

**Herzlich Willkommen**

Paul Rohorzka



IT WORKS – WE CARE

**TECHTALK**



# Deployment von .NET Applikationen

Grundlagen – Vorbereitung – Durchführung

# Deployment Inhalt

## Grundlagen und Vorbereitung

- Assemblies
- Konfiguration
- Logging

## Durchführung

- XCOPY
- Setups
- ClickOnce

## Goodie

- Datenbanken

# Assemblies

## Grundlagen und Vorbereitung

- Assemblies
- Konfiguration
- Logging

## Durchführung

- XCOPY
- Setups
- ClickOnce

## Goodie

- Datenbanken

# Assemblies

- Was ist ein Assembly?
- Welcher Code wohin?
- Ausgeholt: Namespaces
- Benennungsregeln

## Assemblies

# Was ist ein Assembly?

- Physischer Bestandteil einer .NET-Applikation
- Beinhaltet Code, Ressourcen und Metainformation (Manifest)
- Können im GAC installiert sein
  - Verschiedene Versionen parallel
  - Starker Name erforderlich (sn.exe)
  - System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
- Dll oder Exe

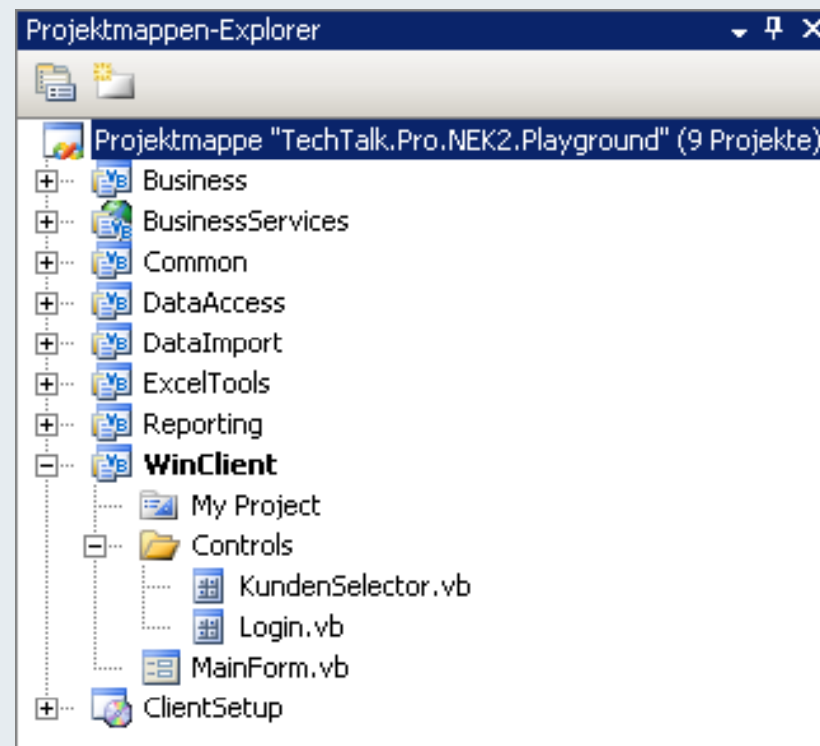
Global  
Assembly  
Cache

%windir%\assembly

## Assemblies

# Welcher Code wohin?

- Ein Assembly  $\equiv$  ein Projekt in Visual Studio
- Logisch zusammengehörender Code in ein Assembly
- Im UI Projekt nur Formulare und Controls!



## Assemblies

# Ausgeholt: Namespaces

- Hierarchische Unterteilung der Menge aller Typen
- Auflösung von Namenskonflikten bei Typen mit gleichem Namen
- Beispiel: RichTextBox:
  - System.Windows.Forms.**RichTextBox**
  - Microsoft.Office.Tools.Excel.Controls.**RichTextBox**
- Konvention:
  - Organisation.Projekt.Applikation.Bereich.Xxx usw.
  - Mehrdeutigkeiten sind global sehr unwahrscheinlich
  - Beispiel: AUVA. EFEU. DomainModel. Leistung. Sachleistung. Abrechnung. Filter

*"... prevent polluting the current scope ..."*

## Assemblies

# Benennungsregeln

- Projektname (Visual Studio und Sourcen)
- Stamm-Namespace (Platzierung der Typen)
- ~~Assemblynamen~~ gemeinsamen Namespace – ALVA.FFFU.DomainModel  
Assemblynamen (Name der erzeugten Datei)
  - oder kleinsten gemeinsamen Nenner – DomainModel
- Stamm-Namespace (Platzierung der Typen)
  - Benennung nach Konvention – Microsoft.Office.InfoPath
- Assemblynamen (Name der erzeugten Datei)
  - Benennung nach
    - Dlls: gemeinsamer Namespace – System.Data.dll
    - Exe: Projektname – Winword.exe



IT WORKS – WE CARE

**TECHTALK**

# Konfiguration

## Grundlagen und Vorbereitung

- Assemblies
- Konfiguration
- Logging

## Durchführung

- XCOPY
- Setups
- ClickOnce

## Goodie

- Datenbanken

# Konfiguration

- Konfigurationsdateien
- appSettings und connectionStrings
- My.Settings
- weiter ausgebaut

