

BESSERE GESCHÄFTSANWENDUNGEN MIT GUI-CONTROL-SUITEN

Robert Walter

INdotNET – Ingolstädter .NET User Group

www.indot.net

10/2017 - 12/2017

UNTER WPF

AUSGANGSLAGE

**Herausforderung:
Welcher GUI-Control Hersteller ist der Beste?**



Die Standard-GUI-Komponenten aus Windows Forms, WPF, und ASP.NET sind oft nur rudimentär gehalten oder teilweise sogar zu leistungsschwach. Sobald Microsoft einen Satz an GUI-Controls zu einer Oberflächentechnologie erstellt hat, werden sie nahezu nicht mehr erweitert. Deswegen gibt es schon seit der PC-Eisenzeit Drittanbieter, die umfangreichere und bessere .NET-Controls anbieten.




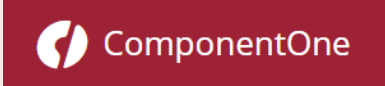



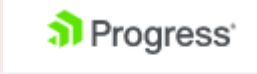

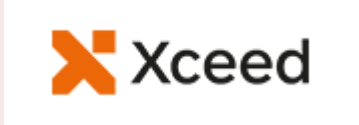
Wir werden verschiedenste GUI-Suiten unterschiedlicher Hersteller anschauen, und dabei untersuchen, wie sie das ein oder andere gängige Problem elegant oder vielleicht mit Abstrichen gelöst haben. Hier ein paar Beispiele:

- *Kann man eine Million Datensätze schnell und flüssig darstellen?*
- *Mit welchem Weg kann man Formulare schneller gestalten?*
- *Kann man das Einfrieren einer Anwendung einfach unterbinden?*
- *Ist es möglich, mehrere Informationen in einer Combobox unterzubringen?*
- *Kann man eine Suche über mehrere Felder - ganz ohne Programmierung - zur Verfügung zu stellen?*



GUI-CONTROL-SUITE HERSTELLER

(A-Z)

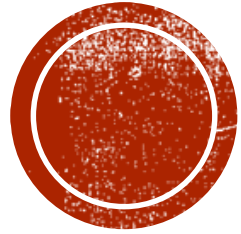
<p>actipro www.actiprosoftware.com</p> 	<p>DevExpress www.devexpress.com</p> 	<p>GrapeCity / ComponentOne www.grapecity.com</p>  	<p>INFRAGISTICS www.infragistics.com</p> 
<p>Nevron www.nevron.com</p> 	<p>Syncfusion www.syncfusion.com</p> 	<p>Progress / Telerik www.telerik.com</p>  	<p>Xceed www.xceed.com</p> 

Inhalt dieser Präsentation:

Analyse durchgeführt

Technische Probleme





DAS WPF-DATAGRID



DAS WPF-DATAGRID

Nr	Vorname	Nachname	Geburtsdatum	...
1	Mr.	.NET	01.01.2001	...
2	Mrs.	Millenium	01.01.2000	...

Fakten:

- Das Datagrid zeigt die meisten Informationen aus einer Liste an. Es ist unverzichtbar für Geschäftsanwendungen – und es ist eines der komplexesten Controls, die es gibt!
- In der ersten Version von WPF war noch kein DataGrid enthalten –
Stellungnahme: ListView verwenden... Was aber, wenn das nicht ausreicht?
Später wurde das DataGrid hinzugefügt...
- Auch im Jahr 2017 (!) wird das WPF-DataGrid bei Drittanbietern noch weiterentwickelt (z.B. Performanceoptimierungen)
- Viele DataGrid-Architekturen bei Drittherstellern waren nicht leistungsfähig genug, sodass es inzwischen sogar mehrere DataGrid-Controls bei einem Hersteller gibt!



PROBLEM: WPF-DATAGRID PERFORMANCE

Ein nachvollziehbares Beispiel:

1. Setze ein neues WPF-Projekt auf
2. Füge ein DataGrid hinzu
3. Lade 2.000 Objekte mit 20 Spalten
4. Scrolle mit dem Scrollbalken hoch und runter

Nummer	String1	String2	String3	Preis	IsTrue	String4	String5
1	String 1	012345678910111213141516171	0,621631760439964-0,870251818499645	291,01	<input checked="" type="checkbox"/>	String 1	012345678910111213141516171
2	String 2	012345678	0,373336678079021-0,917082117366177	566,53	<input type="checkbox"/>	String 2	012345678
3	String 3	012345678910111213141516171	0,584165975723493-0,0240762797296402	80,56	<input type="checkbox"/>	String 3	012345678910111213141516171
4	String 4	012345678910111213141516171	0,106324951213936-0,815570040054419	711,81	<input checked="" type="checkbox"/>	String 4	012345678910111213141516171
5	String 5	012345678910111213141516171	0,985320804633815-0,278246732092577	719,28	<input checked="" type="checkbox"/>	String 5	012345678910111213141516171
6	String 6	012345678910111213141516171	0,218752895118088-0,892813560968644	247,85	<input type="checkbox"/>	String 6	012345678910111213141516171
7	String 7	012345	0,0360425873827388-0,313887281023845	180,50	<input type="checkbox"/>	String 7	012345
8	String 8	012345678910111213141516171	0,0370554635473785-0,382227954167047	887,74	<input checked="" type="checkbox"/>	String 8	012345678910111213141516171
9	String 9	012345678910111213141516171	0,454944455276683-0,989851186978561	404,30	<input type="checkbox"/>	String 9	012345678910111213141516171
10	String 10	012345678910111213141516171	0,381725168964698-0,316542762944728	741,14	<input checked="" type="checkbox"/>	String 10	012345678910111213141516171
11	String 11	012345678910111213141516171	0,455491341396929-0,547984961209812	986,93	<input type="checkbox"/>	String 11	012345678910111213141516171
12	String 12	01234567891011	0,343385028347087-0,150975464913517	484,25	<input checked="" type="checkbox"/>	String 12	012345678910111213141516171
13	String 13	0123456	0,480523117110377-0,669235564148629	426,33	<input type="checkbox"/>	String 13	0123456
14	String 14	012345678910111213141516171	0,270278694699648-0,843578220737902	878,28	<input type="checkbox"/>	String 14	012345678910111213141516171
15	String 15	012345678910111213141516171	0,724096997046888-0,927829612012873	819,98	<input type="checkbox"/>	String 15	012345678910111213141516171
16	String 16	01234567891011	0,995469617655254-0,873951618035301	561,69	<input type="checkbox"/>	String 16	012345678910111213141516171
17	String 17	0123456789101112131415	0,90221013217336-0,151675490267424	552,03	<input checked="" type="checkbox"/>	String 17	012345678910111213141516171
18	String 18	012345678910111213141516171	0,488057626172927-0,326958968922011	842,42	<input type="checkbox"/>	String 18	012345678910111213141516171
19	String 19	012345678910111213141516171	0,886079998633862-0,897981288795351	948,76	<input checked="" type="checkbox"/>	String 19	012345678910111213141516171
20	String 20	012345678910111213141516171	0,785708891128054-0,563520616182834	991,23	<input type="checkbox"/>	String 20	012345678910111213141516171
21	String 21	012345678910111213141516171	0,538146948226796-0,438258168025994	564,03	<input type="checkbox"/>	String 21	012345678910111213141516171

→ Das DataGrid ist träge, hängt nach, und hat störende Verzögerungen, bis man die gewünschten Datensätze sieht!



