Access NetView

http://accessnetview.codeplex.com

Hintergrund

Daten jenseits von Text

Microsoft Access ist ein wunderbares Werkzeug um Daten aus einer relationalen Datenbank einfach und schnell an die Benutzeroberfläche zu bekommen. Obwohl viele Möglichkeiten zur Verfügung stehen um die Darstellung und Interaktion mit den Daten zu beeinflussen, sind es dennoch immer textuelle Repräsentationen der Daten.

Textuelle Darstellungen in der Form einzelner Texte oder in Listenform können bei komplexeren Zusammenhängen oder größeren Datenmengen leicht unübersichtlich werden. Hier möchte Access NetView einspringen. Dem Entwickler einer Endbenutzeranwendung in Microsoft Access soll ein Steuerelement zur Hand gegeben werden mit dem Daten optisch aussagekräftiger dargestellt und auch manipuliert werden können.

Features

- **Databinding**
  - pro Datensatz ein Item, repräsentiert von einem Label
  - Viele Eigenschaften können an Daten gebunden werden (Texte, Farben, Position und Größe)
- **Interaktion** mit der **Maus** (Positionierung, Größenänderung) – Drag and Drop
- **Interaktion** mit der **Tastatur** (ToDo)
- Maßangaben in **beliebigen Einheiten** („Weltkoordinaten“). Die Umsetzung auf das Formular übernimmt das Steuerelement.
- Freie Wahl der **Orientierung** mittels Eigenschaft – alle acht denkbaren Orientierungen werden unterstützt (tlw. ToDo)
- **Einfache Einsetzbarkeit** – Steuerung über ein einfaches Programmiermodell, angelehnt an die Access-Nomenklatur
- **Ereignisse** – auf Ebene des Controls und jedes Items
- Mischung von **datengebundenen** und **nicht datengebundenen** Items
- **Erweiterbares Design**
  - Umwandlung von Werten aus der Datenbank in Weltkoordinaten (*ValueConverter*)
  - Einschränkungen bei Positions- und Größenänderungen (Raster, erlaubte/unerlaubte Bereiche, ...) und automatische/halbautomatische Positionierung und Größenbestimmung (*PlacementPolicy*)
  - Definition mit welchen Bereichen eines Items Manipulationen mit der Maus durchgeführt werden können (wo wird verschoben, die Höhe, die Breite verändert? – *ManipulationKindProvider*)
  - Ein Item kann aus mehreren Access-Steuerelementen bestehen (*ItemLayoutManager* – ToDo)

Was man damit machen kann

Im Folgenden finden sich als Anregung einige mögliche Anwendungsfälle:

- **Pläne aller Art**
  - Kalender und Stundenpläne
    - einfach (z.B. 1., 2., 3., ... Stunde)
    - mit genauen Zeiten (z.B. 7^{45}-8^{35}, 8^{40}-9^{30}, 10^{40}-...)
  - Raum- und Sitzpläne (Klasse, Konferenzraum, Theater, ...)
  - Ressourcen- und Auslastungspläne (Mitarbeiter, Räume, Fahrzeuge, Maschinen, ...)
- Alternative **Listendarstellung**
  - Dynamische Höhen
- Alternative **Darstellung von Hierarchien**
  - Baum
  - Stern
  - TreeMap
- Darstellung und Interaktion mit **vielfältig verknüpften Daten**
  - Abhängigkeiten von Objekten (z.B. Formulare -> Abfragen -> Tabellen) in einer Datenbank
  - Navigation durch eine Anwendung
  - Navigation durch einen Datenbestand
- Einfacher **Berichtseditor**
  - Mit Zoom
  - MDE-fähig
- Windows 8 **Kachel-Design**

Access NetView in eigenen Projekten einsetzen

Anforderungen

Um Access NetView einsetzen zu können wird Access 2003 oder höher benötigt.

Das Datenbankformat ist Access 2002/2003. Tests mit Access 2002 stehen allerdings noch aus. Da keine besonderen Features von Access 2003 benutzt werden sollte auch eine Rückkonvertierung nach Access 2000 möglich sein.

Vorgangsweise

Hinweis: Für die jeweils aktuelle Variante der Anleitung bitte die online-Dokumentation (DOCUMENTATIONS-Tab auf http://accessnetview.codeplex.com) konsultieren.

