

Microsoft Windows Installer

Johannes Passing



Agenda

1. **Softwareinstallation unter Windows**
2. **Übersicht Windows Installer**
3. **Erstellen von MSI-basierten Setups**
4. **Automatische Installation und Softwareverteilung**

1. Softwareinstallation unter Windows

Johannes Passing



1. Softwareinstallation unter Windows

➤ Aufgaben eines Setup-Programms

- **Installation von Ressourcen**
 - Lokale Dateien
 - Systemdateien (System32 etc)
 - Registry-Einträge
- **Registrierung von COM-Servern**
- **Registrierung von Typbibliotheken**
- **GAC-Registrierungen**
- **Lizenz anzeigen und akzeptieren lassen**
- **Ggf. Lizenz-Schlüssel überprüfen**
- **Versionskontrolle**

✍ **Xcopy-Deployment reicht in den meisten Fällen nicht aus**

1. Softwareinstallation unter Windows

➔ Abhängigkeiten

- **Auflösung von Abhängigkeiten zur Installationszeit in der Regel nicht möglich**

Ein Setup sollte daher

alle ‚kleineren‘ Abhängigkeiten enthalten
(MDAC, MSXML etc.)

Alle ‚großen‘ Abhängigkeiten überprüfen
(Microsoft Office, SQL Server etc.)

1. Softwareinstallation unter Windows

✚ Geteilte Ressourcen

- **Neue Versionen dürfen nicht überschrieben werden**
 - **„DLL-Hell“**
- **Eine Ressource darf nicht entfernt werden, solange sie von nur einem einzigen Programm verwendet wird**
 - **Referenzzählung notwendig**
 - **Problem: Setups verschiedener Hersteller müssen gemeinsame Referenzzählung verwenden**

1. Softwareinstallation unter Windows

➔ Setup-Ansätze

● ‚Klassisch‘, bspw. Wise 9

■ Ein Setup enthält vollständige Logik zur Installation und Deinstallation

- Während Installation wird Log erstellt, anhand dessen deinstalliert werden kann
- Alle Regeln (Versionierung, Referezzählung etc.) müssen befolgt werden

■ Skript-basiert

■ Keine besondere Infrastruktur notwendig

■ Eintrag in Registrierung, sodass über ‚Add/Remove Software‘ deinstallierbar

≠ HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

1. Softwareinstallation unter Windows

➤ Windows Installer

- **Bestandteil von Windows 2000, XP, 2003, ME**
 - **Optional für Windows NT, 95, 98**

- **Bildet Infrastruktur**
 - **Bspw. Wise for Windows Installer basiert auf MSI**

- **Windows Installer Service (WIS)**
 - **Führt Installation und Deinstallation aus**
 - Implementiert gemeinsame Installations/Deinstallations-Logik für alle Setups
 - Implementiert Referenzzählung etc.
 - **Setups (MSI) sind zumeist nur deklarativ**

2. Übersicht Windows Installer

Johannes Passing



2. Übersicht Windows Installer

➔ Dateitypen

- **MSI**
 - Vollständige Installation
- **MSP**
 - Patch
- **MST**
 - Transform
- **MSM**
 - Merge Module

2. Übersicht Windows Installer

➔ Struktur

● Components

- (kleine) Sammlung von u.a.
 - Dateien
 - Registrierungs-Einträgen
- Systemweit eindeutig identifizierbar mittels GUID
- Können von verschiedenen Setups gemeinsam genutzt werden
- Bilden kleinste installierbare Einheit
 - Einheit der Referenzzählung
- Eine der enthaltenen Ressourcen bildet den KeyPath
 - Wird für Reparatur verwendet

✍ **Eine Ressource sollte niemals Mitglied mehrerer Components sein**

2. Übersicht Windows Installer

➤ Struktur (Fortsetzung)

● Features

- **Gruppe von Components**

- Eine Component darf in mehreren Features enthalten sein

- **Bildet eine (optional) vom Benutzer installierbare Einheit**

- **Nur innerhalb eines Setups eindeutig identifizierbar**

- **Kann sich in verschiedenen Status befinden**

- Not installed
 - Advertised
 - Run from source
 - Installed

2. Übersicht Windows Installer

➤ Struktur (Fortsetzung)

