<Property Id="TEST_REG_PROP" Secure="yes">
  <RegistrySearch Id="FindTestReg"
    Root="HKLM"
    Key="SOFTWARE\$ (var.COMPANY_NAME)\$(var.PRODUCT_NAME)\var1"
    Name="maxsize"
    Type="raw" />
</Property>

```

Slika 15 Iskanje nastavitve v sistemskem registru

- Namestitev storitve: vključena je kar v komponento, ki vsebuje izvršno datoteko storitve. Slika 16 prikazuje vključitev namestitve storitve. V ta namen je rezervirana značka XML "*ServiceInstall*", ki se dodeli vse lastnosti storitve. Na sliki je tudi razvidno, da sta ime storitve in njen opis, ki se prikazuje med sistemskimi nastavitvami storitev, lokalizirana. Določeno je tudi, da se med namestitvijo storitev tudi zažene ter med odstranitvijo ustavi.

```

<Component Id="testService" Guid="56DACE72-74E4-44B7-A451-
2A51C9D0F968">
  <File Id="TestServiceExe" Name="TestWinService.exe"
Source="..\Out\Bin\TestWinService.exe" DefaultLanguage="1060"
KeyPath="yes" Vital="yes"/>
  <ServiceInstall
    Id="TestServiceInstaller"
    Type="ownProcess"
    Name="TestWixService"
    DisplayName="!(loc.TestServiceDisplayName)"
    Description="!(loc.TestServiceDescription)"
    Start="demand"
    Account="LocalSystem"
    ErrorControl="normal"
    Interactive="no" />
    <ServiceControl Id="StartService" Name="TestWixService"
Start="install" Wait="yes" />
    <ServiceControl Id="StopService" Name="TestWixService"
Stop="both" Wait="yes" Remove="uninstall" />
  </Component>

```

Slika 16: Del izvorne kode za namestitev storitve

- Zapiranje običajne izvršne datoteke, če je ta zagnana med odstranitvijo (in med nadgraditvijo, kar prikazuje Slika 17. Za vgraditev te funkcije je potrebno projektu nastaviti referenco na programsko knjižnico "*WiXUtils*" (glej WiX utils, stran 33).

```

<util:CloseApplication
  Id="CloseMyApp"
  Target="[MyAppExe]"
  CloseMessage="yes"
  Description="!(loc.MyAppStillRunning)"
  RebootPrompt="no"
  ElevatedCloseMessage="no"
/>
...
<InstallExecuteSequence>
  ...
  <Custom Action="WixCloseApplications" Before="RemoveFiles">REMOVE =
  "ALL"</Custom>
  ...
</InstallExecuteSequence>

```

Slika 17: Zaustavljanje programa med odstranjevanjem programa

- Nastavitve na nadgradnjo:
 - za namene nadgradnje je uporabljena značka XML "*Upgrade*" z identifikacijsko številko nadgraditve, ki je določena z vnaprej določeno konstanto (glej sliko 18). Za pravilno nadgraditev je vključena tudi posebna akcija "*RemoveExistingProducts*". Na primeru je vprogramirana velika nadgradnja, ki najprej odstrani staro verzijo v transakcijski fazi, nato pa se izvrši namestitev nove verzije.
 - Da se namestitveni paket nadgradi, pa morajo veljati naslednje lastnosti:
 - če je nameščena verzija paketa med "*Minimum*" in "*Maximum*", potem namestitev napolni spremenljivko "*UPGRADEFOUND*" z identifikacijsko številko nameščenega paketa, sicer ostane vrednost spremenljivke nespremenjena;
 - "*MigrateFeature*" določa, da se vse že nameščene funkcijske enote nadgradijo.

