

Wie weit ist .NET 2.0?

Neuerungen – ADO.NET – ASP.NET

Aufbau

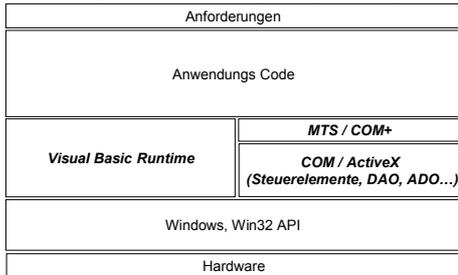
- ◆ I. Grundlegendes
Das .NET-Framework
- ◆ II. NewSpeak
Spracherweiterung in VisualBasic 2005
- ◆ III. Ran an die Daten!
ADO.NET
- ◆ IV. Mit den Daten ins Netz
ASP.NET

I. Grundlegendes **Das .NET-Framework**

Aufgaben
Bestandteile
Common Language Runtime

Was ist .NET?

- ◆ Entwicklung mit klassischen System (hier VB6)



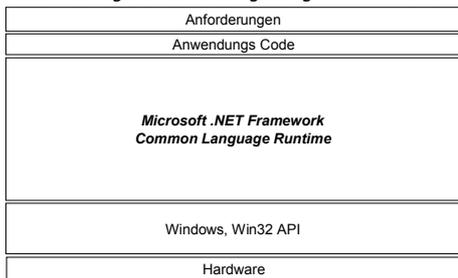
Paul Rohorzka

Wie weit ist .NET 2.0?

4

Was ist .NET?

- ◆ Neue mächtigere Laufzeitumgebung → .NET Framework

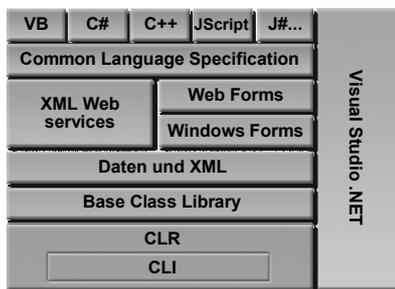


Paul Rohorzka

Wie weit ist .NET 2.0?

5

Das .NET Framework



Paul Rohorzka

Wie weit ist .NET 2.0?

6

Die Common Language Runtime (CLR)

- ◆ Kern der .NET-Ausführungseinheit
 - Objektorientiert
 - Kompiliert für die CLR zu "Managed Code"
- ◆ Aufgaben der CLR
 - Speicher Management
 - Multithreading
 - Durchsetzen von Sicherheitsrichtlinien
 - Grundlage der Interoperabilität
- ◆ Just-In-Time (JIT) Kompilierung
 - Managed Code wird erst am Zielsystem nativ kompiliert

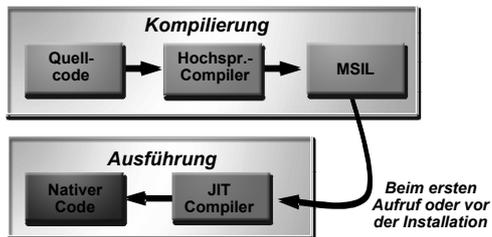
Paul Rohozka

Wie weit ist .NET 2.0?

7

Die Common Language Runtime (cont'd)

- ◆ Was ist der Vorteil von JIT?
 - Unterschiedliche Zielplattformen



Paul Rohozka

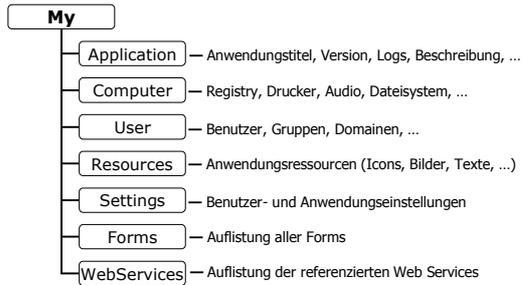
Wie weit ist .NET 2.0?

8

II. NewSpeak Spracherweiterung in VisualBasic 2005

My-Classes
Using-Anweisung
Partial Types
Operatorenüberladung

Spracherweiterungen: Die My-Klassen



Paul Rohorzka

Wie weit ist .NET 2.0?