# Konfiguration

# Konfigurationsdateien in .NET

## ■ Hierarchische Konfigurationsdaten in XML-Dateien

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="userSettings" type="System.Configuration.UserSettingsGroup, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c5619" />
    <section name="TechTalk.Pro.NEK2.Playground.winClient.My.MySettings" type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c5619" />
    </sectionGroup>
    <sectionGroup name="applicationSettings" type="System.Configuration.ApplicationSettingsGroup, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c5619" />
    <section name="TechTalk.Pro.NEK2.Playground.winClient.My.MySettings" type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c5619" />
    </sectionGroup>
  </configSections>
  <appSettings>
    <add key="mandant" value="Man01"/>
    <add key="defaultview" value="Coolview"/>
  </appSettings>
  <connectionStrings>
    <add name="configDatabase" connectionString="Data Source=TTV-PROD\sql2008;Initial Catalog=ConfigData;User Id=ConfigReader;Pwd=brumbrumm"/>
    <add name="theData" connectionString="Server=TTV-PROD\sql2008;Initial Catalog=AppData;User Id=AppRw;Pwd=!v1e2r3y4s5e6c7r8e9t!"/>
  </connectionStrings>
  <system.diagnostics>
    <sources>
      <!-- This section describes the logging sources -->
      <source name="DefaultSource" switchName="DefaultSwitch">
        <listeners>
          <add name="FileLog"/>
          <!-- Uncomment the below section -->
          <!--<add name="EventLog"/>-->
        </listeners>
      </source>
    </sources>
    <switches>
      <add name="DefaultSwitch" value="Information" />
    </switches>
    <sharedListeners>
      <add name="FileLogger"
          type="Microsoft.VisualBasic.Logging.FileLogTraceListener, Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
          initializeData="FileLogWriter"
          baseFileName="NEK2Playground"
          location="Custom"
          customLocation="c:\log\"
          delimiter=";"
          autoFlush="true"
          append="true"/>
      <!--<add name="EventLog" type="System.Diagnostics.EventLogTraceListener" initializeData="NEK2Playground"/>-->
    </sharedListeners>
  </system.diagnostics>
  <system.serviceModel>
    <bindings>
      <wsHttpBinding>
        <binding name="WSHttpBinding_IKundenService" closeTimeout="00:01:00"
            openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="00:01:00"
            bypassProxyOnLocal="false" transactionFlow="false" hostNameComparisonMode="StrongWildcard" />
      </wsHttpBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

# Konfiguration

## appSettings & connectionStrings

- Einfache Konfigurationsmöglichkeit

```
<appSettings>
  <add key="mandant" value="Man01"/>
  <add key="defaultview" value="coolview"/>
</appSettings>

<connectionStrings>
  <add name="configDatabase" connectionString="Server=(local
  <add name="theData" connectionString="Server=sql2008\apps
</connectionStrings>
```

- Lesen über ConfigurationManager

```
Dim mandant As String = ConfigurationManager _
    .AppSettings("mandant")

Dim dataConnectionString As String = ConfigurationManager _
    .ConnectionStrings("theData") _
    .ConnectionString
```

- Untypisiert

# Konfiguration

## Visual Basic: My.Settings

- Unterstützung durch Visual Studio

### Windows XP:

%userprofile%\Lokale Einstellungen\  
Anwendungsdaten\<<Company>

### Windows Vista/7:

%userprofile%\AppData\  
Local\<<Company>

- Eigene configSections in der \*.config

```
<userSettings>  
  <TechTalk.Pro.NEK2.Playground.winClient.My.MySettings>  
    <setting name="standardfarbe" serializeAs="String">  
      <value>255, 153, 153</value>  
    </setting>  
  </TechTalk.Pro.NEK2.Playground.winClient.My.MySettings>  
</userSettings>  
  
<applicationSettings>  
  <TechTalk.Pro.NEK2.Playground.winClient.My.MySettings>  
    <setting name="Anwendungstitel" serializeAs="String">  
      <value>Meine Anwendung</value>  
    </setting>  
  </TechTalk.Pro.NEK2.Playground.winClient.My.MySettings>  
</applicationSettings>
```

userSettings:  
lesen und schreiben

applicationSettings:  
nur lesen

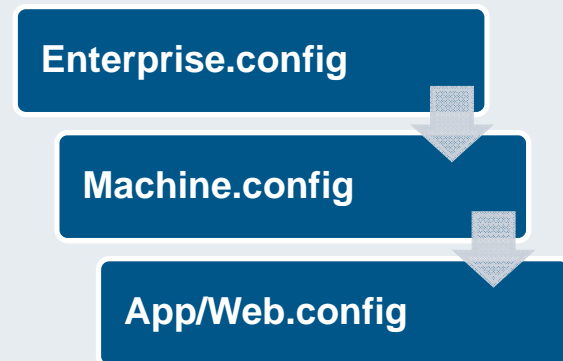
- Typisierter Zugriff  
auf Einstellungen

```
Me.BackColor = My.Settings.standardfarbe  
Me.Text = My.Settings.Anwendungstitel
```

## Konfiguration

# Konfiguration weiter ausgebaut

- Hierarchische Struktur
  - Einstellungen werden vererbt
  - können ev. überschrieben werden



- Verschlüsselung einer Web.config
  - Einzelne Bereiche können vom Administrator verschlüsselt werden (ConnectionStrings!)
    - `aspnet_regiis -pef mySecretSection`
  - Transparent für Entwickler
    - Nur auf dieser Maschine (MachineKey)

# Logging

## Grundlagen und Vorbereitung

- Assemblies
- Konfiguration
- Logging

## Durchführung

- XCOPY
- Setups
- ClickOnce

## Goodie

- Datenbanken

# Logging

- Warum loggen?
- Loggen mit System.Diagnostics
- Microsoft.VisualBasic.Logging
- Externe Komponente: log4net

## Logging

# Warum loggen?

Profiling

### Technik

- Leistungsparameter (Speicher, Platten)
- Häufig aufgerufene Funktionen (Caching?)

### Usability

- Was wird selten aufgerufen?
- Wo hat der Benutzer herumgesucht?

### Fehlersuche

- Wie war der Programmfluss?
- Wo genau ist ein Fehler/Problem aufgetreten?

### Revisionserfordernisse

- Wer hat welchen Akt geöffnet?