1. Die letzte Release von Access NetView herunterladen
   a. DOWNLOADS-Tab auf http://accessnetview.codeplex.com
2. Das eigene Projekt vorbereiten
   a. Die im Download enthaltene MDB-Datei in das selbe Verzeichnis wie die eigene eigene Datenbank kopieren.
   b. Im VisualBasic-Editor einen Verweis auf NetView setzen.
   c. Das Formular NetViewControl aus NetView in das eigene Projekt importieren.
3. Das Steuerelement in einem Formular verwenden

a. Das vorhin importierte Formular NetViewControl in das Formular als Unterformular einbetten.
b. Im Form_Open-Ereignis die erforderlichen Eigenschaften des Steuerelements setzen.

Beispiel:

```
Private m_NetView As NV_NetView

...

Private Sub Form_Open(Cancel As Integer)
   Set m_NetView = NetViewControl.Form.TheNetView

   With m_NetView
      ' Kosmetik
      .BackColor = RGB(255, 255, 153)   ' helles Gelb
      .BorderColor = vbBlue

      ' Die Größe der Welt
      .Minimum1 = -100
      .Maximum1 = 100
      .Minimum2 = 1
      .Maximum2 = 11

      ' Databinding aufsetzen
      .RecordSource = "SELECT * FROM MyQuery"

      .BindItemBusinessIdTo "Id" ' Erforderlich
      .BindItemTextTo "MyInfo"
      .BindItemTextColorTo "ColorA"
      .BindItemTipTextTo "MyTipText"
      .BindItemBackColorTo "ColorB"
      .BindItemBorderColorTo "ColorC"
      .BindItemPosition1To "Pos1"
      .BindItemExtent1To "Ext1"
      .BindItemPosition2To "Pos2"
      .BindItemExtent2To "Ext2"
   End With
End Sub
```

Konzepte, Begriffe, Anpassung

Multidimensionale Daten

Mit dem Access NetView können Daten(sätze) dargestellt werden, deren Zustand sich aus mehreren Attributen zusammensetzt (siehe Aufstellung unten). Im Gegensatz zu einer in Access eingebauten ListBox kann nicht nur ein einziges Attribut[1] (ListBox: ein Text) dargestellt werden. In Access NetView stehen viele Eigenschaften zur Verfügung die zur Darstellung von Information verwendet werden können.

[1] Genau genommen stellt auch die Reihenfolge der Darstellung Information dar.

Beispielsweise könnte die Hintergrundfarbe eines Items an den Status einer Bestellung gebunden werden.

Die folgenden drei Gruppen von Eigenschaften gibt es:

- Texte
- Position und Größe in zwei Dimensionen
- Grafische Attribute

Aktuell stehen zehn Eigenschaften zur Verfügung, zehn weitere sind geplant. Die Werte dieser Eigenschaften können per Databinding aus den Daten gelesen werden.

Items

Ein Item wird von einem Label im NetView repräsentiert. Ein Item kann ad hoc erzeugt werden (AddItem() Methode) oder von Databinding erstellt (und ggf. wieder gelöscht) werden.

Eigenschaften

Folgende Eigenschaften werden aktuell von einem Item angeboten:

Eigenschaft	Beschreibung
BusinessId	Der Schlüssel des Items (erforderlich für Databinding)
Text	Der im Item angezeigte Text
TipText	Der Text des Tooltips
Position1	Position entlang der Achse 1
Extent1	Ausdehnung entlang der Achse 1
Position2	Position entlang der Achse 2
Extent2	Ausdehnung entlang der Achse 2
BackColor	Hintergrundfarbe
BorderColor	Rahmenfarbe
TextColor	Textfarbe
Readonly	True: Item kann manipuliert werden False: Item kann nicht manipuliert werden. Events werden aber weiterhin gefeuert.
Visible	True: Der Item wird angezeigt (falls der entsprechende Ausschnitt aus der Welt im darstellbaren Bereich ist). False: Der Item wird nicht angezeigt.

Diese Symbol kennzeichnet die datengebundenen Eigenschaften.