● Product

- Gruppe von Features

- Entspricht einer MSI-Datei

■ ProductVersion

- Entspricht der Version der zu installierenden Software

- Wird bei Updates geändert

■ ProductCode (GUID)

- Systemweit eindeutige Kennung des Products

- Wird bei Major Updates geändert

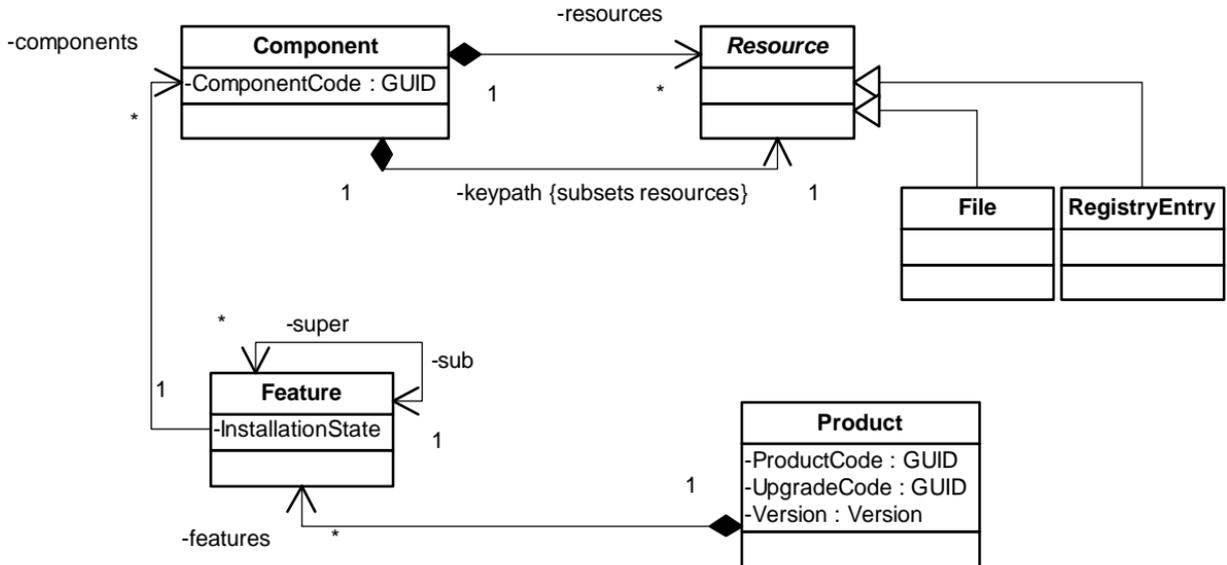
■ UpgradeCode (GUID)

- Über alle Versionen einer Software stabile Kennung

- Wird niemals geändert

- Wird verwendet, um bereits installierte Products anderer Versionen zu finden

2. Übersicht Windows Installer



2. Übersicht Windows Installer

➔ Struktur (Fortsetzung)

● Properties

- **Während der Installation verwendete Variablen**
 - Bspw INSTALDIR, SOURCELIST, ALLUSERS
- **Mithilfe von Properties können Konditionen verfasst werden**
 - Bspw konditionale Installation einer Datei abhängig vom Betriebssystem
- **Mittels msixexec können Property-Werte übergeben werden**

2. Übersicht Windows Installer

➤ Benutzerschnittstelle

- **Rein deklarativ**
- **Es werden Dialoge deklariert**
 - Anordnung von (vordefinierten) Controls
 - Zuordnung von Events

 - **Beispiel: Dialog zur Auswahl des Installationsordners**
- **Anhand von Sequenztabelle werden die Dialoge zu einem Wizard zusammengefasst**
 - **Es existieren verschiedene Sequenzen für Admin- und Benutzerinstallation**
- **Die Benutzerschnittstelle kann zum Installationszeitpunkt in verschiedenem Umfang aktiviert oder vollständig deaktiviert werden (User Interface Levels, Property UILEVEL)**

2. Übersicht Windows Installer

➔ CustomActions

- **Die meistgenutzten Aktionen werden bereitgestellt**
 - Installation von Ressourcen
 - Registrierung von COM-Servern und Typbibliotheken
 - Shortcuts erstellen
 - Etc

