



# Windows Installer Xml - Overview

Jan Zieschang, 26.10.2007,  
Berlin





Aktuelle Probleme

Was ist Windows Installer Xml?

Vorteile von WIX

Nachteile von WIX

Entwicklungsansätze/Vorgehen

Aktuelle Erfahrungen

Diskussion

Sonstiges / Links



- Erstellung von Installationspaketen zu meist mit VS.NET
- Installationspakete haben sehr viele Einschränkungen und Limitierungen
- Kein Einheitliches Vorgehen
- „Erfahrunginseln“ vorhanden
- Erstellung von MSI-Paketen dauern lange, trotz Klick-Erstellung
- Wiederholung gleicher Fehler bei der Installation
- „Projektleiter mit Messer im Rücken“ (Reinhard Köhler)



- Ein Open Source Toolset zum Erstellen von Windows Installer Paketen
- Komplette Definition des Paketes basiert auf Xml
- Entwicklung erfolgt durch Microsoft Mitarbeiter, früher Microsoft selber
- Entwickelt mittels C#
- Alternativer Ansatz zur Erstellung von Windows Installer kompatiblen Paketen



- Menschen lesbar, da Xml
- Basierend auf Kommandozeilen Tools – Kein VS.NET notwendig
- Integration in eine Build-Umgebung (MsBuild-Target)
- Kompilierähnliches vorgehen (Pre-Compile und Compile | candle und light)
- Anpassung der Installation ohne WIX-Tools möglich, allerdings Build-Server notwendig
- Integration in die VS-IDE möglich (Votive)
- Separates Click-Tool vorhanden
- Verteiltes Erstellen/Bearbeiten einer Installation möglich
- Definition von Platzhaltern und Variablen
- Nicht .NET spezifische Lösung, nur Windows spezifisch



## Nachteile von WIX / Windows Installer / MSI

6

- Sehr komplexes Konstrukt und sehr mächtig
- Bestimmte Regeln müssen beachtet werden
- Teilweise schwierige Ausführung von „Sonderaufgaben“
- Einarbeitung in die Syntax, Regeln und „Actions“ Aufwändig
- Initiales Erstellen eines Paketes fehleranfällig
  - „Versauen“ des Testrechners möglich
  - Teilweise bemerken der Fehler, wenn es zu spät ist
- WIX IDE-Integration nicht immer 100%ig korrekt
- Patches (MSP) sehr schwierig zu erstellen

## Installer- Entwickler

Schnelleres Erstellen eines funktionierenden Paketes

Bessere Usability möglich

Effektivere Modularisierung

Administrative Installationspakete

Komplexere Installationsszenarien

## All-In-One- Entwickler

Initiales Erstellen mit Try-And-Error

Beschränkung auf Minimum (UI-lose Installation)

Einfache Anwendungsinstallationen



- **Sämtliche Installationspakete sollten mittels eines Build-Systems (Batch-Skripte, CCNET, TFS) zentral erstellt werden!**
- Nutzen von Platzhaltern und Parametern
- Jeder im Team sollte Grundsätze der Installation verstehen
- Genau Dokumentation der Anwendungsanforderungen und bei der Installation Variablen Parameter → WIKI
- Installationstests durch MSI-Entwickler, Entwickler, Anwender/DAU (PL)
- Beim Erstellen der Anwendung bereits auf Script-/Batch-Fähigkeit achten
- Keine Hard-Coded-Parameter