10

Spracherweiterungen: Die My-Klassen (cont'd)

- ◆ Bieten Zugriff auf häufig benötigte Informationen
- ◆ Deutlich geringerer Kodieraufwand
- ◆ Erhöhung der Produktivität
- ◆ Unterstützung durch Intellisense!



Paul Rohorzka

Wie weit ist .NET 2.0?

11

Anweisungen

- ◆ Continue Anweisung (Do, For, While)
 - Springt zum nächsten Durchlauf der Schleife
- ◆ Using Anweisung
 - Vereinfacht die Freigabe von Ressourcen

```

Dim p As Pen = New Pen(c)           Using p As New Pen(c)
Try                                  g.DrawLine(p, 0, 0, 50, 50)
  g.DrawLine(p, 0, 0, 50, 50)      End Using
Finally
  If Not p Is Nothing Then
    CType(p, IDisposable).Dispose()
  End If
End Try
  
```

Paul Rohorzka

Wie weit ist .NET 2.0?

12

Generics

```
Public Class List(Of ItemType)
    Private elements() As ItemType
    Private count As Integer

    Public Sub Add(ByVal element As ItemType)
        If count = elements.Length Then Resize(count * 2)
        elements(count) = element : count += 1
    End Sub

    Public Default Property Item(ByVal index As Integer) As ItemType
        Get : Return elements(index) : End Get
        Set : elements(index) = value : End Set
    End Property
End Class
```

```
Dim intList As New List(Of Integer)
```

```
intList.Add(1)
intList.Add(2)
intList.Add("Hallo") ' Kompilier-Fehler

i = intList(0) ' Keine Typumwandlung erforderlich
```

Paul Rohorzka

Wie weit ist .NET 2.0?

13

Generics (cont'd)

- ◆ Wozu Generics?
 - Typsicherheit
 - Keine Typumwandlung → erhöhte Performance
- ◆ Wie funktionieren Generics?
 - Generic Type → CLR Template
 - Runtime → Kopie des Templates
 - Für jede Kombination von Typ-Parametern

Paul Rohorzka

Wie weit ist .NET 2.0?

14

Generics (cont'd)

- ◆ Typ-Parameter werden angewandt auf
 - Klassen, Strukturen, Interfaces, Delegates

```
Class Dictionary(Of KeyType, ValueType)
```

```
Structure Pair(Of FirstType, SecondType)
```

```
Dim Customers As Dictionary(Of String, Customer)
```

```
Dim Orders As Dictionary(Of String, List(Of Order))
```

```
Dim Spellings As Dictionary(Of Integer, String)
```

Paul Rohorzka

Wie weit ist .NET 2.0?

15

Partial Types

◆ Implementation eines Typs über mehrere Dateien

- Für Klassen und Strukturen (und Interfaces in C#)
- Deklaration mit neuem Typ-Modifizierer `partial`

◆ Bieten einige Vorteile

- Große Implementierungen können aufteilt werden
- Entwickler-Code kann von generiertem Code getrennt werden
 - Verhindert, dass eine Neugenerierung Änderungen überschreibt
 - Von WinForms und streng typisierten DataSets verwendet
- Mehrere Entwickler können am selben Typ arbeiten
- Ermöglicht "code-beside" im Unterschied zu "code-behind"
 - Wird von ASP.NET 2.0 verwendet
- Einfachere Wartung und Quellcode-Management

Paul Rohorzka

Wie weit ist .NET 2.0?

16

Partial Types (cont'd)

```
' Demo, Part1.vb
Imports System

Partial Public Class Demo
    Public Sub New()
        Console.WriteLine("P1")
    End Sub
End Class
```

```
' Demo, Part2.vb
Partial Public Class Demo
    Private i As Integer
End Class
```

```
' Demo, Part3.vb
' OK in Visual Basic
Public Class Demo
    ' Error!
    Private i As Integer
    ' OK
    Private j As Integer

    Public Sub Test()
        ' OK in Visual Basic
        Console.WriteLine("P3")
    End Sub
End Class
```

Paul Rohorzka

Wie weit ist .NET 2.0?