- **Mittels CustomActions können weitere Funktionen genutzt werden**
 - (native) DLL mit bestimmten Exports
 - Von `System.Configuration.Install.Installer` abgeleitete .Net-Klasse
 - VB-Skripte
 - ...weitere

2. Übersicht Windows Installer

➤ Installationstypen

● **Erst-/Vollinstallation**

■ **Installation eines MSI-Paketes**

- Inkl. Features und Components
- Ggf. unter Benutzung von Advertisement

■ **Per-machine Installation**

- Installierte Software steht allen Benutzern zur Verfügung
Verwendung des ‚All Users‘ Profil sowie HKLM-Hive
- Erfordert Administrator-Rechte

■ **Per-user Installation**

- Installierte Software steht nur dem installierendem Nutzer zur Verfügung
Verwendung des Benutzerprofils sowie HKCU-Hive
- Problem: Als Nicht-Administrator fehlen dem Benutzer ggf. Rechte, um die Installation durchführen zu können
☞ Managed Applications als Lösung

2. Übersicht Windows Installer

➤ Installationstypen (Fortsetzung)

● Major Updates

- Umfangreiche Änderungen
- ProductCode und ProductVersion werden geändert
- UpgradeCode bleiben stabil

- Installation älterer Pakete über neuere Versionen kann verhindert werden
- Deinstallation von Paketen älterer Version kann erzwungen werden

2. Übersicht Windows Installer

✦ **Installer Database**

- **MSI-Datei enthält relationale Datenbank**
 - Alle Informationen (außer Dateien) werden in Tabellen abgelegt

- **Dateien werden in Cabinet-Datei gespeichert**
 - In MSI eingebettet
 - Externe .cab-Dateien
 - Zerteilen des Setups auf mehrere Dateien in Floppy-Größe etc.

- **Basiert auf COM Structured Storage**
 - Tabelle = Stream
 - Transaktional
 - Mittels SubStorages können MSIs-Dateien weitere MSI-Dateien enthalten

- **Nicht Unicode-basiert**
 - Codepages müssen angewendet werden

2. Übersicht Windows Installer

➔ Windows Installer Service (WIS)

- Als Windows Service implementiert
- Funktionen:
 - Installation von Paketen
 - Deinstallation von Paketen
 - Rollback bei fehlerhafter Installation
- Zur automatischen Softwareinstallation und –verteilung instrumentalisierbar
 - Installiert die über Group Policies verteilte Software
- Zusätzlich: Automation-API
 - Verwaltung der Tabellen mittels SQL

2. Übersicht Windows Installer

➔ Windows Installer Service (WIS) (Fortsetzung)

● **Verwaltet Registrierung über**

- **Installierte Products**
- **Installierte Features**
- **Installierte Components**

```
<HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\<SID>\
```

● **Wird verwendet von**

- **Shell**
 - Startmenü-Einträge, File extensions
- **Control Panel**
 - Add/Remove Software
- **Active Directory Group Policy**
- **System Management Server**
- **msiexec**

2. Übersicht Windows Installer

➔ Phasen einer Installation durch WIS

1. Acquisition

- **Sequenztabellen werden durchlaufen**
 - InstallExecuteSequence
 - InstallUISequence
 - AdminExecuteSequence
 - AdminUISequence
- **Daraus wird Installations- sowie Deinstallationskript generiert**

2. Execution

- **Das erstellte Skript wird WIS übergeben**
- **Skript wird ggf. mit geänderten Rechten ausgeführt**
- **Jede zu ersetzende Ressource wird gesichert**
 - Werden erst bei Commit entfernt
 - Erhöhter Platzbedarf während Installation

2. Übersicht Windows Installer

➔ Phasen einer Installation durch WIS (Fortsetzung)

● 3. Rollback (falls Execution fehlschlug)

- Rollback-Skript wird ausgeführt
- Gesicherte Dateien werden wiederhergestellt

● CustomActions müssen Commit/Rollback-Logik selbst implementieren

2. Übersicht Windows Installer

➤ Install-on-demand, Reparatur

● Idee

- Es werden nur die unbedingt notwendigen Elemente installiert
- Wenn auf ein nicht installiertes Element zugegriffen wird, wird dieses von der Installationsquelle nachinstalliert