### Angriffserkennung

- Nicht erfolgreiche Loginversuche

Auditing

## Logging

# Loggen mit System.Diagnostics

- Klassen Debug und Trace:
  - Write, WriteLine
  - WriteLine, WriteLine
- Verschiedene Daten können mitgeloggt werden
  - (Timestamp, ProcessId, Callstack, ...)
- Listeners protokollieren die Einträge
- Einige Standard-Listener:
  - DefaultTraceListener
    - loggt in den Debugger, Visual Studio Ausgabefenster!
  - TextWriterTraceListener
  - EventLogTraceListener

## Logging

# Microsoft.VisualBasic.Logging

- Namespace hinter My. Appl i cati on. Log
- WriteEntry, WriteException
- Bietet FileLogTraceListener
  - LogFileCreationSchedule – automatisch neues Logfile
    - Daily
    - Weekly
  - MaxFileSize, ReservedDiskSpace – Größenbeschränkung

## Logging

# Konfiguration des Logging

- Abschnitt `<system.diagnostics>`

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  ...
  <system.diagnostics>
    <sources>
      <source name="DefaultSource" switchName="DefaultSwitch">
        <listeners>
          <add name="FileLog"/>
        </listeners>
      </source>
    </sources>
    <switches>
      <add name="DefaultSwitch" value="Information" />
    </switches>
    <sharedListeners>
      <add name="FileLog"
          type="Microsoft.VisualBasic.Logging.FileLogTraceListener, ..."
          initializeData="FileLogWriter"/>
    </sharedListeners>
  </system.diagnostics>
  ...
</configuration>
```

Wer spricht?

Wer hört?

Wer kann hören?

## Logging

# Externe Komponente: log4net

- Open Source – <http://logging.apache.org/log4net>
- Logger (~Source)
  - auch hierarchisch
- Appender (~Listener)
  - File
  - EventLog
  - Datenbank
  - E-Mail
- Vielfältige Konfigurationsmöglichkeiten
  - werden on-the-fly übernommen



IT WORKS – WE CARE

**TECHTALK**



# XCOPY- Deployment

Grundlagen  
und  
Vorbereitung

- Assemblies
- Konfiguration
- Logging

Durchführung

- XCOPY
- Setups
- ClickOnce

Goodie

- Datenbanken

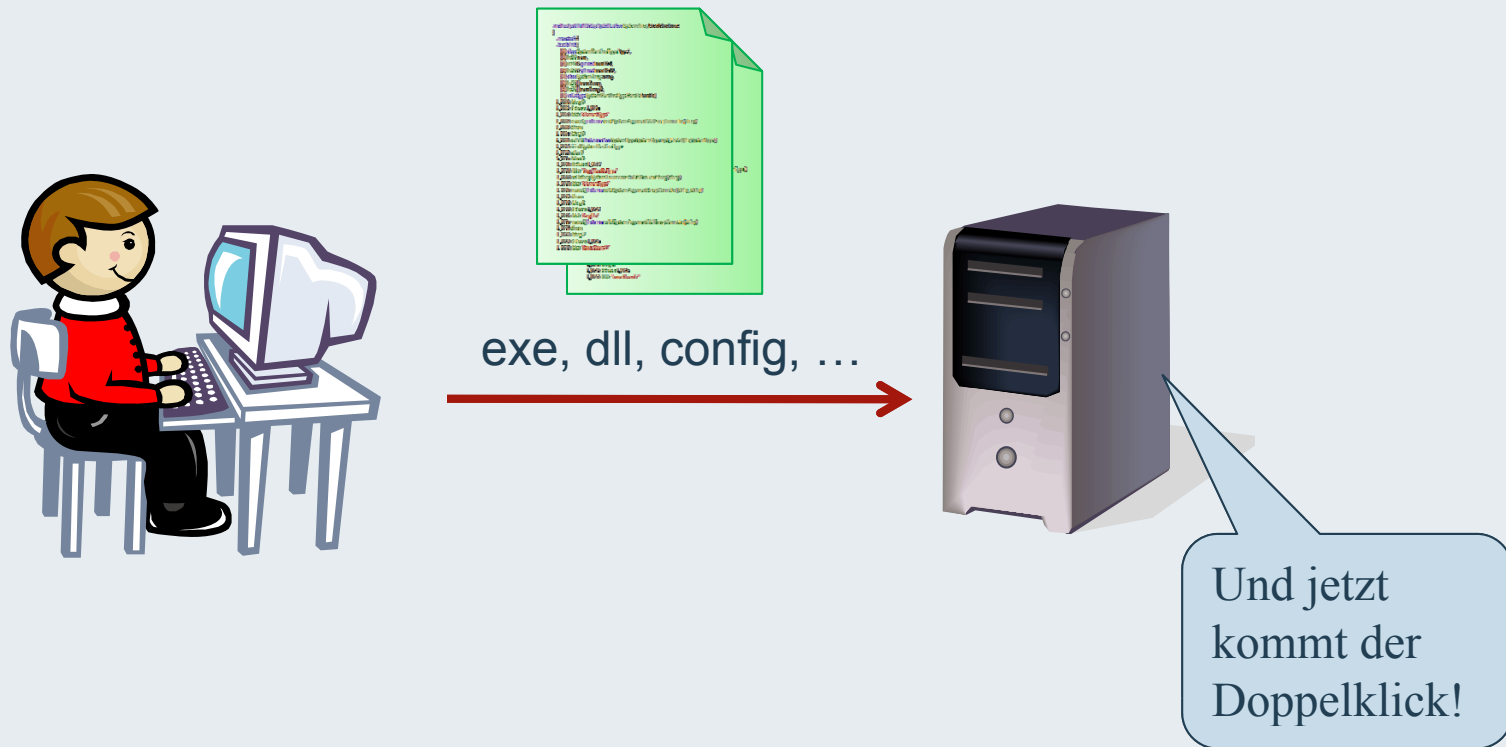
# XCOPY-Deployment

- Was ist das?
- Reicht das tatsächlich?

## XCOPY-Deployment

# Was ist XCOPY-Deployment?

- Verteilen einer Anwendung per Dateikopie  
Explorer, FTP, XCOPY, RoboCopy, ...



## XCOPY-Deployment

# Reicht das tatsächlich?

... also nicht  
sehr oft! ☹

- XCOPY-Deployment reicht, wenn
  - Anwendung ausschließlich auf .NET basiert
    - kein COM/ActiveX (Ausnahme: reg-free COM)
  - keine Komponente im GAC registriert sein muss