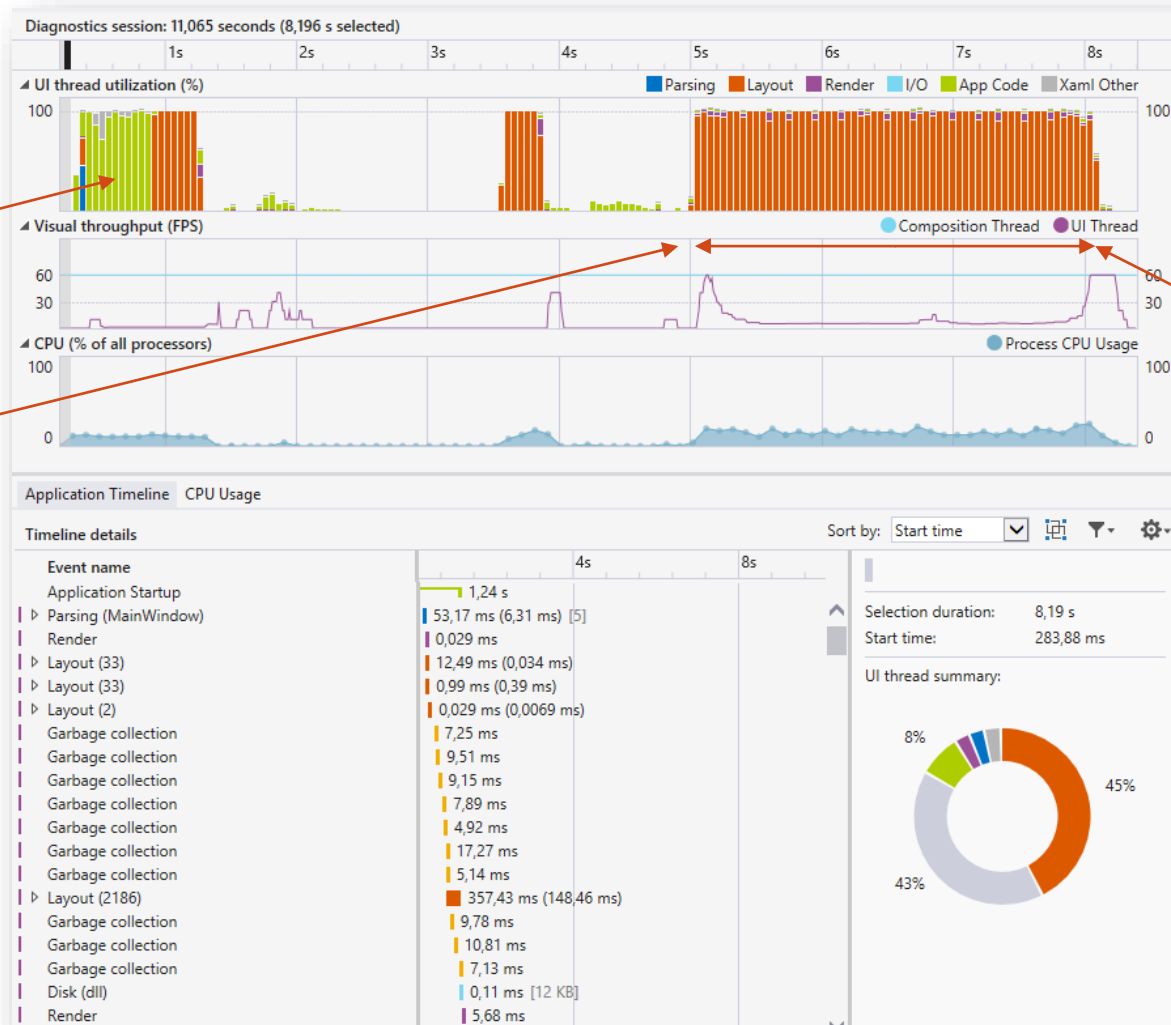
5. Zusatz: Stelle eine Standardsortierung per Code ein.
→ Der Sortierungsindikator wird nicht angezeigt!



ANALYSE: WPF STANDARD DATAGRID PERFORMANCE

Daten generieren

Scrollen



98% Layout
~ 1% Render!!



ANALYSE: WPF DRITTANBIETER DATAGRID PERFORMANCE

Daten generieren

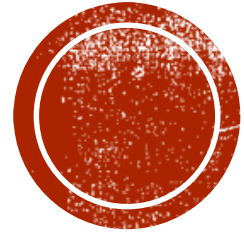
Scrollen



55% Layout
~ 1% Render!!

→ Dritthersteller haben das Problem im Griff!





WEITERE GÄNGIGE GUI-KONZEPT- & WPF-PROBLEME

PROBLEM: ZAHLEN-TEXTBOX

Eine TextBox soll so eingeschränkt werden, sodass nur Integer erlaubt sind.

Gängige Herangehensweisen:

- a) `IDataErrorInfo` implementieren, um Warnhinweis zu generieren.
- b) **Eine** `System.Windows.Forms.MaskedTextBox` **mit** `WindowsFormsHost` **hosten.**
(auch eine Windows Forms TextBox kann es! ABER: Eine ungültige Eingabe mit DataBinding lässt den Benutzer das Feld nicht mehr verlassen!)

→ Lösungsansatz von Drittanbietern: **MaskedTextBox** oder **typisierte TextBox**

Datentyp bzw. DataAnnotation auslesen, und optimales Controls dafür anbieten.



PROBLEM: FLEXIBLE SUCHE IM DATAGRID

Für ein DataGrid benötigt man eine beliebige Suche nach allen Feldern.

Gängige Herangehensweisen:

- a) Komplettes Suchformular selbst nachprogrammieren.
- b) Ein paar Felder mit eingeschränkten Suchmöglichkeiten bereitstellen.

→ Lösungsansatz von Drittanbietern: **Automatische integrierte Filter im DataGrid**

Aufgrund des Datentyps wird eine Excel-ähnliche Filtermöglichkeit dargestellt.



PROBLEM: INFORMATIONSPARME COMBOBOX

Eine ComboBox bietet zu wenige Informationen, um schnell den richtigen Datensatz finden zu können. z.B. beim Formular: Order - ComboBox: Employee (FirstName, LastName, Nr, ...)

Gängige Herangehensweisen:

- a) Standard-ComboBox belassen
- b) Mehrspaltige ComboBoxen mit Suchfeld selbst programmieren
- c) Wechsel zu einer Unteransicht-Datenauswahl

→ Lösungsansatz von Drittanbietern: **Lookup-ComboBox**

Eigenes, konfigurierbares Control mit einem integrierten DataGrid.



PROBLEM: FORMULAR LAYOUTING

Setze 20 Felder auf ein Detailformular sauber untereinander!

☹️
|Label 1 - |Textbox 1
| Label 2 |- Textbox 2
| Label 3 -|Textbox3

Gängige Herangehensweisen:

- a) Felder mit Visual Studio-Designer Einrastfunktion anordnen.
- b) Ein Grid Container mit zwei Spalten anlegen und jedes Control in eine Zelle setzen.

→ Lösungsansatz von Drittanbietern: **Form-Layout-Manager**

Labels und Controls müssen nur nacheinander hinzugefügt werden, und die Position wird automatisch gesetzt.



PROBLEM: ANWENDUNG FRIERT EIN

☹ *Langlaufender Code im UI-Thread blockiert die GUI. Die Anwendung friert ein.*

Gängige Herangehensweisen:

- a) Background-Worker
- b) Threads und SynchronizationContext
- c) async-await einsetzen

Was oft vergessen wird – Rückmeldung dazu auf der UI zu geben!

