



complement
getting ideas done

Microsoft Visual Studio 2008 Anwendungsentwicklung für Vista

„Vista Revisited“ - NIK Nürnberg

28.02.2008



complement
getting ideas done

complement AG

Thomas Hemmer
CTO

thomas.hemmer@complement.de

Microsoft Visual Studio 2008

Anwendungsentwicklung für Vista - Agenda

VisualStudio 2008 – Plattform für Vista Anwendungen

LINQ – Language Integrated Query

Vista und WPF - **Revisited**

VisualStudio 2008 & .NET 3.5 im Schnelldurchlauf



complement
getting ideas done

Microsoft Visual Studio 2008

Anwendungsentwicklung für Vista - Agenda

VisualStudio 2008 – Plattform für Vista Anwendungen

LINQ – Language Integrated Query

Vista und WPF - **Revisited**

VisualStudio 2008 & .NET 3.5 im Schnelldurchlauf



complement
getting ideas done

.NET Evolution

	2002	2003	2005	2006	2008
Tool (Visual Studio)	VS.NET 2002	VS.NET 2003	Visual Studio 2005	VS2005 + Extensions	VS2008
Languages	C# v1.0 VB.NET (v7.0)	C# v1.1 VB.NET (v7.1)	C# v2.0 VB2005 (v8.0)		C# v3.0 VB9
Framework Libraries	NetFx v1.0	NetFx v1.1	NetFx v2.0	NetFx v3.0	NetFx v3.5
Engine (CLR)	CLR v1.0	CLR v1.1	CLR v2.0		

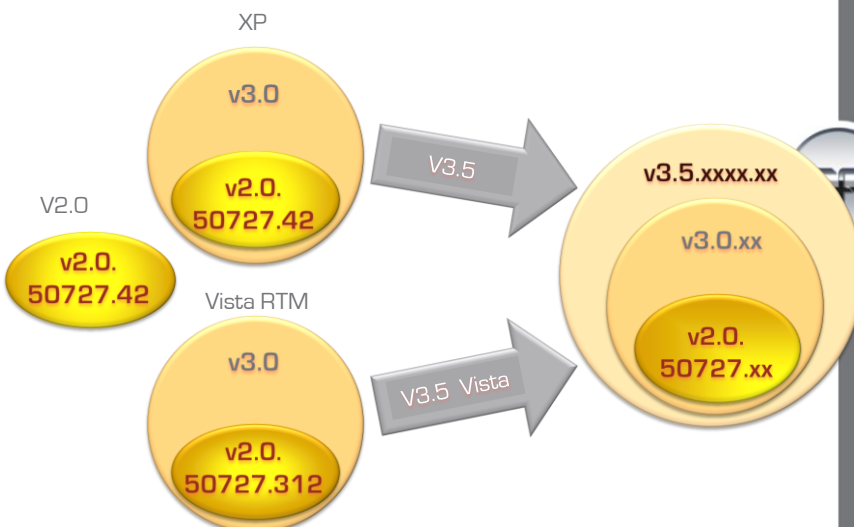


5

© complement AG 2008. All rights reserved.

Donnerstag, 28. Februar 2008

.NET Evolution



6

© complement AG 2008. All rights reserved.

Donnerstag, 28. Februar 2008

Update auf Visual Studio 2008

- Problemlos von Visual Studio 2005
- „no brainer“
 - Visual Studio 2008 „Multi Targeting“
 - .NET 3.5 ist Obermenge von .NET 2.0 / 3.0
- Visual Studio 2008 ist Microsofts Plattform für Entwicklung von
 - Client-Anwendungen
 - Office-Erweiterungen
 - Web-Anwendungen, Server-Anwendungen, Mobile ...



complement
getting ideas done

Update auf Visual Studio 2008



Visual Studio 2008

- + Service Pack 1
- + SP1 Update for Vista
- + WF Extensions
- + WPF & WCF Extensions
- + SharePoint Workflow
- + Visual Studio Tools for Office Second Edition
- + ASP.NET AJAX Extensions
- + Device Emulator v2.0
- + .NETCF v2.0 SP2
- + WM 5.0 Pocket PC SDK
- + WM5.0 Smartphone SDK