✍ Weniger Speicherplatzbedarf

✍ Insbesondere bei automatischer Softwareverteilung sinnvoll

● Installation Source resiliency

- Bei on-demand-install werden die Dateien von der Installationquelle bezogen
- Mittels der Property SOURCELIST können mehrere Quellen angegeben werden

2. Übersicht Windows Installer

➤ Install-on-demand, Reparatur (Fortsetzung)

● Product Advertisement

■ Anwendung scheint installiert

- Startmenüeinträge eingerichtet
- File extensions registriert

■ Erst bei erstmaliger Aktivierung wird die Installation begonnen

■ Basiert auf Unterstützung der Shell

- Startmenü-Shortcuts verwenden Indirektion, die vom WIS aufgelöst wird

■ Einrichtung mit msixexec oder GPO

2. Übersicht Windows Installer

➤ Install-on-demand, Reparatur (Fortsetzung)

● Feature Advertisement

- Einzelne Features eines Products werden on-demand nachinstalliert

● Reparatur

- Bei Zugriff auf Component wird KeyPath geprüft
- Existiert dieser nicht, gilt Component als defekt
- ✍ Component wird erneut installiert

2. Übersicht Windows Installer

➔ Administration mittels Msiexec

● Kommandozeilen-Werkzeug zur

- Installation, Deinstallation, Reinstallation, Reparatur von MSI-Dateien
 - Optional unter Verwendung von Transforms
 - Anpassung der Installation durch Übergeben von Property-Werten
- Anwenden von Patches (MSP-Dateien)

3. Erstellen von MSI-basierten Setups

Johannes Passing



3. Erstellen von MSI-basierten Setups

➤ Erstellen eines neuen Setups

- **Es existieren viele Windows Installer-basierte Programme zur Erstellung von Setups**
 - **Wise for Windows Installer**
 - **InstallShield**
 - **Microsoft VisualStudio.NET**
 - Sehr beschränkter Funktionsumfang
 - **Orca**
 - Bestandteil der Windows 2000 Support Tools
 - Manuelles Editieren der Datenbank-Tabellen
 - **Windows Installer Xml Toolset (Wix)**

3. Erstellen von MSI-basierten Setups

➔ Erstellen eines neuen Setups (Fortsetzung)

● Merge Modules können verwendet werden

- Ähnlich wie MSI
- Können nicht selbstständig installiert werden

✍ Modularisierung

- ✍ Für viele oft verwendete Abhängigkeiten bieten Microsoft, Wise etc. bereits Merge Modules an

3. Erstellen von MSI-basierten Setups

➤ ‚Konvertieren‘ einer klassischen Setup-Lösung in eine Windows Installer-basierte Lösung

- **Durch Wrapping**

- **Durch Observation**

- **Es existieren Werkzeuge, die dies automatisch erledigen**

- Ein Setup wird observiert
- Alle Änderungen werden protokolliert
- Daraus wird eine MSI-Datei erstellt

- **Probleme**

- Ein solches Setup kann nur eine einzige Konfiguration abbilden
- Es werden meist auch Änderungen protokolliert, die nicht auf das Setup zurückzuführen sind
- Wenn das System nicht im Originalzustand war, werden integrale Bestandteile des Setups nicht installiert (und protokolliert), da sie auf dem System schon vorhanden waren

3. Erstellen von MSI-basierten Setups

➤ Erstellen eines neuen Setups mit Wix

- **OpenSource-Werkzeug von Microsoft**
 - <http://sourceforge.net/projects/wix/>
 - **Common Public License**
- **Wird von Microsoft selbst verwendet**
- **Generiert aus einer XML-Datei MSI-Dateien etc.**

- **Vorgehen**
 - **Erstellen einer Wix-Skriptes**
 - **Compilation mit candle.exe**
 - **Linking mit light.exe**