17

Überladen von Operatoren

◆ Standard-Operatoren können nun überladen werden

- Unterstützt Klassen und Strukturen

```
' Class with overload of +
Public Class Dem
    Private x As Byte
    Private y As Byte

    Public Sub New(ByVal i As Byte, _
        ByVal j As Byte)
        Me.x = i
        Me.y = j
    End Sub

    Public Overrides Function _
        ToString() As String
        Return String.Format( _
            "{0},{1}", x, y)
    End Function
End Class
```

```
Public Shared Operator +( _
    ByVal d1 As Dem, _
    ByVal d2 As Dem) As Dem
    Return New Dem(d1.x + d2.x, _
        d1.y + d2.y)
End Operator

End Class
' Test code
Dim d1 As Dem = New Dem(1, 0)
Dim d2 As Dem = New Dem(2, 2)
Dim d3 As Dem = d1 + d2
d3 += d1
Trace.WriteLine(d3) ' Prints [4,2]
```

Paul Rohorzka

Wie weit ist .NET 2.0?

18

Weitere Verbesserungen

- ◆ Vorzeichenlose Ganzzahltypen
 - Zusätzliche Typen `UShort`, `UInteger` und `ULong`
- ◆ Neuer Operator `IsNot`
 - Vermeidet sperrige Syntax mit `Not` und `Is`
- ◆ Explizite Untergrenze 0 bei Arrays
 - Macht Grenzen klarer, z.B. `Dim b(0 To 7) As Byte`
- ◆ Neue Compiler-Features
 - CLS compliance und Überprüfung auf uninitialisierte Variablen
 - Neue Optionen inklusive Warning-Steuerung
- ◆ XML Dokumentation Kommentare mit `'''`

III. Ran an die Daten! **ADO.NET**

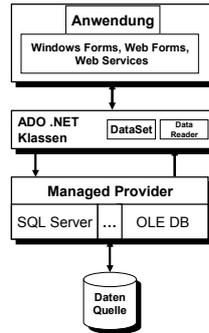
Architektur
Neues in ADO.NET 2.0
DataBinding in WinForms
DataGridView

Das ADO.NET Modell

- ◆ Daten holen
 - Verbinden und holen
 - Statischer Vorwärts- / Read Only Cursor
 - Kein Locking
- ◆ Nicht verbunden arbeiten (disconnected)
 - XML Unterstützung (offline Persistierung)
 - Ähnlich Recordset
- ◆ Daten aktualisieren
 - Verbinden und SQL ausführen

ADO.NET Architektur

- ◆ Managed Provider
 - Zugriff auf Datenquellen
 - *Datenquellenklassen*
 - Connection
 - Command, Parameter
 - DataReader
- ◆ Offline Klassen
 - DataTable
 - DataSet (mehrere DataTables)



Paul Rohorzka

Wie weit ist .NET 2.0?

22

Verbesserungen bezüglich Performance

- ◆ Interne Indizierung der Datensätze
 - Einfügen und löschen gehen $\log(n)$
 - Aktualisierungen fast konstant
- ◆ Binäre Serialisierung der Daten
 - 1.x-Datensätze konnten nur in XML serialisieren
 - Praktische für Datenaustausch
 - Schlecht für die Performance
 - Binäre Serialisierung ist nun eine Option in 2.0
 - Schnell und kompakt, besonders für viele Datensätze
 - `DataSet.RemotingFormat = SerializationFormat.Binary`

Paul Rohorzka

Wie weit ist .NET 2.0?

23

DataTable Instanzen ohne DataSet

- ◆ Wichtige DataSet-Operationen nun auch direkt auf DataTable möglich
 - ReadXml, ReadXmlSchema, WriteXml, WriteXmlSchema
 - Clear, Clone, Copy, Merge, GetChanges
- ◆ DataTable ist nun "auto-serializable":
 - Eine DataTable Instanz kann direkt von einem Web Service oder via Remoting übernommen werden.

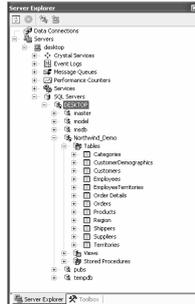
Paul Rohorzka

Wie weit ist .NET 2.0?