  - nichts in die Registry geschrieben werden muss
  - keine Shell-Integration erforderlich ist
    - Eintrag in Startmenü
    - Assoziation von Dateitypen
  - die Quelle vertrauenswürdig ist (Security)



IT WORKS – WE CARE

**TECHTALK**



# Setups mit VisualStudio

Grundlagen  
und  
Vorbereitung

- Assemblies
- Konfiguration
- Logging

Durchführung

- XCOPY
- Setups
- ClickOnce

Goodie

- Datenbanken

# Setups mit VisualStudio

- Projektvorlage Setup-Projekt
- Erstellen eines einfachen Setups
- Vorgehen bei Upgrade-Paketen

## Setups mit VisualStudio

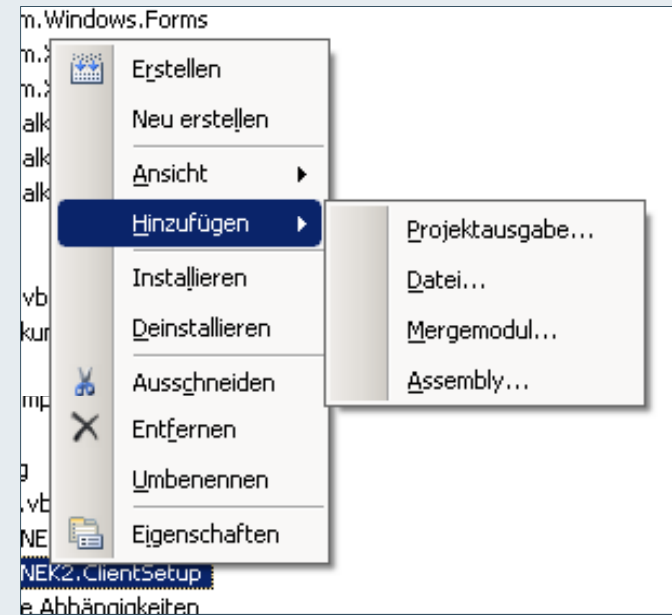
# Projektvorlage Setup-Projekt

- Erstellen einfacher Setup-Pakete
  
- Unterstützt (mit eigenen Ansichten):
  - Hinzufügen von Elementen im Dateisystem
  - Hinzufügen von Einträgen in die Registry
  - Registrierung von Dateitypen
  - Parametrisierung einer vorgegebenen Benutzeroberfläche
  - Benutzerdefinierte Aktionen
  - Kontrolle von Startbedingungen

# Setups mit VisualStudio – Walkthrough

## 1. Was soll verteilt werden?

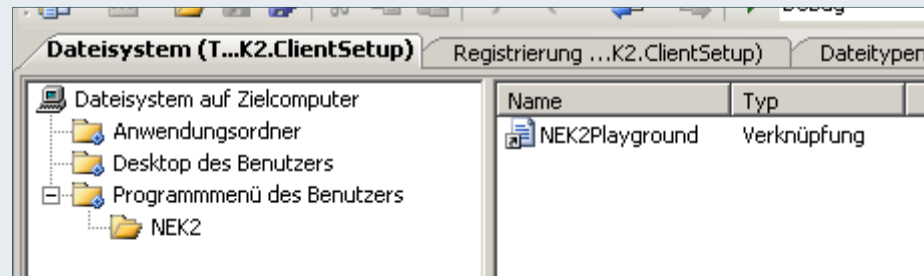
- Hinzufügen
  - Projektausgabe – meist der Client
  - Dateien – Hilfe, Readme, ...
  - Mergemodul – anderes "Teil-Setup"
  - Assembly – falls nicht automatisch über Projektverweise mitgenommen



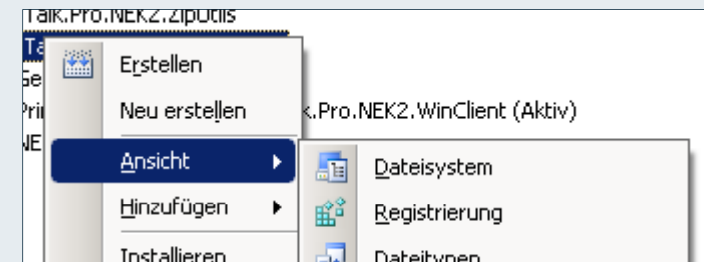
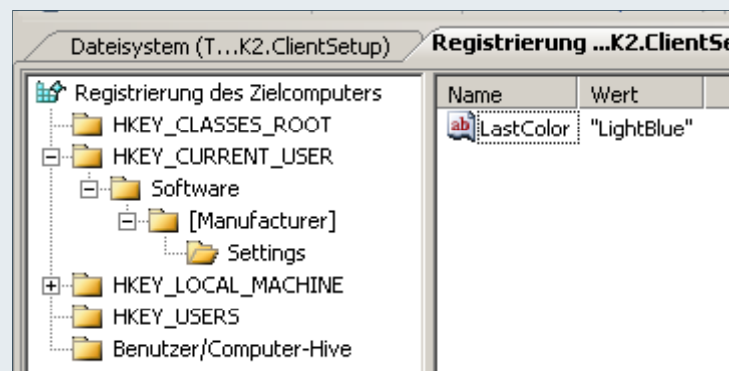
# Setups mit VisualStudio – Walkthrough

## 2. Was soll am Ziel geschehen?

- Ansicht
  - Dateisystem



- Registry



# Setups mit VisualStudio – Walkthrough

## 3. Anpassung der Dialoge

- Ansicht
  - Benutzeroberfläche

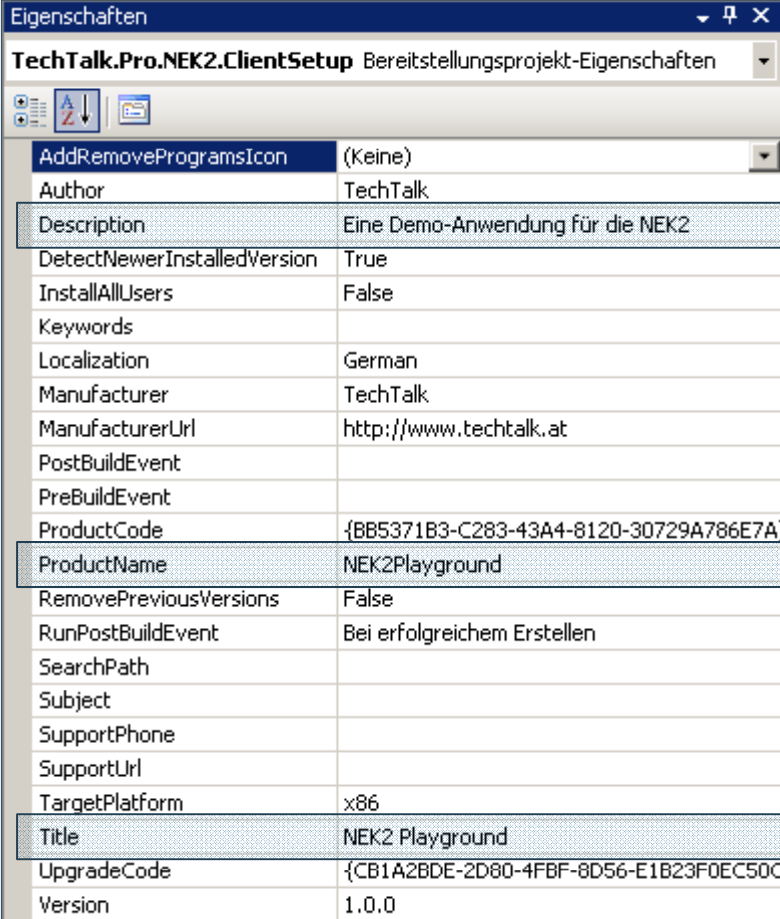
The screenshot shows the Visual Studio IDE with the 'Benutzeroberfl...2.ClientSetup' project selected. The project tree on the left shows the 'Installs' folder expanded, with 'Willkommen' selected. The Properties window on the right shows the 'Willkommen' dialog properties, including 'BannerBitmap', 'CopyrightWarning', and 'WelcomeText'. A callout box points to the 'Willkommen' dialog in the project tree, stating 'Anpassungen über die Eigenschaften'. A context menu is also visible, showing the 'Ansicht' (View) option selected, with 'Benutzeroberfläche' (User Interface) highlighted.

Willkommen Dialogfeldeigenschaften für die Benutzeroberfläche	
BannerBitmap	NEK.bmp
CopyrightWarning	Warnung: Dieses Programm ist durch österreichische Urheberrechtsgesetze ur