Geplante Eigenschaften

Folgende Eigenschaften sind noch geplant:

Eigenschaft	Beschreibung
BorderWidth	Die Breite der Rahmenlinie
BorderStyle	Die Linienart für den Rahmen
FontName	Die zur Textdarstellung verwendete Schriftart
FontSize	Die Größe der Schriftart
FontBold	Gibt an ob der Text fett dargestellt wird
FontItalic	Gibt an ob der Text kursiv dargestellt wird
FontUnderline	Gibt an ob der Text unterstrichen wird
PaddingTop*	Innenabstand des Texts vom oberen Rahmen des Labels
PaddingRight*	Innenabstand des Texts vom rechten Rahmen des Labels
PaddingBottom*	Innenabstand des Texts vom unteren Rahmen des Labels
PaddingLeft*	Innenabstand des Texts vom linken Rahmen des Labels

* Diese Eigenschaften sind mehrfach fraglich: Ist es sinnvoll sie überhaupt anzubieten? Ist es sinnvoll sie datengebunden zu machen? Nicht zuletzt ist die hier intendierte Bedeutung in Access mit MarginXxx verbunden. Das steht aber im Gegensatz zu der Nomenklatur wie sie im Web (HTML und CSS verwendet) wird.

Databinding

Access NetView unterstützt Databinding. Nach Angabe der Datenquelle (Eigenschaft RecordSource) und der Felder an die Eigenschaften der Items gebunden werden (Eigenschaften BindXxxTo), übernimmt das Steuerelement das Erzeugen, Aktualisieren und Löschen der Items (Methode Requery).

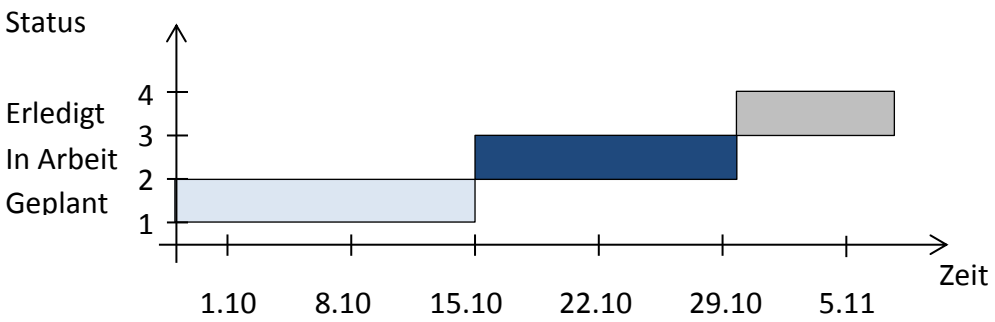
Die Verbindung zwischen den Datensätzen in der Datenquelle und den Items im Steuerelement erfolgt anhand der Eigenschaft BusinessId. Bei Databinding ist daher die Angabe eines Feldes für die BusinessId erforderlich.

Eigenschaften die nicht für Databinding konfiguriert sind, werden im Zuge des Aktualiserens aus der Datenquelle nicht verändert.

World-/Device-Koordinaten

Alle Positions- und Größenangaben erfolgen in einem frei definierbarem zweidimensionalen Weltkoordinatensystem. Dieses Weltkoordinatensystem verwendet für die darzustellenden Daten sinnvolle Wertebereiche. Wenn beispielsweise in den Daten Datumsangaben vorhanden sind, kann eine Achse des Weltkoordinatensystems die Zeit in Tagen repräsentieren. Die andere Achse

könnte hingegen den Status eines Datensatzes in Form einer fortlaufenden ganzen Zahl aufnehmen. Die Skalen der beiden Achsen können unabhängig von einander definiert werden:



Wenn die Daten in der Datenquelle noch nicht in sinnvoller Form vorliegen, kann eventuell ein ValueConverter eingesetzt werden um die Daten aus der Datenquelle in eine andere Form zu übersetzen.

Das Weltkoordinatensystem wird von zwei Achsen aufgespannt (Achse 1 und Achse 2). Alle achsenbezogenen Angaben werden entsprechend mit 1 oder 2 gekennzeichnet. Die Position des Items auf jeder dieser Achsen wird durch eine Position (Position1 bzw. Position2) und eine Ausdehung (Extent1 bzw. Extent2) definiert.

Die tatsächlichen Koordinaten der Items im Formular werden vom Steuerelement berechnet.

Device/World-Mapping

Das Device/World-Mapping ist verantwortlich für die Umwandlung von Positions- und Größenangaben aus dem Weltkoordinatensystem in das Gerätekoordinatensystem des Formulars. Mithilfe des Device/World-Mappings können beliebige Bereiche des Weltkoordinatensystems in ihre Repräsentationen im Control umgerechnet werden (und umgekehrt).

Definiert wird das Device/World-Mapping durch das WorldDefinitionArea, die Position und Größe des Steuerelements und die gewünschte Orientierung der Darstellung. Diese Parameter finden sich als Eigenschaften des Steuerelements wieder.