24

Graphische Datenwerkzeuge: Server Explorer

- ◆ Sehen und verändern
 - Daten
 - Datenbank-Schema
- ◆ Drag & drop der Datenbank-Objekte
 - Tabellen & Sichten
 - Stored Procedures



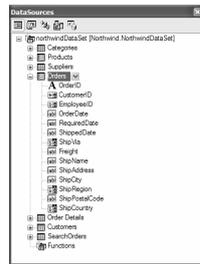
Paul Rohorzka

Wie weit ist .NET 2.0?

31

DataSources in Visual Studio 2005 Drag & Drop auf ein UI

- ◆ Data Binding
- ◆ Drag & Drop der Spalten
- ◆ Wählbare UI-Steerelemente
 - Grid controls
 - Individual controls

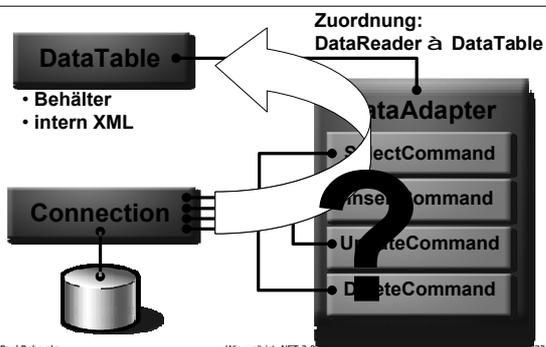


Paul Rohorzka

Wie weit ist .NET 2.0?

32

Der DataAdapter



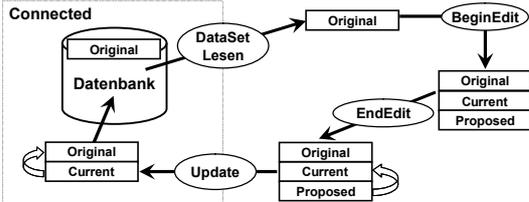
Paul Rohorzka

Wie weit ist .NET 2.0?

33

DataTable Funktionalität Versioning of DataRows

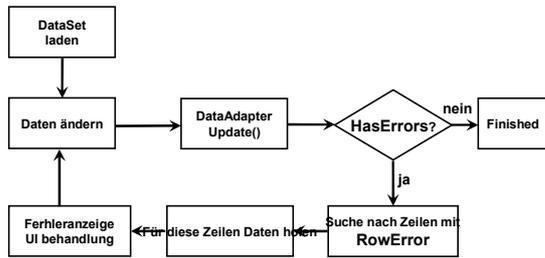
- ◆ Drei Versionen der Datensätze sind max. gleichzeitig verfügbar.
 - Original, Current, Proposed
 - myRow("feld", DataRowVersion.Current)



Paul Rohorzka Wie weit ist .NET 2.0? 34

Concurrency und ADO.NET

- ◆ Ähnlich wie "Read-For-Update"



Paul Rohorzka Wie weit ist .NET 2.0? 35

IV. Mit den Daten ins Netz ASP.NET

Architektur
Databinding
MasterPages

Klassische Entwicklung für das Web

```
<html><body><% Dim rs, fld
Set rs = Server.CreateObject("ADODB.RecordSet")
rs.Open "SELECT * FROM tblKunden", ...

If Not rs.EOF Then
Response.Write("<table><tr>")
For Each fld In rs.Fields
Response.Write("<td>" & fld.Name & "</td>")
Next
Response.Write("</tr>")
While Not rs.EOF
Response.Write("<tr>")
... Read the Field Values here
Response.Write("</tr>")
Wend
Response.Write("</table>")
End If %></body></html>
```

Paul Rohorzka

Wie weit ist .NET 2.0?

37

Klassische Entwicklung für das Web Was sind die Probleme?

- ◆ Skriptsprachen
 - Keine Typsicherheit
 - Werden interpretiert → Performance leidet
- ◆ Programmier-Modell
 - Unstrukturiert
 - UI und code sind in einer Datei
 - Wiederverwendbarkeit?

Paul Rohorzka

Wie weit ist .NET 2.0?