WelcomeText	Der Installer wird Sie durch die zur Installation von [ProductName] erforderliche

# Setups mit VisualStudio – Walkthrough

## 4. Anpassung der Eigenschaften

- ProductName
- Title
- Description
- ...



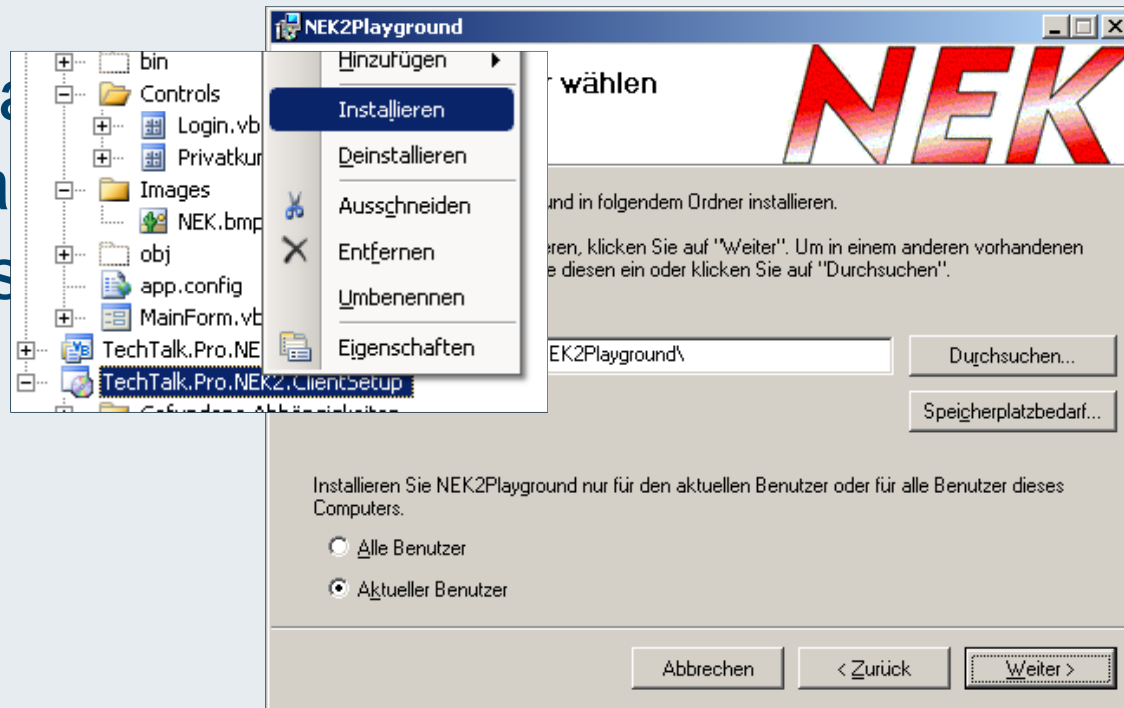
The screenshot shows the 'Eigenschaften' (Properties) window for a 'TechTalk.Pro.NEK2.ClientSetup' project. The window displays a list of properties for the 'Bereitstellungsprojekt-Eigenschaften' (Setup Project Properties). The 'ProductName' and 'Title' properties are highlighted with a blue selection bar.

Property	Value
AddRemoveProgramsIcon	(Keine)
Author	TechTalk
Description	Eine Demo-Anwendung für die NEK2
DetectNewerInstalledVersion	True
InstallAllUsers	False
Keywords	
Localization	German
Manufacturer	TechTalk
ManufacturerUrl	http://www.techtalk.at
PostBuildEvent	
PreBuildEvent	
ProductCode	{BB5371B3-C283-43A4-8120-30729A786E7A}
ProductName	NEK2Playground
RemovePreviousVersions	False
RunPostBuildEvent	Bei erfolgreichem Erstellen
SearchPath	
Subject	
SupportPhone	
SupportUrl	
TargetPlatform	x86
Title	NEK2 Playground
UpgradeCode	{CB1A2BDE-2D80-4FBF-8D56-E1B23F0EC50C}
Version	1.0.0

# Setups mit VisualStudio – Walkthrough

## 5. Test-Installation

- Am Entwicklungsrechner direkt aus VisualStudio
- Oder das Paket auf einem anderen Rechner testen



## Setups mit VisualStudio

# Vorgehen bei Upgrade-Paketen

- Setup für eine Folgeversion:
  - DetectNewerInstalledVersion = True
  - RemovePreviousVersions = True
  - Version <erhöhen>
  - ProductCode <verändern>
  - UpgradeCode <nicht verändern>

Andernfalls wird die neue Version parallel oder gar nicht installiert!



IT WORKS – WE CARE

**TECHTALK**



# ClickOnce Deployment

Grundlagen  
und  
Vorbereitung

- Assemblies
- Konfiguration
- Logging

Durchführung

- XCOPY
- Setups
- ClickOnce

Goodie

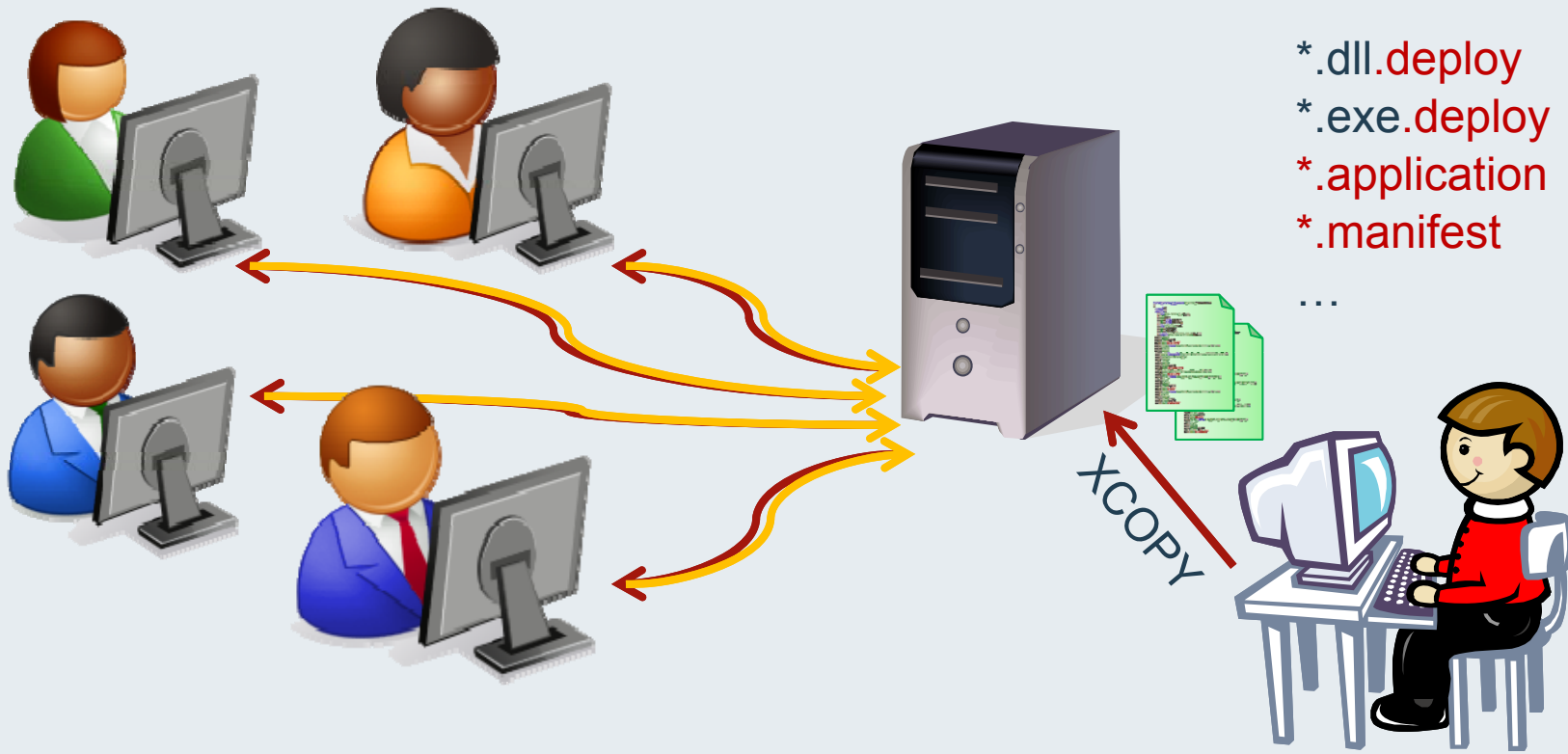
- Datenbanken

# ClickOnce Deployment

- Überblick und Features
- Vorbereitung Erstdeployment
- Bereitstellungs-Website
- Veröffentlichte Dateien

# ClickOnce Deployment Überblick

- "Frontend-Updater" des .NET-Frameworks



## ClickOnce Deployment

# Features

- Keine lokalen Admin-Rechte für Deployment erforderlich
- Admin-Sicht:
  - Voraussetzung: HTTP-Server (muss nicht IIS sein)
  - Neue Version per XCOPY auf den Server
- Benutzer-Sicht:
  - Voraussetzung: .NET 2.0
  - ev. lokale Installation (Offline-Verwendung)
  - Automatische Updates