WorldDefinitionArea

Das WorldDefinitionArea ist ein Rechteck das jenen Ausschnitt aus dem Weltkoordinatensystem definiert der genau auf den am Formular sichtbaren Bereich des Steuerelements abgebildet weren soll.

Position und Größe des Steuerelements im Formular

Die Position und die Größe des Steuerelements im Formular stellt auch das Pendant zum WorldDefinitionArea auf Seiten des Devicekoordinatensystems dar. Das Mapping stellt sicher, dass diese Fläche entsprechend der einstellten Orientierung auf das WorldDefinitionArea abgebildet wird.

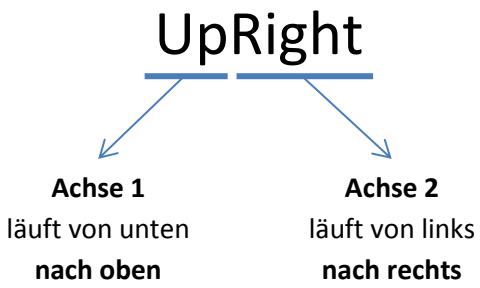
Orientation

Die Abbildung der zwei gerichteten Achsen des Weltkoordinatensystems auf die zwei anderen ebenfalls gerichtete Achsen des Gerätekoordinatensystems ergibt insgesamt acht mögliche

Kombinationen. Welche der Kombinationen vom Steuerelement angewendet wird bestimmt die Eigenschaft Orientation.

Orientierungsnamen

Der Name einer Orientierung ist ein zusammengesetztes Wort. Es gibt an in welche Richtung im Formular sich die beiden Achsen erstrecken:



Die gewohnte Orientierung im Formular wäre damit durch die Orientierung RightDown gegeben: die Achse 1 läuft von links nach rechts (Laufrichtung der x-Achse) und die Achse 2 von oben nach unten (Laufrichtung der y-Achse).

Im Anhang findet sich eine Aufstellung aller acht Orientierungen.

ManipulationKindProvider

Der ManipulationKindProvider bestimmt in welchem Bereich eines Items welche Manipulationsmöglichkeiten bestehen. Konkret liefert der ManipulationKindProvider für jede Geräteachse (horizontal und vertikal) die Art der Manipulation:

ManipulationKind	Bedeutung
None	Keine Manipulation
Move	Bewegung. Die Ausdehnung bleibt unverändert.
SizeAtBegin	Bewegen der Kante am Beginn (links bzw. oben). Die Kante am Ende (rechts bzw. unten) bleibt unverändert.
SizeAtEnd	Bewegen der Kante am Ende (rechts bzw. unten). Die Kante am Beginn (links bzw. oben) bleibt unverändert.
SymmetricSizeAtBegin	Symmetrisches Bewegen beider Kanten sodass die Mitte in dieser Dimension gewahrt bleibt. Die Kante am Beginn (links bzw. oben) folgt der Maus. Die Kante am Ende (rechts bzw. unten) bewegt sich in die entgegengesetzte Richtung).
SymmetricSizeAtEnd	Symmetrisches Bewegen beider Kanten sodass die Mitte in dieser Dimension gewahrt bleibt. Die Kante am Ende (rechts bzw. unten) folgt der Maus. Die Kante am Beginn (links bzw. oben) bewegt sich in die entgegengesetzte Richtung).

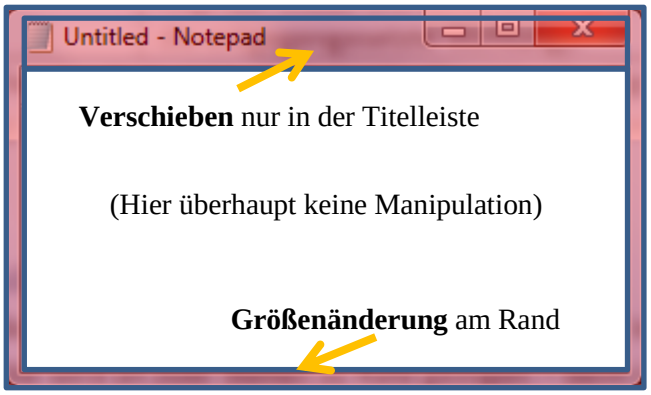
Standard-ManipulationKindProvider

Jeder neue Item hat den DefaultManipulationKindProvider konfiguriert. Hier lässt sich am Rand in die jeweilige Richtung die Größe verändern und in der Mitte der gesamte Item verschieben.