```

<Upgrade Id="$(var.UPGRADE_GUID)">
  <UpgradeVersion
    Minimum="1.0.0.0" Maximum="99.0.0.0"
    Property="UPGRADEFOUND"
    Language="1060"
    MigrateFeatures="yes"
    IncludeMinimum="yes"
    IncludeMaximum="no" />
</Upgrade>
...
<InstallExecuteSequence>
  ...
  <RemoveExistingProducts After="InstallInitialize"/>
  ...
</InstallExecuteSequence>

```

Slika 18: WiX elementi za nadgradnjo

4.6 Primerjava WiX z ostalimi orodji

Orodje WiX se uporablja na podoben način kot graditev programske opreme – iz datotek izvorne kode, ki se najprej prevaja in nato poveže v namestitveni paket. To je bistvena razlika

glede na ostala orodja za graditev namestitvenih paketov. Nekaj ostalih razlik je navedenih v tabeli.

Tabela 5: Nekaj najpomembnejših razlik med WiX in ostalimi orodji

Podprtost	WiX	InstallShield	Wise
Tekstovni zapis datoteke izvorne kode	Da	Da/Ne	?
Grafični vmesnik za izdelavo izvorne kode namestitve	Ne	Da	Da
Integracija z MS Visual Studiem	Da	Da	Ne
Validacija MSI in MS Logo	Ne	Da	Da
Uporaba spremenljivk okolja operacijskega sistema	Pogojno – z uporabo predprocesorskih direktiv	Da	Da
Razhroščevalnik MSI	Ne	Da	Ne
Izdelava podatkovnih baz	Pogojno – z dodatnimi akcijami	Da (MySQL, Oracle)	Da (MS SQL)
Dinamično dodajanje datotek	Ne	Da	
Podprtost uporabniško določenih akcij	Delno	Da	Da
Izdelava paketa, ki vsebuje več podprtih jezikovnih skupin	Ne	Da	Ne
Izdelava samorazpakirajočega paketa namestitve	Ne	Da	Da

5 Zaključek

Tehnologija MSI se še vedno razvija, vendar predvsem v smeri poenostavitve systemske administracije računalniške programske opreme. Dograditve osnovne tehnologije MSI, kar se tiče graditve in uporabe namestitvenih paketov, so večinoma posledica razvoja operacijskih sistemov oziroma računalniškega okolja in novih platform.

Diplomska naloga je napisana na osnovi izkušenj, ki sem jih pridobil v podjetju, kjer sem zaposlen. Vsekakor so orodja WiX huda konkurenca prodajnim različicam. WiX se še razvija in pridobiva na uporabi iz dneva v dan. Orodja WiX se da enostavno vklopiti v okolje graditve celotnega produkta. Velika prednost WiX-a je tudi v datotekah izvorne kode namestitvenega paketa, ker so to čisto navadne tekstovne datoteke: da se jih lahko spreminjati z vsakim orodjem za urejanje teksta, predvsem pa se njihova narava pokaže pri delu s kontrolo nad različicami izvorne kode (primerjanje različic, vgraditev posebnih lastnosti ipd.).

Menim, da WiX-u manjka še malo prijaznosti, ker le graditev iz datotek XML zamegljuje celotno sliko, kako namestitveni paket izgleda. Zaradi tega so v porastu tudi podporna orodja za izdelavo namestitev s pomočjo WiX-a. Z malo iznajdljivosti se da narediti lep izdelek, ki omogoča tako rekoč vse, kar omogočajo "ta veliki".

Reference

Seznam slik

Slika 1: Namestitev po posameznih fazah	14
Slika 2: Zastopanost orodij za graditev namestitvenega paketa v tehnologiji MSI.....	17
Slika 3: Glavne tabele datotek .msi paketa z medsebojnimi relacijami.....	18
Slika 4: Orodje za analizo dnevniških datotek namestitve	24
Slika 5: Windows Clean Up Utility	25
Slika 6: Orodja programskega paketa WiX	27
Slika 7: Primer XML elementa Wix	29
Slika 8: Primer predprocesorskih direktiv	31
Slika 9: Uporabniški vmesnik	32
Slika 10: Obvezni elementi datoteke izvorne kode vključitve grafičnega vmesnika	33