→ Daten werden geladen...

→ 65% erledigt

→ Lösungsansatz von Drittanbietern: **Progress-Indicator-Control**

Control überblendet alles, und zeigt animierte Rückmeldung an – die Anwendung bleibt flüssig.



PROBLEM: SCHÖNES, BESONDERES DESIGN

☹ *Entwickler können meistens selbst keine schönen Designs bauen.*

Gängige Herangehensweisen:

- a) Design? Egal!
- b) Selbst machen und daran scheitern
- c) Frameworks nutzen, die Design mitliefern

→ Lösungsansatz von Drittanbietern: **Theming**

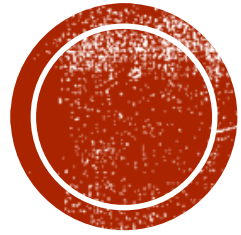
- **PROBLEM:** Nicht alle Hersteller schaffen ein 100% Theming!
LÖSUNG: Der Knackpunkt: Das Window – es muss durch ein eigenes Window ausgetauscht werden. Das bieten aber nicht alle an.
- **PROBLEM:** Nicht alle Hersteller bieten alle Controls an (z.B. Label, TreeView, usw.)!
LÖSUNG: WPF Standard Controls werden von allen Herstellern beim Theming ebenfalls gestyled.
- **PROBLEM:** Eine Reihe von Standard-Designs sind festgelegt, können aber nicht einfach individuell angepasst werden.
LÖSUNG: Custom Theming wird durch hauseigene Tools ermöglicht. Das geht aber nicht bei allen Herstellern.



PROBLEM: SCHÖNES, BESONDERES DESIGN

Hersteller	Window-Theming	Standard-Controls-Theming	Custom Theming
actipro	-	-	-
DevExpress	X	X	X (DevExpress WPF Theme Designer)
GrapeCity / ComponentOne	-	X	○ (Microsoft Expression Blend)
INFRAGISTICS	-	X	○ (Microsoft Expression Blend)
Nevron	-	X	-
Syncfusion	X	X	X (Syncfusion Theme Studio)
Progress / Telerik	○ (kein normales Hauptfenster - nur modale Fenster)	X	X (Telerik Color Theme Generator)
Xceed	-	?	-





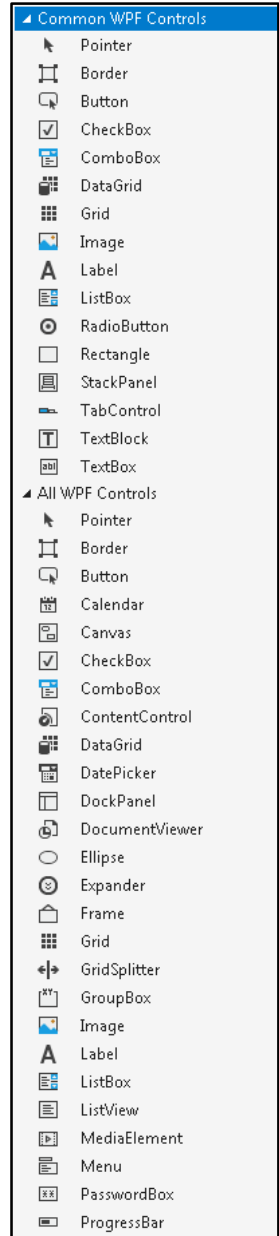
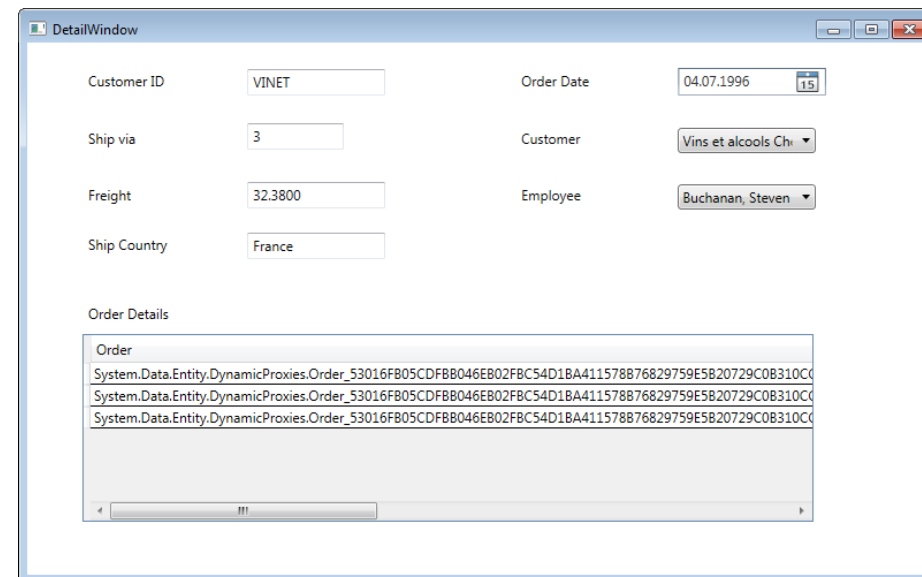
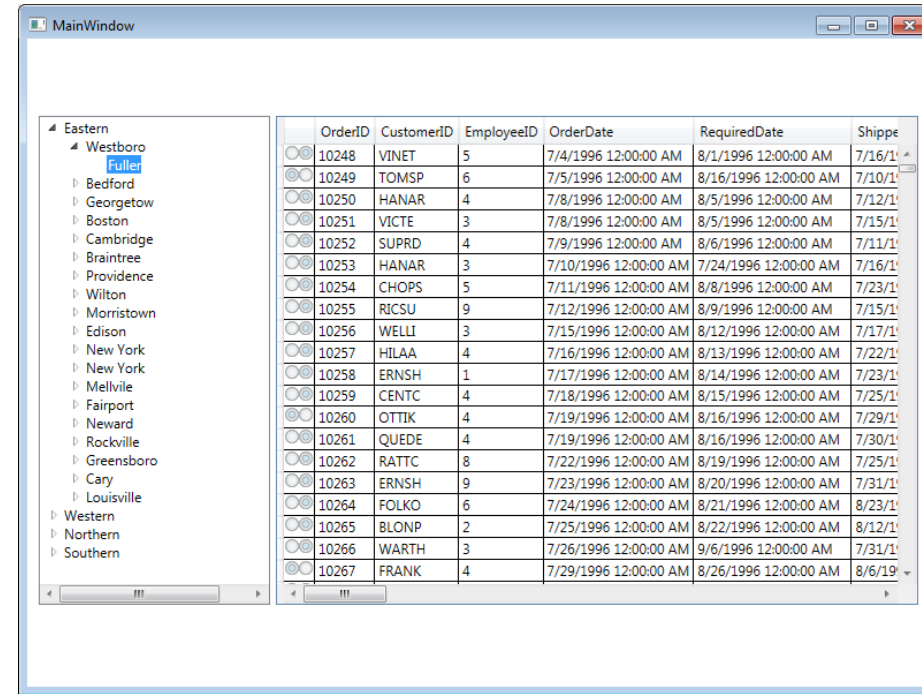
VERGLEICHSANWENDUNG NORTHWIND GUI

STANDARD-CONTROLS

Control-Vergleich mit der WPF
Demoanwendung: **NorthwindGUI**

Grundlegende Anforderungen:

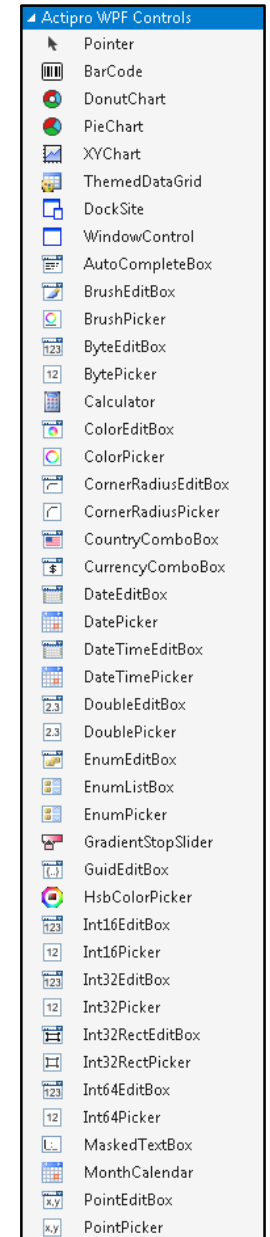
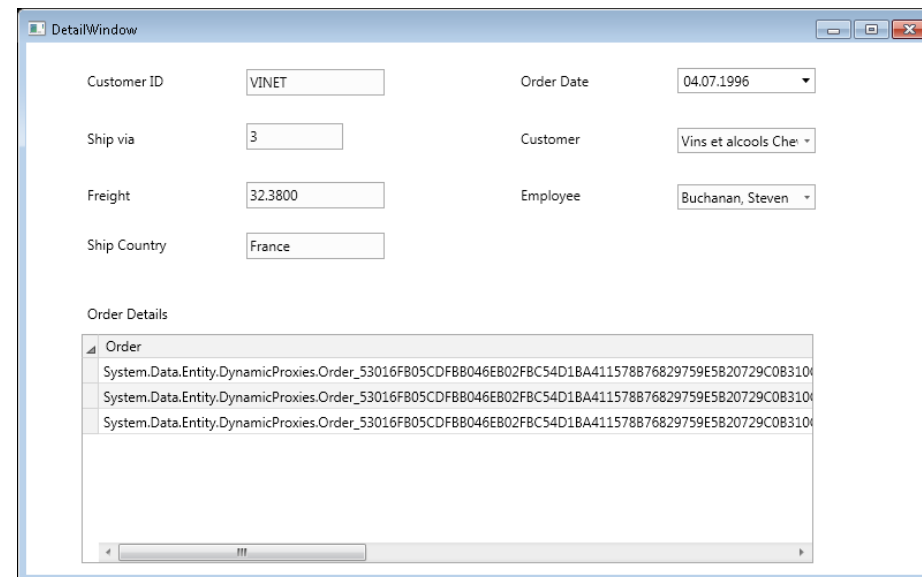
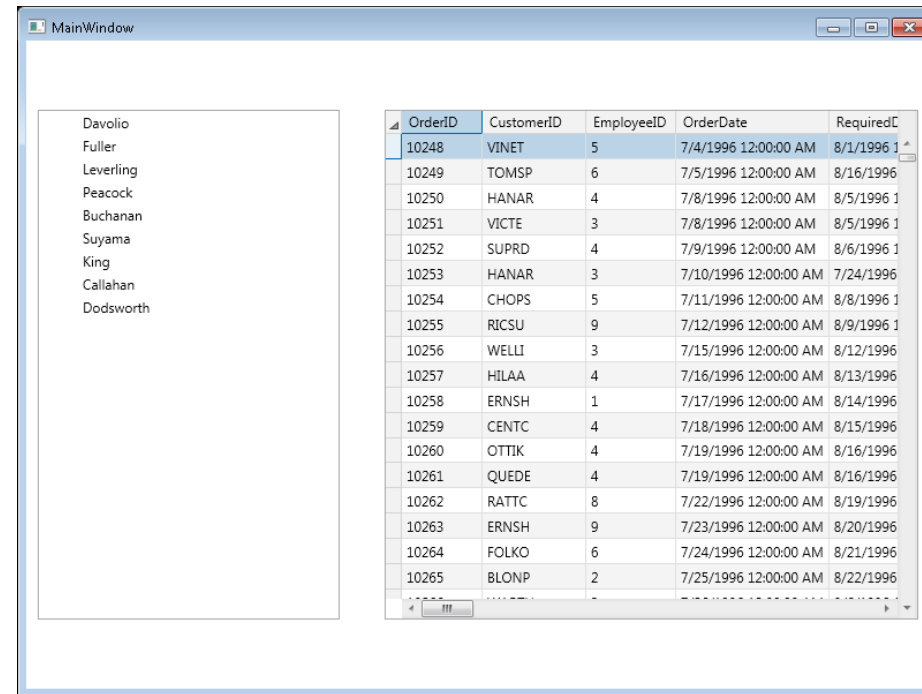
- Laden der Daten aus der Datenbank
- TreeView der Employees mit Filterung im DataGrid
- Anzeige der Orders im DataGrid (830 Datensätze) mit TemplateColumn
- Doppelklick auf eine Zeile öffnet die Detailansicht
- Detailansicht mit DatePicker, ComboBox, ComboBox mit Template und Untertabelle



ACTIPRO

www.actiprosoftware.com

- **Vorteile:**
 - Viele Controls
- **Nachteile:**
 - Teilweise zu wenig Dokumentation (ThemedDataGrid)
 - TreeView-Ersatz: ‚TreeListBox‘ nicht standardkonform (Fehlermeldung: *Do not set the ItemsSource property on this control. Set the RootItem property instead.*)
 - Es ist nicht ersichtlich, wie man mehrere Ebenen in die TreeListBox einbindet



DEVEXPRESS

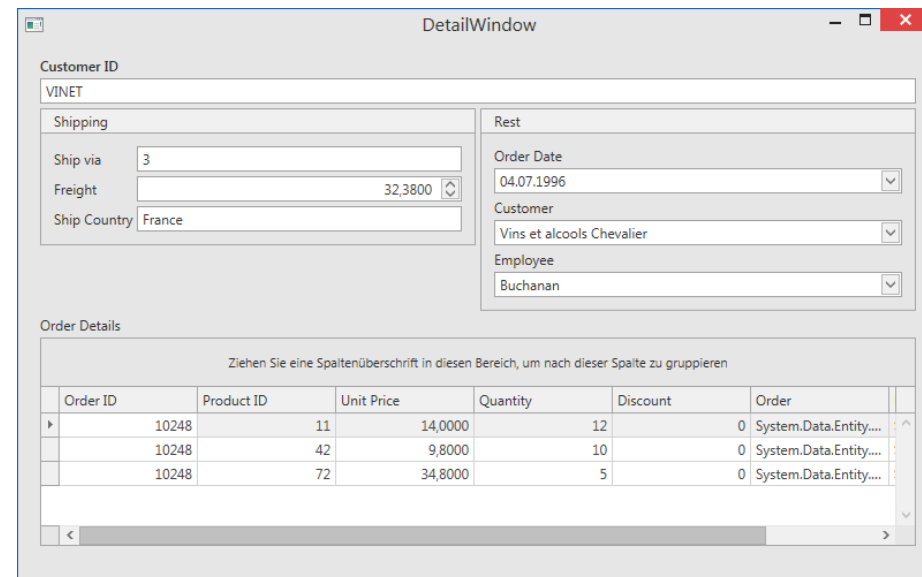
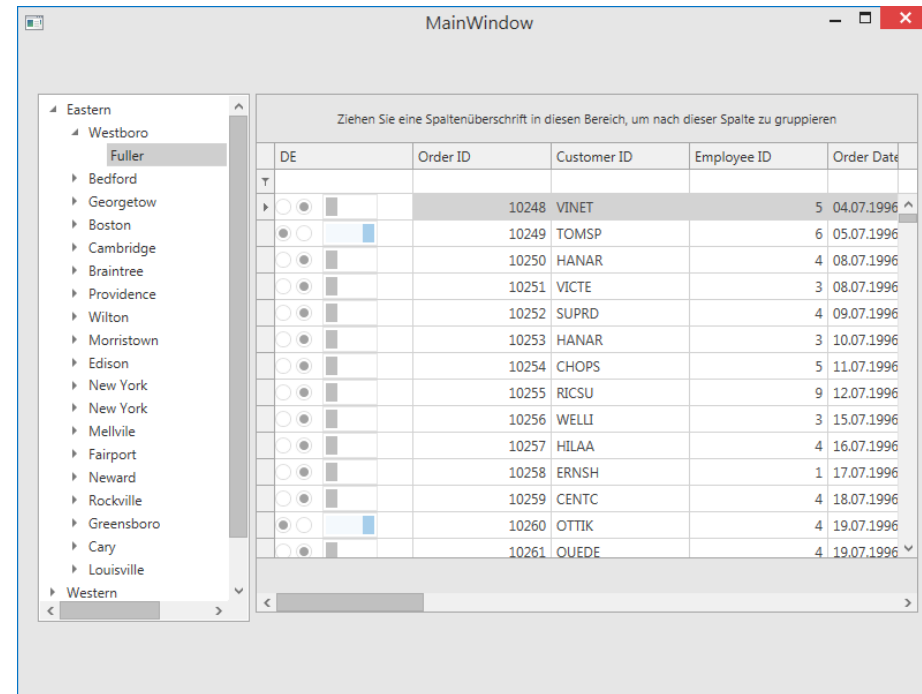
www.devexpress.com