Eigene Provider

Eigene ManipulationKindProvider sind dann erforderlich, wenn das Verhalten des DefaultManipulationProviders nicht gewünscht ist.

Zum Beispiel könnte gewünscht werden, dass ein Item wie ein Fenster in Windows manipuliert werden kann:



Einen eigenen ManipulationKindProvider erstellen und einsetzen

1. Eine neue Klasse anlegen
2. Die Schnittstelle NV_IManipulationKindProvider implementieren
3. Eine Ereignisprozedur für das ItemAdded-Ereignis erstellen
4. Dort die Eigenschaft ManipulationKindProvider jedes Items auf eine neue Instanz des Providers setzen.

PlacementPolicy

Die PlacementPolicy kann beeinflussen, an welcher Position und in welcher Größe ein Item dargestellt wird. Sie wird an zwei Stellen herangezogen:

- beim Lesen der Daten für die Items aus einer Datenquelle und
- beim Interagieren mit der Maus.

Die Policy erhält jeweils das unmittelbar nach der Datenbindung oder Interaktion gültige Rechteck des Items (in Weltkoordinaten) und kann es ganz oder teilweise verändern.

Die PlacementPolicy ist die wichtigste und mächtigste Erweiterungsmöglichkeit für das Verhalten des Steuerelements.

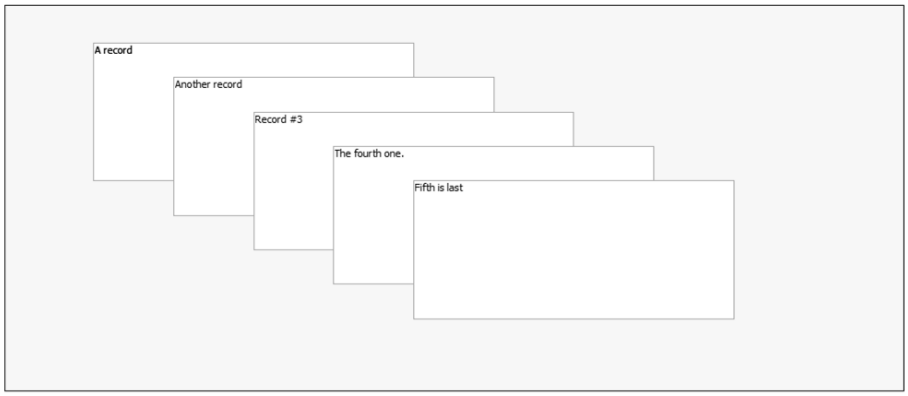
Mit einer PlacementPolicy können im Wesentlichen zwei Verhaltensänderungen erreicht werden:

- Es können **Einschränkungen** auf Position und/oder Größe der Items durchgesetzt werden. Es kann damit erzwungen werden, dass Items nur in einer bestimmten Rasterung erscheinen können, oder dass ihre Größe nur bestimmte vordefinierte Größen annehmen kann.
    - *Beispiel 1: Bei einem Kalender sollen Termine nur im Viertelstunden-Raster erstellt werden können. Die Endezeit könnte auf 20:00 beschränkt werden.*
    - *Beispiel 2: Bei einer Zimmerverwaltung können Buchungen nur vollständig von einem Zimmer weg auf ein anderes Zimmer verschoben werden. Eine Buchung darf nicht zwischen zwei Zimmern liegen.*

- Das **Layout** (die Anordnung der Items) kann ganz oder teilweise von der Policy übernommen werden. Wenn aus den Daten selbst keine sinnvollen Positionierungsangaben abgeleitet werden können, kann die Anordnung der Items von der Policy übernommen werden. Dabei könnte aber bspw. die Höhe des Items dennoch aus den Daten kommen.
    - *Beispiel 1: Eine Liste von Namen soll sortiert dargestellt werden. Die Policy übernimmt komplett die Steuerung von Positionen und Größen der Items.*
    - *Beispiel 2: In der Datenbank existiert eine Tabelle von Aufgaben. Für jede Aufgabe wurden bereits der Aufwand geschätzt. Es soll eine Liste der Aufgaben erstellt werden in der die Höhe jeder Aufgabe dem geschätzten Aufwand entspricht. Die Policy würde hier nur die Breite und die Positionierung bestimmen. Die Höhe kommt per Databinding aus den Daten.*
    - *Beispiel 3: In der Datenbank sind Datensätze hierarchisch miteinander verknüpft. Die Policy berechnet auf Basis der Daten ein Layout und weist jedem Item seinen entsprechenden Platz zu.*
    - *Beispiel 4: Für einen Seminarraum soll eine Bestuhlung definiert werden. Dazu sind aber jene Bereiche des Raums zu berücksichtigen in den kein Stuhl stehen kann (z.B. Säulen). Die Policy kann hier verhindern dass ein Stuhl an den Platz einer Säule gestellt wird, oder dass mehrere Stühle auf den selben Platz geschoben werden.*