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

```
  <Product Id="$(var.productID)"
```

```
    Name="$(var.app)" Version="$(var.version)"
```

```
    Manufacturer="CapeVision GmbH"
```

```
    UpgradeCode="$(var.upgradeGUID)" Language="1031" Codepage="1252">
```

```
  <Package InstallerVersion="200" Compressed="yes" ShortNames="yes"/>
```

```
  <Media Id="1" Cabinet="AgentServiceInstallation.cab" EmbedCab="yes" CompressionLevel="medium"  
  VolumeLabel="$(var.app)$(var.version)"/>
```

```
  <Upgrade Id="$(var.upgradeGUID)">
```

```
    <UpgradeVersion OnlyDetect="no" Property="VERSIONFOUND" Maximum="$(var.version)" IncludeMaximum="yes" />
```

```
    <UpgradeVersion OnlyDetect="yes" Property="NEWERFOUND" Minimum="$(var.version)" IncludeMinimum="no" />
```

```
  </Upgrade>
```

```
  <CustomAction Id="NoDowngrade" Error="!(loc.condProductAlreadyExists)" />
```



```
<Condition Message="!(loc.condFrameworkNotFound)">FRAMEWORK20 = "#1" OR Installed</Condition>

<!--<Condition Message="!(loc.condPasswordMissing)">SERVICEPASSWORD OR Installed</Condition>-->

<Directory Id="TARGETDIR" Name="SourceDir">

  <Directory Id="ProgramFilesFolder">

    <Directory Id="CVINSTALLLOCATION" Name="CapeVision">

      <Component Id="c_ASBasisC" Guid="f8590118-6f14-4372-97ff-08993fbc1009" DiskId="1">

        <CreateFolder>

          <Permission GenericRead="yes" Traverse="yes" User="[USERS]"/>

          <Permission GenericAll="yes" Traverse="yes" User="[SERVICEUSER]" Domain="[SERVICEDOMAIN]"/>

          <Permission GenericAll="yes" Traverse="yes" User="[ADMINS]"/>

        </CreateFolder>

        <utils:User Id="serviceUser" Name="[SERVICEUSER]" Password="[SERVICEPASSWORD]"
Domain="[SERVICEDOMAIN]" PasswordNeverExpires="yes" FailIfExists="no" CreateUser="no" PasswordExpired="no"
RemoveOnUninstall="no" CanNotChangePassword="yes">

      </utils:User>

    </Directory>

  </Directory>

</Directory>
```