complement
getting ideas done

Microsoft Visual Studio 2008

Anwendungsentwicklung für Vista - Agenda

VisualStudio 2008 – Plattform für Vista Anwendungen

LINQ – Language Integrated Query

Vista und WPF - **Revisited**

VisualStudio 2008 & .NET 3.5 im Schnelldurchlauf



complement
getting ideas done

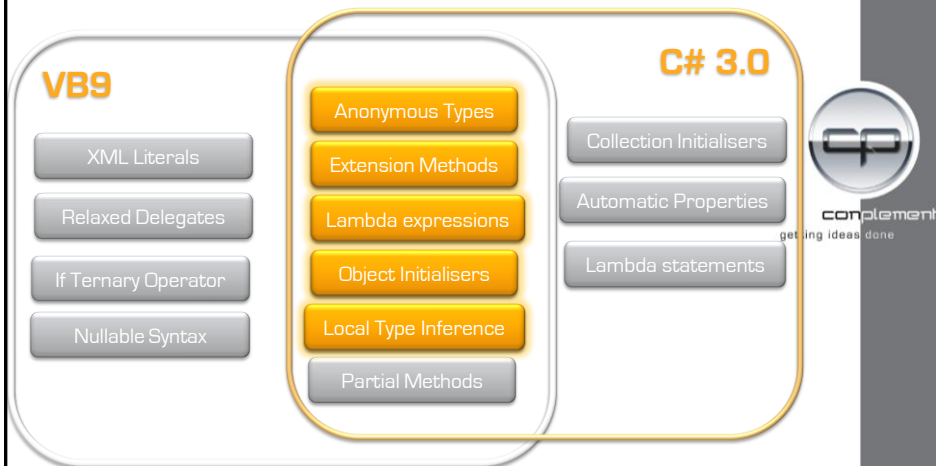
C# 3.0 - Entwurfsziele

- Integration von Objekten, relationalen Daten und XML (LINQ)
- Sowie
 - Verbesserung der Prägnanz der Sprache
 - Hinzufügen funktionaler Konstrukte
 - Vermeidung von enger Kopplung an bestimmte APIs
 - 100 % abwärtskompatibel



complement
getting ideas done

C# 3.0 – Erweiterungen für LINQ



Local Variable Type Inference

```
int i = 42;  
string s = "NIK";  
double d = 3.14;  
int[] numbers = new int[] {1, 2, 3};  
Dictionary<int,Order> orders = new  
    Dictionary<int,Order>();
```

```
var i = 42;  
var s = "NIK";  
var d = 3.14;  
var numbers = new int[] {1, 2, 3};  
var orders = new Dictionary<int,Order>();
```

"Der Datentyp auf der rechten Seite der Zuweisung"

Object Initialisers / Collection Initialisers

```
public class Point
{
    private int x, y;

    public int X { get { return x; } set { x = value; } }
    public int Y { get { return y; } set { y = value; } }
}
```



complement
getting ideas done

```
Point a = new Point { X = 0, Y = 1 };
```

```
Point a = new Point();
a.X = 0;
a.Y = 1;
```

Property Zuweisung

Object Initialisers / Collection Initialisers

Einschub: Auto-Implemented Properties

```
public class Point
{
    public int X { get; set; }
    public int Y { get; set; }
}
```

Auto implemented
Properties

```
Point a = new Point { X = 0, Y = 1 };
```

```
Point a = new Point();
a.X = 0;
a.Y = 1;
```

Property Zuweisung



complement
getting ideas done

Anonymous Types

XYZ

```
class XYZ
{
    public string Name;
    public int Age;
}
```

```
var o = new { Name = "Jenny", Age = 31 };
```

```
var results = new[]
{
    new { Name = "P1", Age = 5 },
    new { Name = "P2", Age = 6 }
};
```



complement
getting ideas done

Extension Methods

```
static bool MyMethod(Person p)
{
    //return MyExtensions.IsOver15(p);
    return p.IsOver15();
}

public static class MyExtensions
{
    public static bool IsOver15(this Person p)
    {
        return p.Age > 15;
    }
}
```



complement
getting ideas done

Extension Methods

Extension Method

```
namespace MyExt
{
    public static class Extension
    {
        public static string Concatenate(
            this IEnumerable<string> strings,
            string separator) {...}
    }
}
```

```
using MyExt;
```

Extensions im aktuellen Scope
bekanntmachen

```
string[] names = new string[] { "Jenny",
    "Daniel", "Rita" };
string s = names.Concatenate(", ");
```

IntelliSense!

obj.Foo(x, y)
↓
XXX.Foo(obj, x, y)



complement
getting ideas done

Query Expressions

- Queries bestehen aus Methodenaufrufen

```
from c in customers
where c.City == "Hove"
select new { c.Name, c.Phone };
```

```
customers
.Where[c => c.City == "Hove"]
.Select[c => new { c.Name, c.Phone }];
```



complement
getting ideas done

Query Expressions

```
var contacts =  
    from c in customers  
    where c.City == "Hove"  
    select new { c.Name, c.Phone };
```