Standard-Policy

Standardmäßig verwendet das NetView-Steuerelement die DefaultPlacementPolicy. Damit werden Items (sofern sie ihre Positions- und Größenangaben nicht aus den Daten erhalten) so angeordnet, dass sie einen Stapel darstellen und jeder Item sichtbar bleibt:



Die DefaultPlacementPolicy hat die Eigenschaft StackSize mit der die Anzahl der Items in einem Stapel definiert wird. Wenn eine größere Anzahl von Items platziert werden muss, wird ein neuer Stapel begonnen.

Eigene Policies

Eigene Policies können selbst erstellt werden um für die speziellen Anforderungen der Software entsprechende Einschränkungen und Layoutmöglichkeiten zu implementieren.

Weiters können PlacementPolicies für allgemeine Aufgaben auch zwischen Projekten getauscht werden. Solche allgemein verwendbare Policies bitte auf http://accessnetview.codeplex.com bekanntgeben. Sie können dann in das Projekt aufgenommen werden.

Eine eigene PlacementPolicy erstellen und einsetzen

1. Eine neue Klasse anlegen
2. Die Schnittstelle NV_IPlacementPolicy implementieren
3. Beim Initialisieren des NetView-Steuerelements der Eigenschaft PlacementPolicy eine Instanz der selbst ertellten Policy übergeben. Diese Policy wird an alle neu erstellten Items weitergereicht.
   Wenn sich nicht jeder Item gleich verhalten soll, kann im ItemAdded-Ereignis die PlacementPolicy für jeden Item überschrieben werden.

ValueConverter

Ein ValueConverter kann dazu verwendet werden um einen Wert aus der Datenquelle in einen für die Darstellung im Weltkoordinatensystem günstigeren Wert umwandeln. ValueConverter kann man dann einsetzen wenn der Wert aus den Daten noch nicht zur Darstellung im Weltkoordinatensystem geeignet ist.

Beispiele:

- *Beispiel 1: Umwandeln eines Bestellstatus in eine Farbe (um dann in weiterer Folge bspw. die Hintergrundfarbe daran binden zu können)*
- *Beispiel 2: Mappen eines Fremdschlüsselwertes in eine aufsteigende Zahl, die die alphabetische Reihenfolge der referenzierten Daten widerspiegelt.*

Einen ValueConverter einsetzen

Für jede Bindung einer Item-Eigenschaft an ein Feld der Datenquelle kann ein ValueConverter als optionaler Parameter mit übergeben werden:

```
With MyNetView
      ...
      .BindItemTextTo "KundeName"                              ' kein Converter
      .BindItemBackColorTo "Status", MyStatusConverter
      ...
End With
```

Standard-ValueConverter

Standardmäßig wird kein ValueConverter eingesetzt. Die Werte aus den Daten werden direkt der jeweiligen Eigenschaft zugewiesen.

MappingValueConverter

Es gibt in NetView zur Zeit eine Implementierung eines ValueConverters, den MappingValueConverter. Mit seiner Hilfe können Mappings definiert werden, die angeben dass ein Wert aus den Daten auf einen anderen Wert für das Steuerelement umgewandelt werden soll.

Den Mapping-Converter konfigurieren

Die Mappings können mit der Methode AddMapping konfiguriert werden. Einen Standardwert kann man mit der Eigenschaft DefaultValue angeben.