38

Die ASP.NET-Plattform

- ◆ Konzeptuell
 - Streng typisiert
 - Objektorientiert → Wiederverwendbarkeit
- ◆ Performance & Zuverlässigkeit
 - Vorkompilierte Seiten
 - Isolieren von Anwendungen

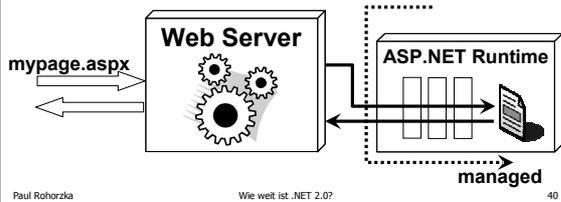
Paul Rohorzka

Wie weit ist .NET 2.0?

39

ASP.NET Architektur

- ◆ Web server unterstützt ASP.NET
 - Kennt die .NET-Runtime
 - Leitet Anfragen an ASP.NET weiter

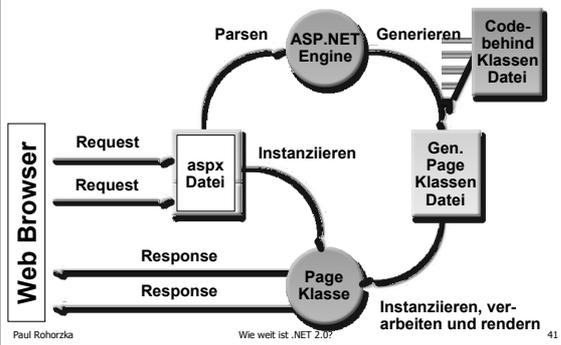


Paul Rohorzka

Wie weit ist .NET 2.0?

40

ASP.NET Modell ASP.NET – Just-In-Time Kompilierung



Paul Rohorzka

Wie weit ist .NET 2.0?

41

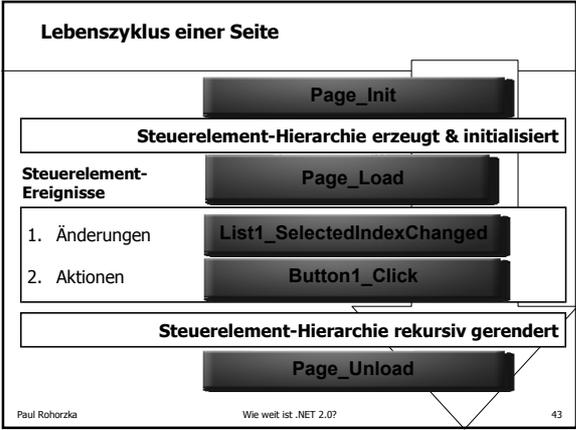
ASP.NET Server Steuerelemente

- ◆ Kapseln das UI-Rendering
 - Ähnlich VB-Steuerelemente
 - Objektorientiertes Konzept
- ◆ Arten von Steuerelementen
 - HTML-Steuerelemente
 - ASP.NET Web Controls
- ◆ Je nach Browser
 - Version des Browsers wird abgefragt
 - Generierter HTML-Code wird optimiert

Paul Rohorzka

Wie weit ist .NET 2.0?

42



Ereignis-Verarbeitung

- ◆ Client-seitige Ereignisse
 - Werden vom Browser ausgeführt
 - JavaScript oder VBScript
- ◆ Server-seitige Ereignisse
 - Erzeugt durch Postback auf die selbe Seite
 - <input type="submit" />
 - JavaScript (doPostBack())
 - Seite und Steuerelement-Instanzen werden erzeugt
 - Ereignisse werden Steuerelementen zugeordnet

Paul Rohorzka Wie weit ist .NET 2.0? 44

ASP.NET Datenbindung

- ◆ Steuerelemente an Daten binden
 - "Single binding" – einzelnen Wert binden (z.B. Text)
 - "Repeated binding" – mehrere Werte binden (z.B. ListBox-Items)
- ◆ Schritte zur Datenbindung
 - Daten von der Quelle holen
 - Eigenschaften DataSource / DataMember initialisieren
 - Aufruf von DataBind() ⇒ Rendering

Paul Rohorzka Wie weit ist .NET 2.0? 45

ASP.NET Databinding

- ◆ Generelle Syntax: <%# source %>
- ◆ Bindung eines einzigen Wertes
 - An Wert eines Members
 - <asp:TextBox Text=' <%# Session("my")%> ' />
 - An Methode
 - <asp:TextBox Text=' <%# foo(a, b, c)%> ' />
 - An andere Ausdrücke
- ◆ "Repeated value binding"

Paul Rohorzka

Wie weit ist .NET 2.0?