- Je nach Funktionalität unterschiedliche Berechtigungen (CAS) erforderlich
  - Zugriff auf Dateisystem? → FullTrust!

## ClickOnce Deployment

# Vorbereitung Erstdeployment

- Veröffentlichung via Projekteigenschaften oder Assistenten

Nur online oder auch offline verwendbar?

The screenshot shows the 'Veröffentlichen' (Publish) tab of the application properties dialog. The 'Anwendung' (Application) section is selected in the left sidebar. The 'Ort der Veröffentlichung' (Location of publication) section includes a dropdown for 'Speicherort des Veröffentlichungsordners' (Storage location of the publication folder) set to 'http://localhost/NEK2Playground/'. The 'Installationsmodus und -einstellungen' (Installation mode and settings) section has two radio buttons: 'Anwendung ist nur online verfügbar' (Application is only available online) and 'Die Anwendung ist auch offline verfügbar (zu starten über das Startmenü)' (The application is also available offline (to start via the Start menu)). The second option is selected. The 'Veröffentlichungsversion' (Publication version) section has four input fields: 'Hauptversion' (1), 'Nebenversion' (0), 'Build' (0), and 'Revision' (3). A checkbox 'Revisionsnummer automatisch mit jeder Veröffentlichung erhöhen' is checked. Buttons for 'Anwendungsdateien...', 'Erforderliche Komponenten...', 'Updates...', and 'Optionen...' are on the right. At the bottom are 'Webpublishing-Assistent...' and 'Jetzt veröffentlichen' buttons.

## ClickOnce Deployment

# Vorbereitung Erstdeployment cont'd

- Updates: automatisch nach Updates suchen

Vor jedem Start  
nach einem  
Update suchen

**Anwendungsupdates**

Die Anwendung soll nach Updates suchen

Zeitpunkt auswählen, wann die Anwendung auf Aktualisierungen überprüfen soll:

Nach dem Starten der Anwendung  
Wählen Sie diese Option aus, um die Anwendungsstartzeit zu verkürzen. Updates werden erst installiert, wenn die Anwendung das nächste Mal ausgeführt wird.

Vor Start der Anwendung  
Wählen Sie diese Option aus, um sicherzustellen, dass mit dem Netzwerk verbundene Benutzer immer das neueste Update ausführen.

Häufigkeit der Überprüfung auf Aktualisierungen angeben:

Bei jedem Ausführen der Anwendung überprüfen

Überprüfung alle: 7 Tag(e)

Geben Sie die mindestens erforderliche Version für diese Anwendung an.

Hauptversion: Nebenversion: Build: Revision:

Aktualisierungspfad (wenn anders als Veröffentlichungsort):  