Beispiel:

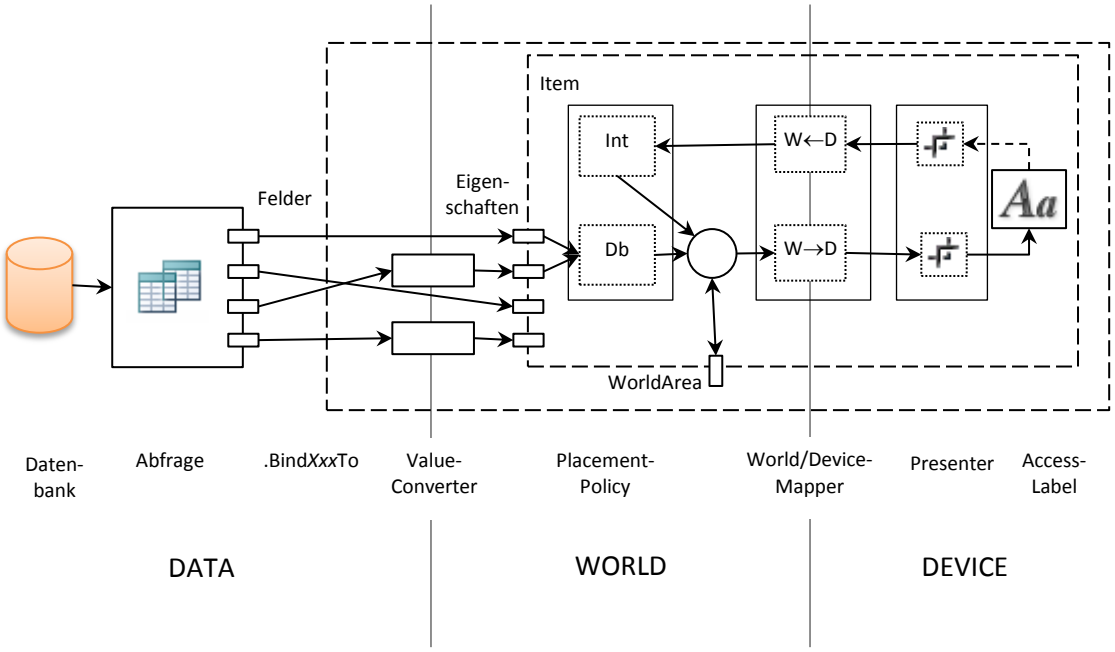
```
Dim StatusMapping As NV_MappingValueConverter

Set StatusMapping = NetView.Create.MappingValueConverter()
With StatusMapping
   .AddMapping StatusEnum.Geplant,  vbCyan
   .AddMapping StatusEnum.InArbeit, vbYellow
   .AddMapping StatusEnum.Erledigt, vbGreen
   .AddMapping Null, vbRed
   .DefaultValue = vbWhite
End With

...

MyNetView.BindItemBackColorTo "Status", StatusMapping
```

Einen eigenen ValueConverter erstellen

1. Eine neue Klasse anlegen
2. Die Schnittstelle NV_IValueConverter implementieren
3. Beim Initialisieren des Databindings für das gewünschte Feld eine Instanz des ValueConverters als zweiten Parameter mitgeben. Ein ValueConverter wird typischerweise keine Zustand halten. Eine Instanz kann daher bei mehreren Feldern verwendet werden.

Datenfluss

Zur Verdeutlichung des Zusammenspiels der einzelnen Komponenten von des Steuerelements ist im Folgenden der Fluss der Daten von der Datenbank bis hin zum tatsächlich sichtbaren Label (und tlw. auch zurück) dargestellt:



Weitere Vorhaben und Ideen (Auswahl)

Im Folgenden sind einige geplante Features aufgeführt, die zwar bereits mitbedacht, aber noch nicht umgesetzt sind:

- Einschränkungen der möglichen Manipulationen (z.B. Verschieben/Größenänderungen in nur eine Richtung) – mittels PlacementPolicy?
- Zooming mit dem Mausrad
- ad Databinding
    - Auf weitere Eigenschaften ausweiten
    - Methode Save() implementieren (Zurückschreiben der Daten, ValueConverter!)
    - Automatisches Polling nach neuen/geänderten Datensätzen (ähnlich Access?)
- Ereignisse
    - Change
    - BeforeUpdate
    - AfterUpdate
- Mit sehr vielen Items umgehen können (LabelManager)
- Z-Reihenfolge implementieren (LabelManager)
- Formatierungen (normal, selektiert, hover, ...)
- Tastaturbedienung
- Implementierung von Verbindungen (NetView!)