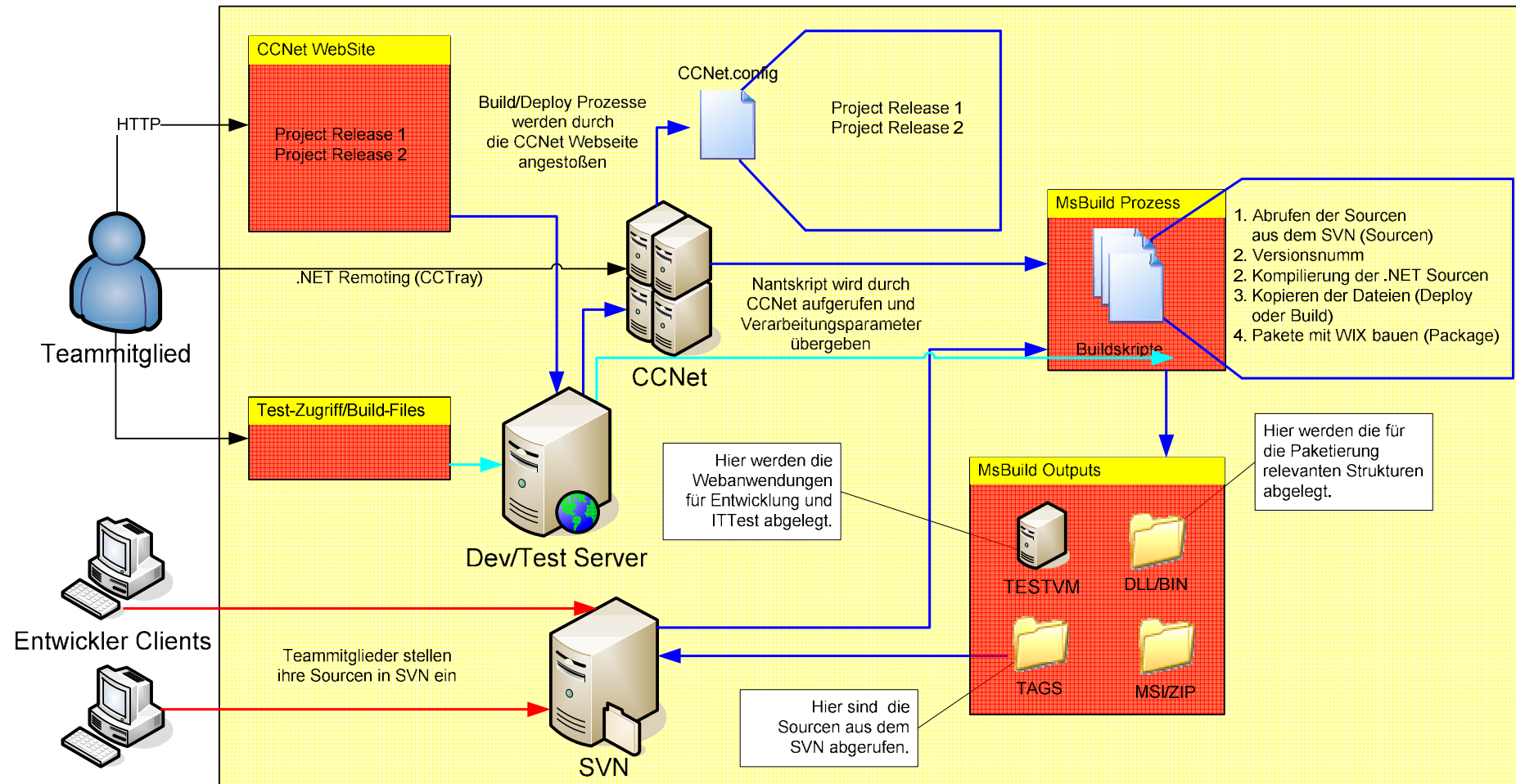
```
<Dialog Id="AdminWelcomeDlg" Width="370" Height="270" Title="[ProductName] [Setup]" NoMinimize="yes">
  <Control Id="Next" Type="PushButton" X="236" Y="243" Width="56" Height="17" Default="yes"
  Text="[ButtonText_Next]">
    <Publish Property="InstallMode" Value="Server Image"><![CDATA[1]]></Publish>
    <Publish Event="NewDialog" Value="AdminInstallPointDlg"><![CDATA[1]]></Publish>
  </Control>
  <Control Id="Cancel" Type="PushButton" X="304" Y="243" Width="56" Height="17" Cancel="yes"
  Text="[ButtonText_Cancel]">
    <Publish Event="SpawnDialog" Value="CancelDlg"><![CDATA[1]]></Publish>
  </Control>
  <Control Id="Bitmap" Type="Bitmap" X="0" Y="0" Width="370" Height="234" TabSkip="no" Text="[DialogBitmap]" />
  <Control Id="Back" Type="PushButton" X="180" Y="243" Width="56" Height="17" Disabled="yes"
  Text="[ButtonText_Back]" />
  <Control Id="BottomLine" Type="Line" X="0" Y="234" Width="374" Height="0" />
  <Control Id="Description" Type="Text" X="135" Y="70" Width="220" Height="30" Transparent="yes" NoPrefix="yes">
    <Text>!(loc.SetupWelcomeDescription)</Text>
  </Control>
  <Control Id="Title" Type="Text" X="135" Y="20" Width="220" Height="60" Transparent="yes" NoPrefix="yes">
    <Text>{\VerdanaBold13}!(loc.SetupWelcomeTitle)</Text>
  </Control>
</Dialog>
```



- Installationspakete bei der DPAG (Martin Brinkmann, Jan Zieschang)
  - Umfangreiches Deployment einer HV-Server-Applikation
  - Verteilung und Parametrisierung der Applikation
  - Ohne UI
  - Administrativ
  
- Installation Agent Service (Jan Zieschang)
  - UI basierende Anwendungsinstallation
  - Installation eines Services
  - Konfigurieren der Anwendung

**Buildmanagement – Zukunft?** 13

### Buildmanagement





- Rop mensching blog - <http://blogs.msdn.com/robmen>
- <http://wix.sf.net>
- Tutorial <http://www.tramontana.co.hu/wix/>
- Orca – Tool zum Prüfen von Installationspaketen, aber auch manipulieren
  
- Paketierungsregeln?
- Offen: Smart-Client -> VS-Deploment
- Build-Integration/-Automatisierung



|                      |   |
|----------------------|---|
| WIX                  | Windows Installer Xml                             |
| MSP                  | Microsoft Patch File                              |
| Administrative Inst. | Sparen von Systemressourcen                       |
| UI                   | User Interface                                    |
| IDE                  | Integrated Development Environment                |
| HV                   | Hochverfügbar                                     |
| Wiki                 | Userbasierender-Online-Wissensspeicher            |
| CCNET                | Cruise Control .NET – Continues Intragration Tool |
| TFS                  | Team Foundation Server                            |