Slika 11: Vključitev datotek za namestitev gonilnika.....	34
Slika 12: WixEdit orodje.....	35
Slika 13: Primer izdelave namestitvenega paketa s pomočjo WiX-a	36
Slika 14: Skripta za zaganjanje orodja za graditev paketa.....	37
Slika 15: Iskanje nastavitev v sistemskem registru.....	38
Slika 16: Del izvorne kode za namestitev storitve.....	38
Slika 17: Zaustavljanje programa med odstranjevanjem programa.....	39
Slika 18: WiX elementi za nadgradnjo	39

Seznam tabel

Tabela 1: Nekaj orodij za graditev namestitvenih paketov	6
Tabela 2: Odvisnosti med posameznimi ID številkami namestitve	11
Tabela 3: Orodja za graditev namestitvenih paketov tehnologije MSI.....	16
Tabela 4: Nekaj vnaprej določenih spremenljivk tehnologije MSI	20
Tabela 5: Nekaj najpomembnejših razlik med WiX in ostalimi orodji	40

Literatura

1. Bob Baker: "The Official InstallShield for Windows Installer Developer's Guide", Wiley, February 15, 2001.
2. Ian Griffiths, Jon Flanders, Chris Sells: "Mastering Visual Studio .NET", O'Reilly Media, marec, 2003, 416 strani.
3. Alex Martelli: "Python in a Nutshell", Second Edition. O'Reilly Media, julij, 2006, 742 strani.
4. Phil Wilson: "The Definitive Guide to Windows Installer", Apress, April 26, 2004.

Viri

1. **About the Installer Database:** Microsoftove MSDN strani [URL: <http://msdn.microsoft.com/en-us/library/aa367441%28v=VS.85%29.aspx>], oktober 2010.
2. **Code page:** Wikipedia [URL: http://en.wikipedia.org/wiki/Code_page], oktober 2010.
3. **Extensible Markup Language (XML):** [URL: <http://www.w3.org/XML/>], oktober 2010.
4. **Language codes:** Microsoftove MSDN strani [URL: <http://msdn.microsoft.com/en-us/library/ms533052%28VS.85%29.aspx>], oktober 2010.
5. **List of ISO 639-1 codes:** Wikipedia [URL: http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes], oktober 2010.
6. **LCID:** Microsoftove MSDN strani [URL: <http://msdn.microsoft.com/en-us/global/bb964664.aspx>], oktober 2010.
7. **MSIEXEC.EXE:** Microsoftove podporne strani [URL: <http://support.microsoft.com/kb/314881>], oktober 2010.
8. **Roadmap to Windows Installer Documentation:** Microsoftove MSDN strani [URL: <http://msdn.microsoft.com/en-us/library/aa371366%28v=VS.85%29.aspx>], oktober 2010.
9. **VBScript:** Wikipedia [URL: <http://en.wikipedia.org/wiki/VBScript>], oktober 2010.
10. **Windows installer:** Microsoftove MSDN strani [URL: <http://msdn.microsoft.com/en-us/library/cc185688%28VS.85%29.aspx>], oktober 2010.
11. **Windows Installer XML (WiX) toolset:** Skupina Sourceforge.net [URL: <http://wix.sourceforge.net/>], oktober 2010.
12. **Wix:** [URL: <http://en.wikipedia.org/wiki/WiX>], oktober 2010.
13. **WixEdit:** [URL: <http://wixedit.sourceforge.net/>], oktober 2010.
14. **WiX Extensions:** MindCapers [URL: **WiX Toolset:** CodePlex Open Source Project Community [URL: <http://wix.codeplex.com/>], oktober 2010.
15. **Wix Localization:** [URL: <http://www.tramontana.co.hu/wix/lesson2.php#2.4>], oktober 2010.
16. **Wix Schema:** Wikipedia [URL: http://www.wixwiki.com/index.php?title=Wix_Schema], oktober 2010.
17. **WiX tutorial:** [URL: <http://www.tramontana.co.hu/wix/>], oktober 2010.
18. **WiX Extensions:** [URL: http://wix.mindcapers.com/wiki/WiX_Extensions], oktober 2010.