Known Issues

- Fehler in Recordsource führen zu eventuell unklarem Laufzeitfehler.
- Nachhinkende Größenänderung – Beim Ändern der Größe des Steuerelements (insbesondere beim Maximieren und Wiederherstellen) wird die Darstellung nicht sofort korrekt angepasst. Sobald eine weitere Manipulation erfolgt (kann auch Panning sein), ist die Darstellung wieder korrekt.
- Sehr große Formulare – Steuerelementpositionen und -größen sind in Integers gespeichert. Bei etwas mehr als 22“ wird die Grenze des Zahlenbereichs erreicht. Wenn ein Formular z.B. sehr breit gemacht wird kommt es daher zu einem Laufzeitfehler. NetView schon vorher abschneiden.
- Performance – bei vielen Items dauert das Databinding als auch Manipulationen an allen Items (Panning, Zooming, Resizing) recht lang.
- Unter Access 2010 dauert das Öffnen eines Formulars mit einem NetView Steuerelement relativ lange.

Wie kann ich das Projekt unterstützen?

- Ausprobieren
- In Anwendung einbauen
- Rückmelden
    - was hat gefallen
    - was ist unklar
    - was fehlt
- Diskutieren (persönlich und im Forum auf Codeplex)

- Einen guten Namen vorschlagen
- An der Dokumentation mitarbeiten
    - Wiki auf Codeplex
    - SourceCode-Dokumentation auf Deutsch
    - Struktur und Layout
- Den Build-Prozess vervollständigen (AutoIt-Scripts)
- Tutorials, Demos oder HowTo-Videos verfassen
    - im Wiki (Contributor werden)
    - Link ins Forum posten
- Eigene Komponenten erstellen und zur Verfügung stellen:
    - PlacementPolicy
    - ManipulationKindProvider
    - ValueConverter
- Am Sourcecode mitarbeiten
    - Known Issues beheben
    - Fehlende Features implementieren
- Eine Entwurfszeitunterstützung implementieren
- Testen unter Access 2002, 2007 und 2013.
- In das 2000er-Format konvertieren unter Access 2000 testen

API-Dokumentation

Die gesamte öffentliche API des Steuerelements ist auf der Projektseite (http://accessnetview.codeplex.com) dokumentiert.

OpenSource und Lizenz

Access NetView ist OpenSource. Es steht jedermann frei das Ergebnis des Projekts selbst oder den Code (oder Teile davon) für eigene Zwecke einzusetzen. Dies gilt sowohl für kommerzielle als auch nicht-kommerzielle Anwendungen.

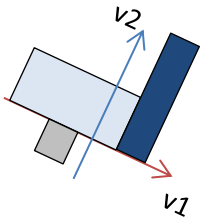
Alle Tätigkeiten rund um das Projekt werden öffentlich durchgeführt. Kommunikation und Beiträge zum und über das Projekt (Anregungen, Diskussionen, Codebeiträge ...) sollen vorzugsweise über die Projekt-Homepage (http://accessnetview.codeplex.com) erfolgen.

Die Codeplex-Seite ist die zentrale Anlaufstelle für das Projekt. Hier gibt es Zugriff auf den aktuellen Sourcecode, Releases, Dokumentation und Hintergrundinfos.

Anhang

Alle Orientierungen im Überblick

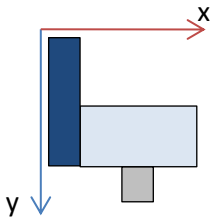
Angenommene Daten im Weltsystem:



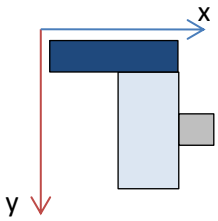
Hinweis: Die schräge Darstellung ist kein Hinweis auf eine freie Drehung. Sie soll andeuten, dass das Weltkoordinatensystem im Allgemeinen kein räumliches Bezugssystem ist.

Darstellung entsprechend der Eigenschaft Orientation

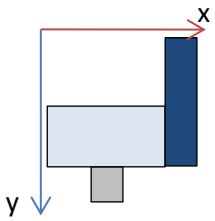
LeftUp



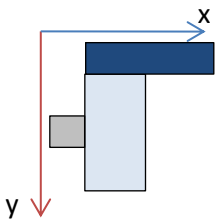
UpLeft



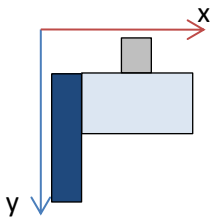
RightUp



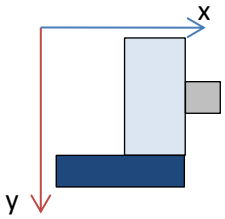
UpRight



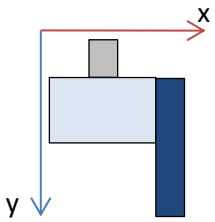
LeftDown



DownLeft



RightDown



DownRight