■ Vorteile:

- Sehr gute Designer-SmartTags
- Standardkonformes Programmiermodell (z.B. DataGrid unterstützt CellTemplates)
- Ausführliche und beispielhafte Dokumentation
- Viele bestehende Lösungen im Support

■ Nachteile:

- Kein eigenes TreeView-Control (stattdessen Navigation Bar verwenden)
- Kein Ladeanzeige-Control
- Kein eingebautes Paging beim DataGrid



- DX.17.1: Data & Analytics
- DX.17.1: Reporting
- DX.17.1: Navigation & Layout
- DX.17.1: Common Controls**
 - Pointer
 - DateTimeChartRangeControl
 - NumericChartRangeControl
 - GalleryControl
 - DropDownButton
 - SimpleButton
 - SplitButton
 - BarcodeEdit
 - ButtonEdit
 - Calculator
 - CameraControl
 - CheckEdit
 - ColorEdit
 - ComboBoxEdit
 - DateEdit
 - DateNavigator
 - FlyoutControl
 - FontEdit
 - HyperlinkEdit
 - ImageEdit
 - ListBoxEdit
 - MemoEdit
 - PasswordBoxEdit
 - PopupCalcEdit
 - PopupColorEdit
 - PopupImageEdit
 - ProgressBarEdit
 - RangeControl
 - RatingEdit
 - SearchControl
 - SpinEdit
 - TextEdit
 - ToggleSwitch
 - TrackBarEdit
 - LookupEdit
- DX.17.1: Scheduling
- DX.17.1: Layout Control
- DX.17.1: Rich Text Editor
- DX.17.1: Spreadsheet



GRAPECITY / COMPONENTONE

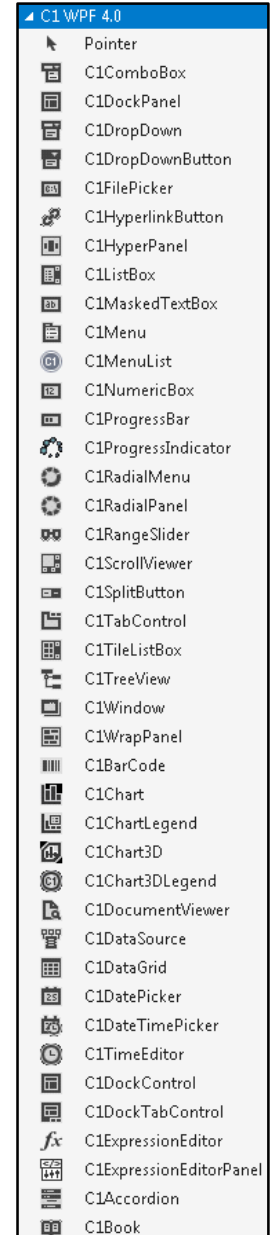
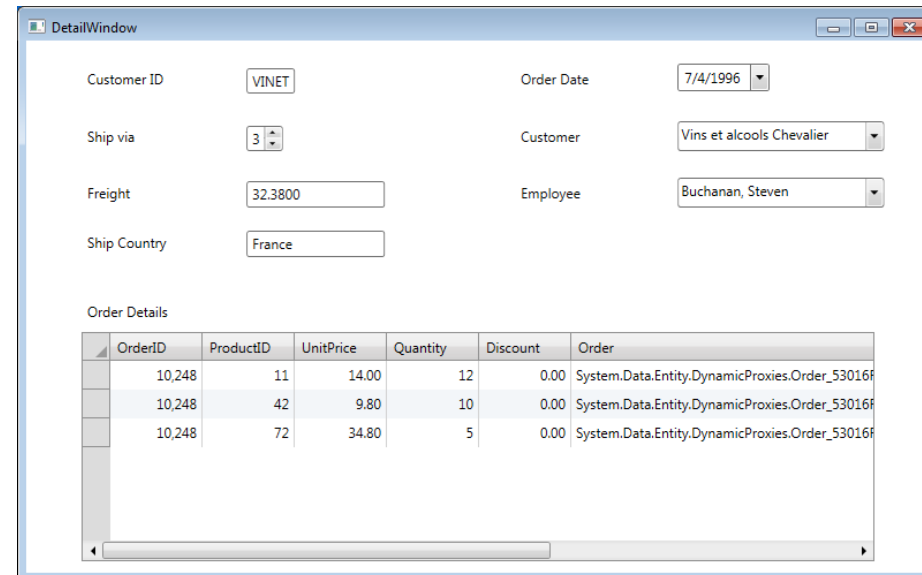
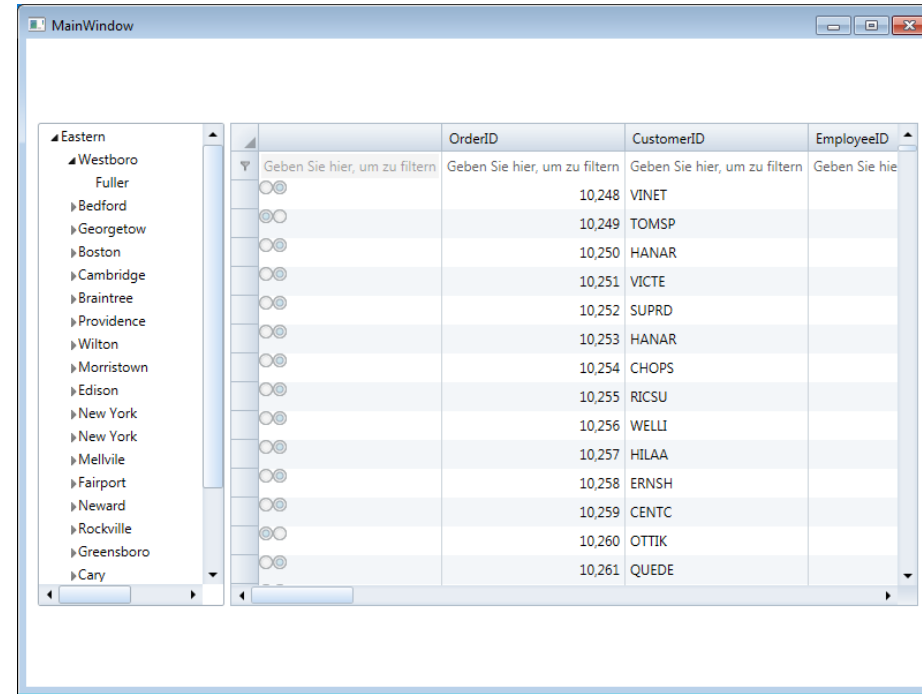
www.grapecity.com