4. Automatische Installation und Softwareverteilung

Johannes Passing



4. Automatische Installation und Softwareverteilung

➔ Unbeaufsichtigte Installation mit Transforms

- Während einer Installation werden die Benutzereingaben von einer Transform-Datei gelesen
 - Benutzereingaben wie Name, Installationsverzeichnis
 - Auswahl von Features
- Eine einmal erstellte Transform-Datei kann für viele Installation verwendet werden
- Start der Installation mit msiexec
- Wird zusätzlich die Benutzeroberfläche deaktiviert (UILEVEL) läuft das Setup vollständig im Hintergrund

4. Automatische Installation und Softwareverteilung

➔ Sicherheit

- **Installationen werden mit den Rechten des ausführenden Nutzers installiert**
- **Problem: Viele Setups benötigen administrative Rechte**
 - **Managed Applications können mit ‚elevated privileges‘ installiert werden**
 - **Insbesondere über GPO verteilte Software ist ‚managed‘**
 - **Über Richtlinien verwaltbar**
 - **Managed Applications gelten als vom Administrator genehmigt**

4. Automatische Installation und Softwareverteilung

➔ Softwareverteilung mittels Group Policy Objects

● Innerhalb einer Windows-Domäne kann Software über Einträge in GPO verteilt werden

- Automatische Installation
- Zentrale Verteilung von Updates
- Anpassung mittels Transforms
- Software kann
 - Pro Maschine verteilt werden
 - Pro Benutzer verteilt werden

- Softwarepakete werden zentral im Netzwerk abgelegt (Distribution Points, vorzugsweise DFS-Shares)
- Durch ACLs kann Reichweite der Verteilung eingeschränkt werden

✍ Reduzierter Administrationsaufwand

4. Automatische Installation und Softwareverteilung

➔ Installation pro Maschine

- **Während des Bootvorgangs eines Domänencomputers werden Applikationen installiert**
 - **Installation findet als SYSTEM statt**
 - Distribution Points müssen entsprechende ACL aufweisen
 - **Software steht allen Nutzern zur Verfügung**

- **Verteilung nur im Assigned- Modus möglich**
 - **Software wird vollständig installiert**

4. Automatische Installation und Softwareverteilung

➔ Installation pro Benutzer

- **Während des Anmeldevorgangs eines Domänencomputers werden Applikationen installiert**
 - **Installation findet normalerweise als anmeldender Benutzer statt**
 - Distribution Points müssen entsprechende ACL aufweisen
 - Meist Installation mit ‚elevated privileges‘ notwendig
 - **Software steht nur dem Nutzer zur Verfügung**
 - **Insbesondere im Zusammenspiel mit Roaming Profiles sinnvoll**

4. Automatische Installation und Softwareverteilung

➔ Installation pro Benutzer (Fortsetzung)

● Verteilung in 2 Modi möglich

■ Assigned

- Software wird vollständig installiert
- Wie bei maschinenweiter Verteilung

■ Published

- Eintrag in ‚Add/Remove Software‘
- Optional Install-on-demand

Registrierung von Extensions und Startmenü-Einträgen

■ Ggf. mit ‚elevated privileges‘

4. Automatische Installation und Softwareverteilung

➤ Upgrades und Deinstallation

- **Pakete können jederzeit ‚redeployed‘ werden**

- **Patches**

- Patches werden nicht separat verteilt
- Patch wird auf MSI-Paket angewendet
- Paket wird ‚redeployed‘

- **Deinstallation**

- **2 Modi verfügbar**

- **Optional Removal:** Paket wird fortan nicht mehr verteilt, bleibt aber installiert
- **Forced Removal:** Paket wird von allen Rechnern deinstalliert

Quellen

- **Microsoft: Windows Installer Overview**
 - <http://www.microsoft.com/windows2000/techinfo/howitworks/management/installer.asp> (14.04.2005)
- **Microsoft: Windows Installer: Benefits and Implementation for System Administrators**
 - <http://www.microsoft.com/windows2000/techinfo/administration/management/wininstaller.asp> (14.04.2005)
- **Microsoft: MSDN Library**
 - <http://msdn.microsoft.com/library/> (14.04.2005)
- **Microsoft: Windows 2000 Server Distributed Systems Guide**
 - Microsoft Press, 2000
- **Rob Mensching: When Setup isn't just Xcopy**
 - <http://blogs.msdn.com/robmen> (14.04.2005)