Query expressions

Local variable type inference

```
var contacts =  
    customers  
    .Where(c => c.City == "Hove")  
    .Select(c => new { c.Name, c.Phone });
```

Lambda expressions

Extension methods

Anonymous types

Object initializers



complement
getting ideas done

```
static void Main()  
{  
    int[] numbers = { 5, 10, 8, 3, 6, 12};  
  
    //Query syntax:  
    IEnumerable<int> numQuery1 =  
        from num in numbers  
        where num % 2 == 0  
        orderby num  
        select num;  
  
    //Method syntax:  
    IEnumerable<int> numQuery2 = numbers.Where(num => num % 2 == 0).OrderBy(n => n);  
  
    foreach (int i in numQuery1)  
    {  
        Console.Write(i + " ");  
    }  
    Console.WriteLine(System.Environment.NewLine);  
    foreach (int i in numQuery2)  
    {  
        Console.Write(i + " ");  
    }  
  
    Console.ReadKey();  
}
```



complement
getting ideas done

Query Expressions

```
from c in Customers
where c.City == "Nürnberg"
select new { c.Name, c.Address };
```



complement
getting ideas done

LINQ Architektur

```
var query = from c in customers where c.City == "Hove" select c.Name;
```

```
var query = customers.Where(c => c.City == "Hove").Select(c => c.Name);
```

Quelle implementiert
IEnumerable<T>

Quelle implementiert
IQueryable<T>

System.Linq.Enumerable
Delegate basiert

System.Linq.Queryable
Expression tree basiert

Objects

SQL

DataSets

Weitere...



complement
getting ideas done

So what´s LINQ ?

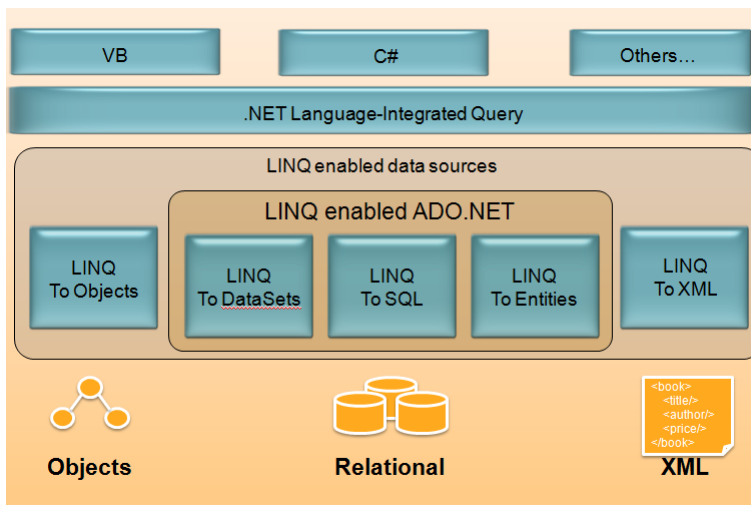
LINQ = Spracherweiterungen + Query Expressions

- LINQ kann mit allen erdenklichen Datenquellen arbeiten
 - Objekte, SQL, XML, Filesystem, SharePoint, LDAP
- LINQ baut auf anderen neuen Features auf
 - Type inference, object initialisers, anonymous types, extension methods, lambda expressions
 - IEnumerable<T> and IQueryable<T> im System.Linq Namespace
 - Query Expressions
- Lambdas können an Expression Trees gebunden werden



complement
getting ideas done

LINQ - Solution Domain



complement
getting ideas done

LINQ to SQL im Zusammenspiel mit LINQ to XML

```
XElement londonCustomers =  
    new XElement("Customers", from c in db.Customers where  
        c.City == "London" select  
        new XElement("Customer",  
            new XAttribute("CustomerID", c.CustomerID),  
            new XElement("Name", c.ContactName),  
            new XElement("Phone", c.Phone) ]  
    );
```

Console.WriteLine

```
<Customers>  
  <Customer CustomerID="AROUT">  
    <Name>Mark Harrington</Name>  
    <Phone>(171) 555-0188</Phone>  
  </Customer>  
  <Customer CustomerID="BSBEV">  
    <Name>Michelle Alexander</Name>  
    <Phone>(171) 555-0112</Phone>  
  </Customer>  
  <Customer CustomerID="CONSH">  
    <Name>Nicole Holliday</Name>  
    <Phone>(171) 555-0182</Phone>  
  </Customer>  
</Customers>
```