■ Vorteile:

- Standardkonformes Programmiermodell (z.B. DataGrid unterstützt CellTemplates)
- Klare Benennung (C1ComboBox, C1NumericBox, C1DataGrid)
- Verstecktes AutoLookup-Feature (löst z.B. Employee selbstständig im C1DataGrid auf)

■ Nachteile:

- Keine Designer-SmartTags
- Fehlende Antworten im Forum
- Manchmal zu knappe Dokumentation – kann aber oft durch Sample-Codes ausgeglichen werden
- Teilweise fehlerhafte Vermischung von WPF und Silverlight in der Dokumentation (Beispiel: Global Theming)
- Ungelöster, aber bekannter ComboBox-Bug (trifft nur zu, falls ItemTemplate gesetzt ist): Während der Auswahl des gebundenen Objekts wird nur die ToString-Ausgabe angezeigt



INFRAGISTICS

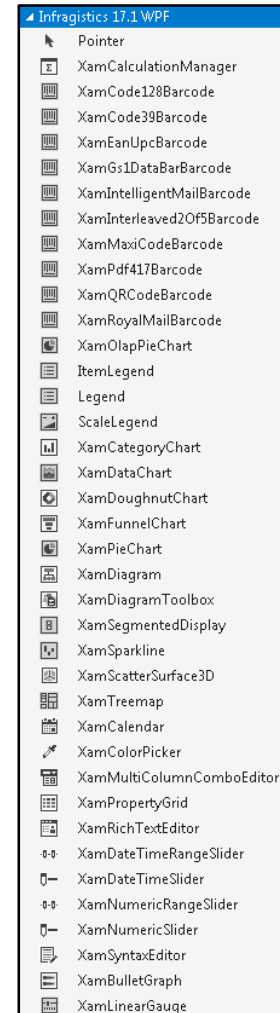
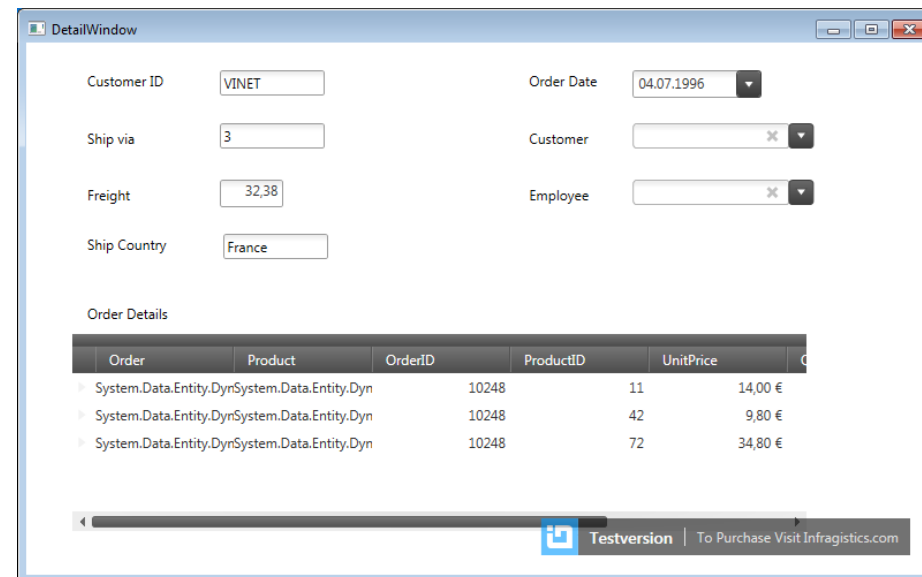
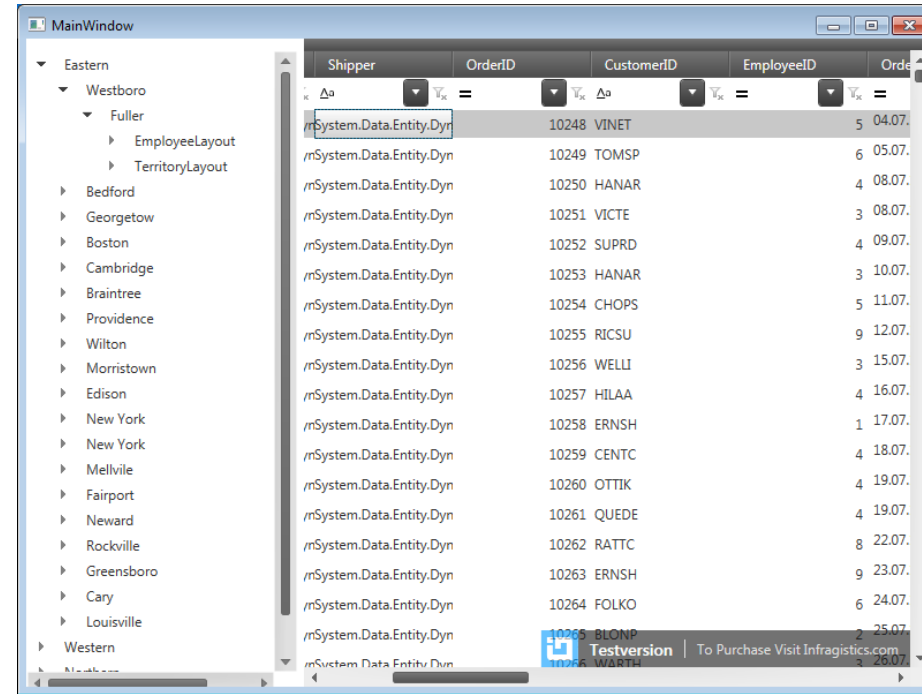
www.infragistics.com

■ Vorteile:

- Sehr viele Controls
- Teilweise umfangreiche SmartTag-Designer
- INFRAGISTICS-Objektmodell in sich konsistent (z.B. Value statt Text)

■ Nachteile:

- Keine DataBinding Designer-Dialoge
- Dokumentation zu minimalistisch
- Erste Verwirrung durch zwei DataGrids (xamGrid, xamDataGrid)
- INFRAGISTICS-Objektmodell nicht standardkonform (z.B. DataSource statt ItemsSource, Value statt Text)
- Beispiel xamDataGrid:
 - Keine Standards bei Benennung (Columns – Fields)
 - Kein CellTemplate möglich (wird ignoriert)
 - Scrollbalken scrollt keine Daten, sondern zeigt nur ein Tooltip an