http://localhost/NEK2Playground/

## ClickOnce Deployment

# Vorbereitung Erstdeployment cont'd

- Optionen: u.a. Bereitstellungswebsite

Diese HTML-Seite wird automatisch generiert

Veröffentlichungsoptionen

Sprache für Veröffentlichung:  
(Standard)

Herausgebername:  
TechTalk Software Support Handelsgesellschaft m.b.H.

Produktname:  
NEK2 Playground

Support-URL:  
Durchsuchen...

Bereitstellungswebsite:  
Deployment.html

Nach jeder Veröffentlichung automatisch Bereitstellungswebseite generieren

Nach dem Veröffentlichlichen Bereitstellungswebseite öffnen

Verhindern, dass Anwendungen über eine URL aktiviert werden

Dateierweiterung ".deploy" verwenden

Übergeben von URL-Parametern an die Anwendung zulassen

Bei CD-Installationen automatisch Setup starten, wenn CD eingelegt wird

Auf Webserver hochgeladene Dateien überprüfen

Anwendungsmanifest für Vertrauensinformationen verwenden

OK Abbrechen

# ClickOnce Deployment

## Bereitstellungs-Website

- Kann automatisch erzeugt werden:



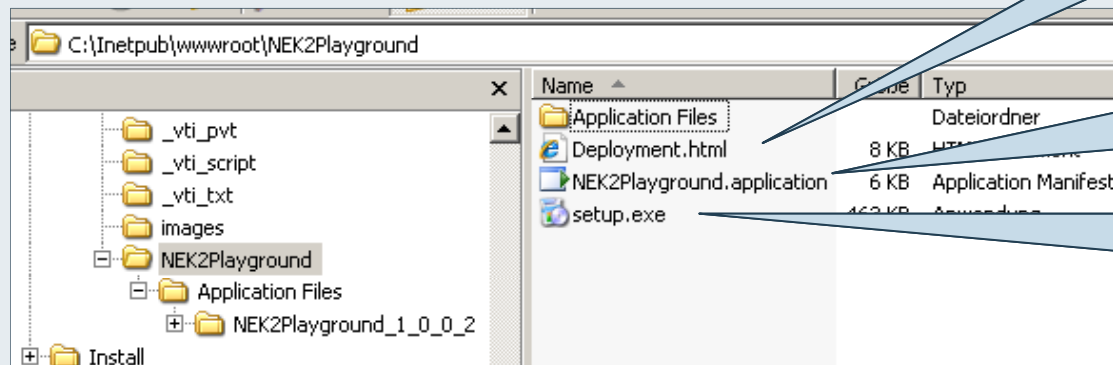
Anwendung  
online starten

Anwendung  
installieren

# ClickOnce Deployment

## Veröffentlichte Dateien

- Wurzel

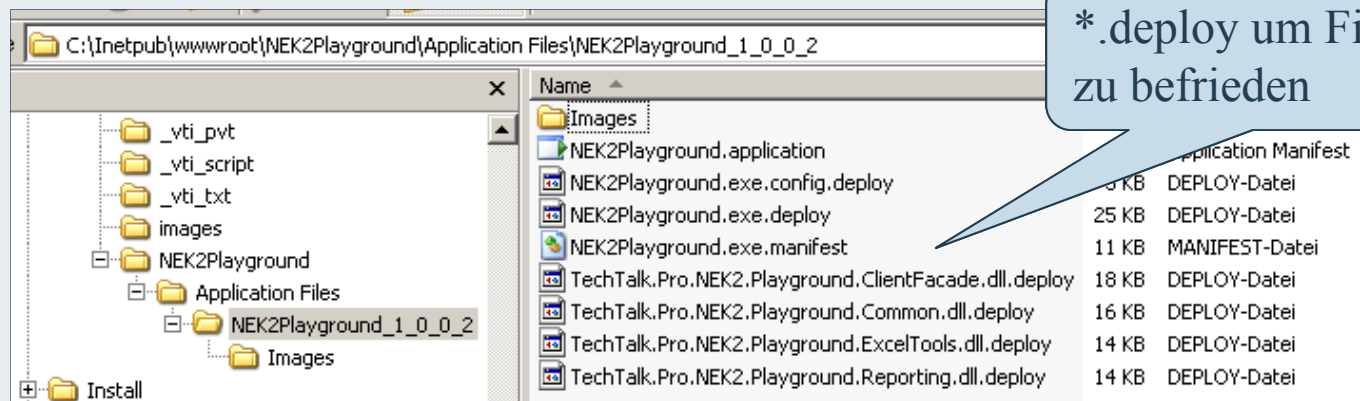


Bereitstellungs-  
website

\*.application für  
online-Ausführung

setup.exe für  
lokale Installation

- Verzeichnis für jede Version



\*.deploy um Firewalls  
zu befrieden



IT WORKS – WE CARE

**TECHTALK**



# Datenbanken upgraden

Grundlagen  
und  
Vorbereitung

- Assemblies
- Konfiguration
- Logging

Durchführung

- XCOPY
- ClickOnce
- Datenbanken

Goodie

- Datenbanken

# Datenbanken upgraden

- Backends und Deployment
- Datenbankupgrades bei ClickOnce
- Datenbankupgrades vorbereiten

# Datenbanken upgraden

## Backends und Deployment

- Jet
  - ev. MDAC erforderlich
  - Als Prerequisite bei ClickOnce
  - MDBs einfach kopieren
- SqlExpress
  - Schwierig bei ClickOnce
  - MDFs attachen?
- SQL Server Compact
  - Läuft in Process
  - Keine StoredProcedures, ...
  - SDFs einfach kopieren
  - Als Prerequisite bei ClickOnce
- Alternativen
  - SQLite (open source), ...

## Datenbanken upgraden


# Datenbankupgrades bei ClickOnce

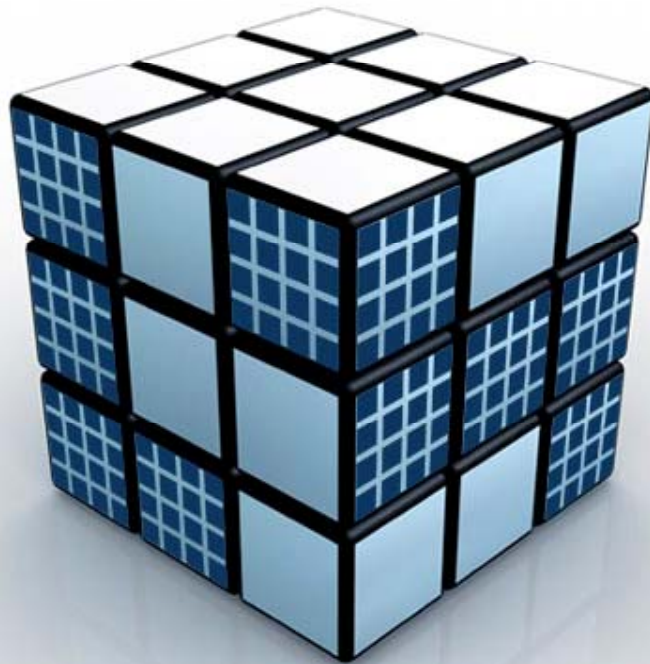
- Datenbankfiles werden beim Update überschrieben
- CurrentDeployment.IsFirstRun:  
Wird diese Version das erste Mal ausgeführt?
  - Dann gibt es die einmalige Chance, das bestehende Datenfile ins neue Verzeichnis zu migrieren:

```
If ApplicationDeployment.IsNetworkDeployed Then
  If ApplicationDeployment.CurrentDeployment.IsFirstRun Then
    Dim oldDatabase = New Database("|DataDirectory|.pre\Database.sdf")
    Dim newDatabase = New Database("|DataDirectory|\Database.sdf")
    ' Hier migrieren
  End If
End If
```

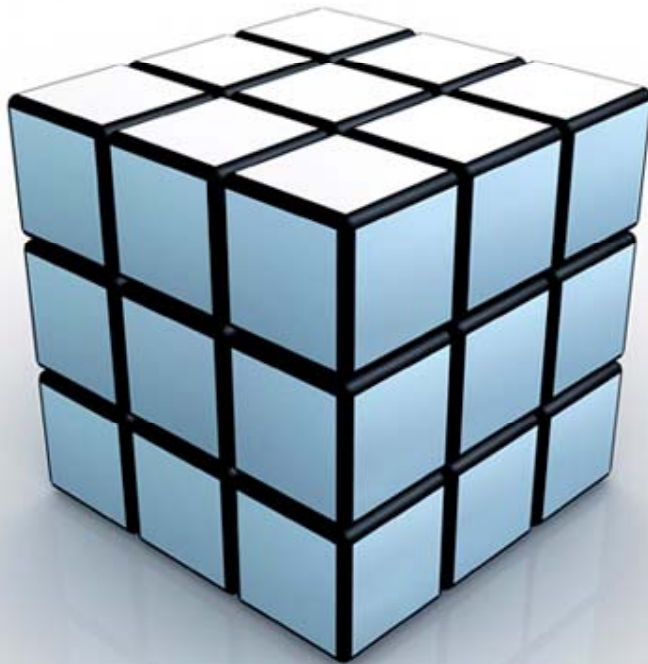
## Datenbanken upgraden

# Datenbankupgrades vorbereiten

- Upgrade der Datenstruktur per Skripts
  - ins Projekt aufnehmen 
  - händisch führen oder per Schema Compare erzeugen
  - oder per Code (API?)
- Migration vorhandener Daten
  - Per Skripts
    - Fehlerbehandlung?
  - Per Code
    - Rückgriff auf BusinessLogik



**Fragen?**



**Vielen Dank!**

Paul Rohorzka

pro@techtalk.at