46

Repeated Value Binding

- ◆ Bindung an
 - DataSets, DataViews, Collections,...
- ◆ Für Steuerelemente wie
 - HTML <select>
 - ListBox, DropDownList, CheckBoxList
- ◆ Wichtige Eigenschaften und Methoden
 - DataSource, DataMember
 - DataTextField, DataValueField
 - DataBind()

Paul Rohorzka

Wie weit ist .NET 2.0?

47

DataSource-Konzept in ASP.NET 2.0

- ◆ Deklaratives ("no-code") DataBinding

<i>Name</i>	<i>Beschreibung</i>
SqlDataSource	Verbindet datengeb. Steuerel. mit SQL Datenbanken
XmlDataSource	Verbindet datengeb. Steuerel. mit XML Daten
ObjectDataSource	Verbindet datengeb. Steuerel. mit Datenkomponenten
SiteMapDataSource	Verbindet Navigations-Steuerel. mit SiteMap-Daten

Paul Rohorzka

Wie weit ist .NET 2.0?

48

SqlDataSource

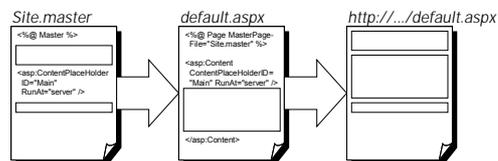
- ◆ Bindung an SQL Datenbank
 - Jede Datenbank (nicht nur Microsoft SQL)
- ◆ Zweiweg-Bindung
 - SelectCommand → Abfrage
 - InsertCommand, UpdateCommand und DeleteCommand → Aktualisierung
- ◆ Optional: Caching von Abfrageergebnissen
- ◆ Parametrisierbare Abarbeitung

Die wichtigsten Eigenschaften

Name	Beschreibung
ConnectionString	Connection string zur Verbindung zur Datenquelle
SelectCommand	Befehl für Auswahlabfragen
InsertCommand	Befehl für Einfügeoperationen
UpdateCommand	Befehl für Aktualisierungen
DeleteCommand	Befehl für Löschoptionen
DataSourceMode	Gibt an, ob DataSet oder DataReader verwendet werden soll (Standard ist DataSet) – Aktualisierbarkeit!
ProviderName	Provider (Standard ist SQL Server .NET Provider)

Master Pages

- ◆ Definieren gemeinsamen Inhalt für mehrere Seiten
 - Platzhalter (<asp:Content>) für Kind-Seiten
 - Content-Seiten referenzieren Masters und befüllen die Platzhalter



Definieren einer Master Page

```
<%@ Master Language="C#" %>
<html>
<body>
  <!-- Banner für alle Seiten, die diesen Master nutzen -->
  <table border="1" style="width: 100%; text-align: center;">
    <tr>
      <td background="darkblue" style="color: white;">
        <span style="font-size: 24pt; color: white;">AER 8
        </span>
      </td>
    </tr>
  </table>

  <!-- Placeholder for content below banner -->
  <asp:ContentPlaceHolder ID="Main" RunAt="server" />
</body>
</html>
```

Paul Rohozka

Wie weit ist .NET 2.0?

52

Eine Master Page verwenden

```
<%@ Page MasterPageFile="~/Site.master" %>
<asp:Content ContentPlaceHolderID="Main" RunAt="server">
  Dieser Content befüllt den Platzhalter "Main",
  der in der Master Page definiert ist.
</asp:Content>
```

```
<configuration>
  <system.web>
    <pages masterPageFile="~/Site.master" />
  </system.web>
</configuration>
```

```
Sub Page_PrePageEventArgs_OnPreRender(ByVal e As EventArgs)
  Page.MasterPageFile = "~/Site.master"
End Sub Page_PreRender
```

Paul Rohozka

Wie weit ist .NET 2.0?

53

Persönliches Fazit

- J** Für die meisten Aufgaben kaum mehr Codieraufwand notwendig
- K** Aus der Vielzahl der angebotenen Möglichkeiten ist es nicht immer leicht, die günstigste Variante zu finden

But... It's cool!

Paul Rohozka

Wie weit ist .NET 2.0?

54