NEVRON

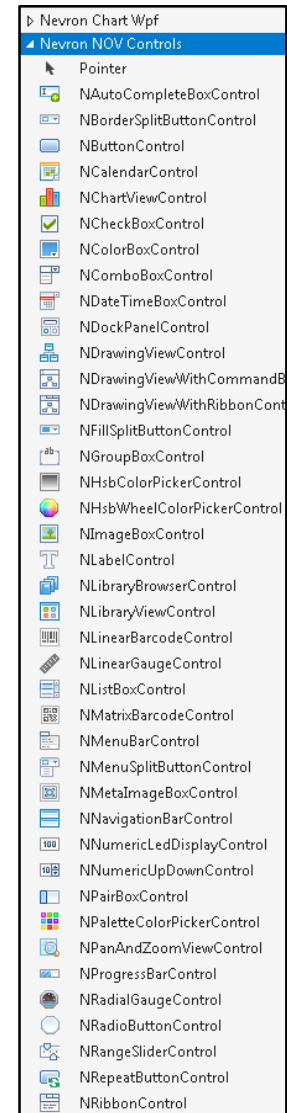
www.nevron.com

- **Vorteile:**
 - Einzigartiger, interessanter Cross-Platform Ansatz (leider nur per Code ohne Designerunterstützung)

- **Hinweise zum fehlerhaften Test:**
 1. Fehler: MessageBox-Hinweis, damit neue Referenz hinzugefügt wurde. Trotz Referenz kommt der Dialog immer wieder.
Oder: Ein ItemsSource-Property ist nicht verfügbar!
 2. Bug an Nevron gemeldet
 3. Erkenntnis von Nevron:
Dies ist ein Bug in ihren Tools, der seit Visual Studio 2015 auftritt – an der Behebung wird gearbeitet
 4. Empfehlung von Nevron:
Vorerst Code-only Cross-Platform Ansatz nutzen!



*Noch nicht getestet –
Technische Probleme
im Designer in
Visual Studio 2017*



SYNCFUSION

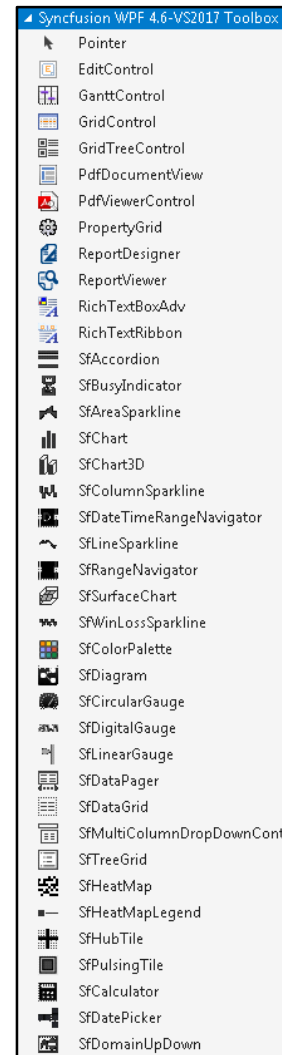
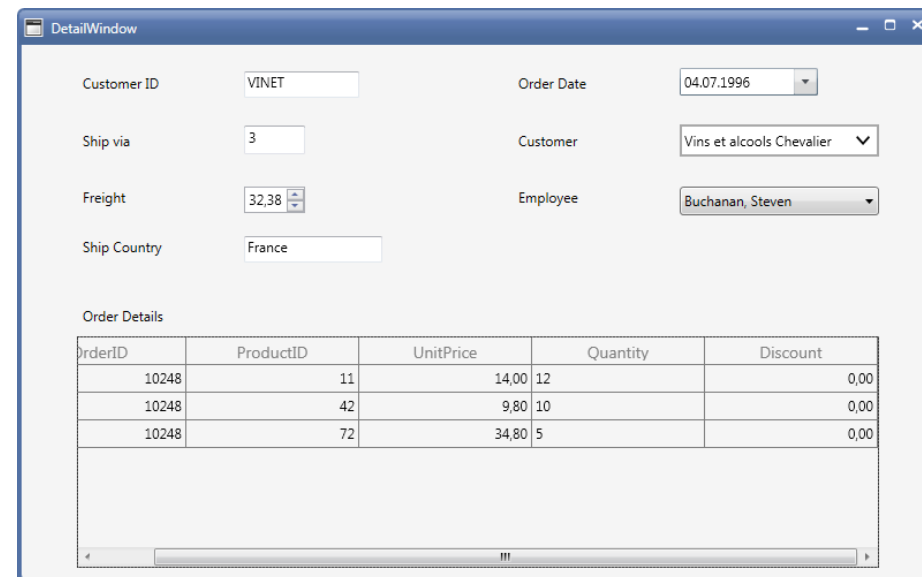
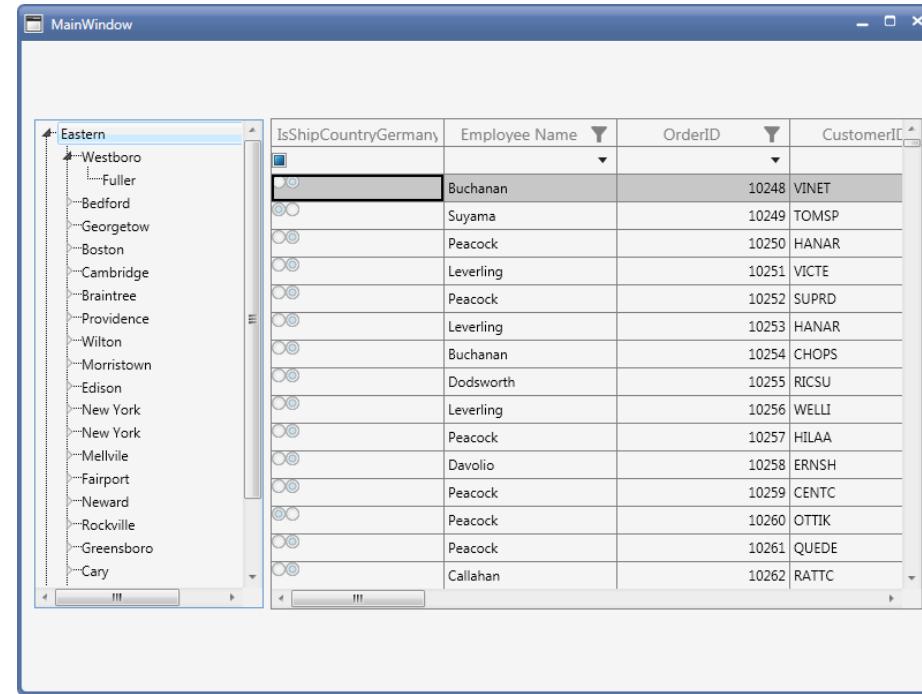
www.syncfusion.com

■ Vorteile:

- Grundlegende Designer-SmartTags
- Standardkonformes Programmiermodell (z.B. DataGrid unterstützt CellTemplates)
- Umfangreiche Dokumentation

■ Nachteile:

- Inkonsistente Control-Benennung (TreeViewAdv, SfDataGrid, GridControl, SfTextBoxExt)
- Schwierige Control-Auswahl (TextBox vs. SfTextBoxExt, UpDown vs. IntegerTextBox)
- Keine Lösung oder einfache Beschreibung, um bestimmtes globales Theme festzulegen