LINQ to SQL im Zusammenspiel mit LINQ to XML

```
<customerUpdates>  
  <customerUpdate>  
    <custid>ALFKI</custid>  
    <phone>206-555-0103</phone>  
  </customerUpdate>  
  <customerUpdate>  
    <custid>EASTC</custid>  
    <phone>425-555-0143</phone>  
  </customerUpdate>  
</customerUpdates>
```

```
<foreach (var cu in customerUpdates.Elements("customerUpdate"))  
{  
    Customer cust = db.Customers.First  
        (c => c.CustomerID == [string]cu.Element("custid"));  
    cust.Phone = [string]cu.Element("phone");  
}  
db.SubmitChanges();
```



Microsoft Visual Studio 2008

Anwendungsentwicklung für Vista - Agenda

VisualStudio 2008 – Plattform für Vista Anwendungen

LINQ – Language Integrated Query

Vista und WPF - **Revisited**

VisualStudio 2008 & .NET 3.5 im Schnelldurchlauf



complement
getting ideas done

DEMO



complement
getting ideas done

Vista & .NET 3.5

- Vista-only APIs im .NET Framework 3.5
 - Windows Forms CommonFileDialog ist Vista Dialog
 - Peer-to-Peer APIs in System.Net.dll
 - Vista-specific Namespaces in System.Core.dll
 - PerformanceData
 - Diagnostics.Eventing
 - Security
 - Performance ist teilweise besser unter Vista



complement
getting ideas done

Vista & VisualStudio 2008

- Vista-only APIs – Win32
 - Von .NET aus nur über *p/invoke* oder *managed wrapper* verfügbar
 - “Vista Bridge” mit Beispiel-Anwendungen und *managed wrapper* Implementierungen



complement
getting ideas done

Microsoft Visual Studio 2008 Anwendungsentwicklung für Vista - Agenda

VisualStudio 2008 – Plattform für Vista Anwendungen

LINQ – Language Integrated Query

Vista und WPF - **Revisited**

VisualStudio 2008 & .NET 3.5 im Schnelldurchlauf



complement
getting ideas done

Einige weitere Features von VisualStudio 2008 / .NET 3.5

- WCF
 - “SOAPless” HTTP communication
 - XML / JSON Serialisierung
 - RSS / ATOM Unterstützung

- MS Office Anwendungen
 - Visual Studio Tools for Office
 - Für MS Office 2003 / 2007
 - Support für Workflow / SharePoint



complement
getting ideas done

DEMO



complement
getting ideas done

Weitere Features von .NET 3.5

Neuerungen in ...

- System.Core.dll
- System.Data.DataSetExtensions.dll
- System.Web.Extensions.dll
- System.WorkflowServices.dll
- System.ServiceModel.Web.dll
- System.AddIn.dll, System.AddIn.Contract.dll
- System.Windows.Presentation.dll
- System.Net.dll
- System.DirectoryServices.AccountManagement.dll
- System.Management.Instrumentation.dll
- System.VisualC.STLCLR.dll



complement
getting ideas done

Fragen ?



Kontakt:

Thomas Hemmer

complement AG

Südwestpark 92

90449 Nürnberg

+49 911 255097620

thomas.hemmer@complement.de



complement
getting ideas done

Referenzen, weiterführende Links

Windows Vista User Experience Guidelines:

- <http://msdn2.microsoft.com/en-us/library/aa511258.aspx>

Vista Bridge (managed wrapper Beispiele für Vista-Win32-APIs)

- <http://msdn2.microsoft.com/en-us/library/ms756482.aspx>

Microsoft NET 3.x Community

- <http://netfx3.com> <http://wpf.netfx3.com/> <http://wcf.netfx.com>

Microsoft Vista Home

- <http://www.microsoft.com/windows/products/windowsvista/default.aspx>
- <http://msdn2.microsoft.com/en-us/windowsvista/default.aspx>



complement
getting ideas done

Referenzen, weiterführende Links

The LINQ Project @ Microsoft:

- <http://msdn2.microsoft.com/en-us/netframework/aa904594.aspx>

LINQ auf MSDN Library:

- [http://msdn2.microsoft.com/en-us/library/bb397926\(VS.90\).aspx](http://msdn2.microsoft.com/en-us/library/bb397926(VS.90).aspx)

Daniel Moth Blog (Microsoft):

- <http://www.danielmoth.com/Blog/>

Virtual Labs zu LINQ auf msdn.microsoft.com

- <http://msevents.microsoft.com/CUI/SearchDisplay.aspx?culture=en-US&tgAudHero=2#eventType=4;culture=en-US;sortKey=;sortOrder=;pageEvent=false;hdnInitialCount=;kwdAny=linq;searchcontrol=yes>



complement
getting ideas done