TELERIK

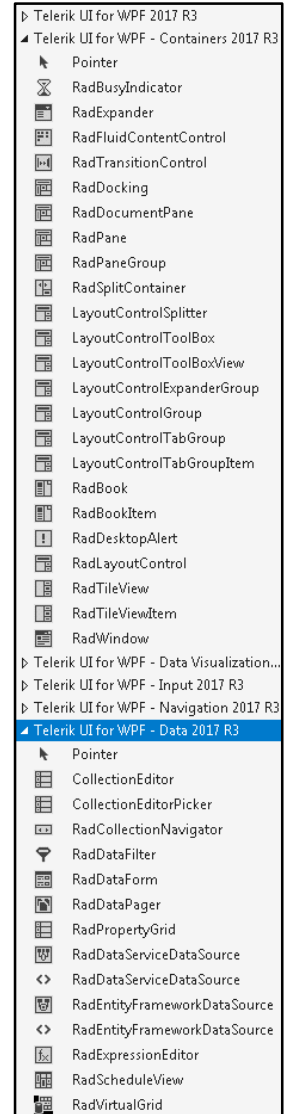
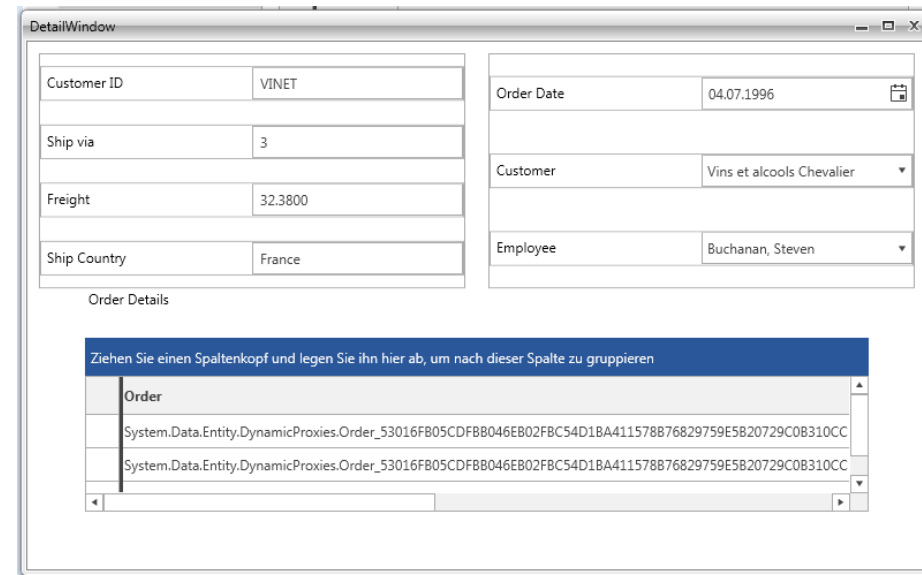
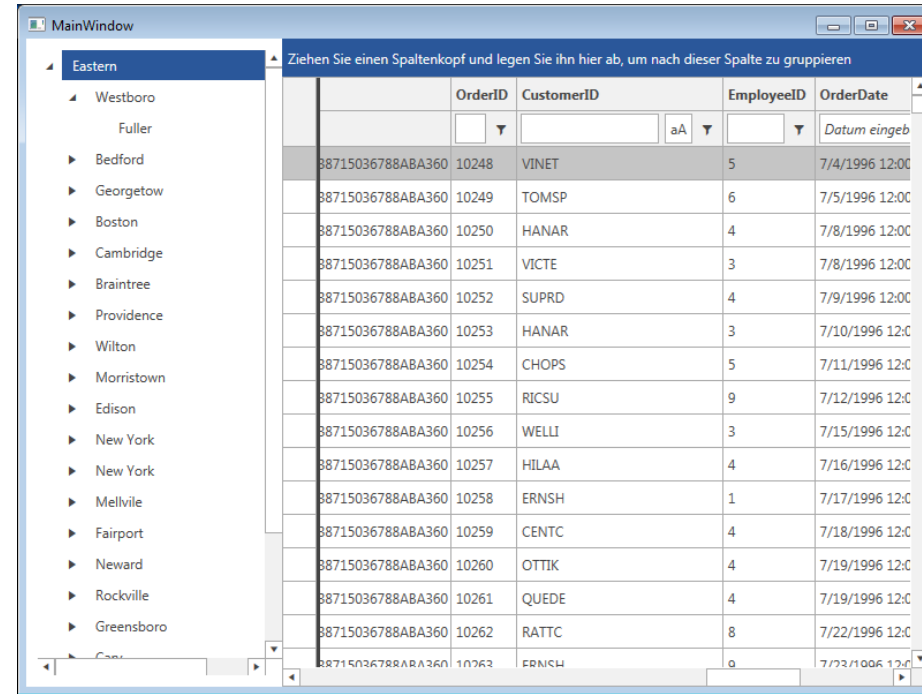
www.telerik.com

■ Vorteile:

- Ziemlich standardkonformes Programmiermodell
- Außergewöhnlicheres Design
- Standardmäßige Animationen enthalten

■ Nachteile:

- SmartTags ohne Inhalt (nur Onlinelinks)
- Inzwischen aufgekauft von Progress – Subjektiver Eindruck: Verschlechterung bei Support & User Group Unterstützung
- Starker Onlinezwang (Installation, Beispielanwendung, SmartTags)
- WPF Projekt-Vorlage ist fehlerhaft (z.B. kein aktiver Build für Solution, Designer-Namespace fehlt, u.a.)
- Style für einfaches Window nicht möglich (RadWindow ist nur für modale Fenster!)



XCEED

www.xceed.com

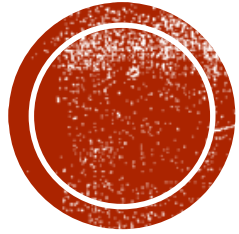
- **Vorteile:**
 - Besonderer Fokus auf leistungsstarkes und umfangreiches DataGrid

- **Hinweise zum fehlerhaften Test:**
 1. Die Toolbox in Visual Studio 2017 Enterprise enthält keine Xceed Controls
 2. Bug an Xceed gemeldet
 3. Erkenntnis von Xceed:
Dieser Bug tritt nur in VS 2017 Enterprise auf
- in VS 2017 Professional funktioniert es.
 4. Empfehlung von Xceed:
Vorerst Toolbox-Elemente über Customize
Toolbox selber hinzufügen.



*Noch nicht getestet –
Designer-Toolbox in
Visual Studio 2017
Enterprise ist leer*





FAZIT



FAZIT

UNTER WPF

Rang	Hersteller	Leistungsumfang	Dokumentation / Beispiele / Support	Visual Studio Designer Integration / Rapid Application Development	Standardkonformität	Style / Theming	Punkte	
1	DevExpress	★ 4	★ 4	★ 4	★ 3	★ 3	18	
2	Syncfusion	★ 4	★ 4	★ 2	★ 2	★ 3	15	
3	Progress / Telerik	★ 4	★ 2	★ 2	★ 3	★ 3	14	
4	Infragistics	★ 3	★ 2	★ 3	-	★ 2	10	
4	GrapeCity / ComponentOne	★ 3	★ 2	★ 1	★ 2	★ 2	10	
5	Actipro	★ 2	★ 1	★ 1	★ 2	★ 1	7	
6	Nevron	★ 3	-	-	-	★ 1	4	
-	Xceed	-	-	-	-	-	0	

Hinweis:

Diese Bewertung ist abhängig von den oben genannten Entscheidungskriterien und den subjektiven Erfahrungen des Referenten. Das bedeutet, dass andere Kriterien und andere Gewichtungen zu einer andere Rangfolge führen. Wenn z.B. der Visual Studio Designer nicht wichtig ist, oder ein spezielles einzigartiges Control benötigt wird, dann wird die Tabelle neu gemischt, und die unteren Hersteller sind dann auf den Top-Rängen.

