

22.5.5 Klasse DataControl (gb.db.form)

Dieses Steuerelement ermöglicht die Anzeige und das Editieren von Daten für ein ausgewähltes Feld mit einem speziellen (internen) Editor. Folgende Voraussetzungen müssen erfüllt sein:

- Es existiert eine Verbindung vom DB-Client zur ausgewählten Datenbank.
- Das DataControl befindet sich im Container DataSource.
- In der DataSource ist die DB-Tabelle festgelegt, auf welche zugegriffen werden soll.
- Für das DataControl ist das Feld angegeben, dessen Daten angezeigt und bearbeitet werden sollen.

Ein (erweitertes) Beispiel finden Sie im Abschnitt 22.5.5.4.

So können Sie ein neues Steuerelement *DataControl* erzeugen:

```
Dim hDataControl As DataControl
hDataControl = New DataControl ( Parent As Container ) As "event name"
```

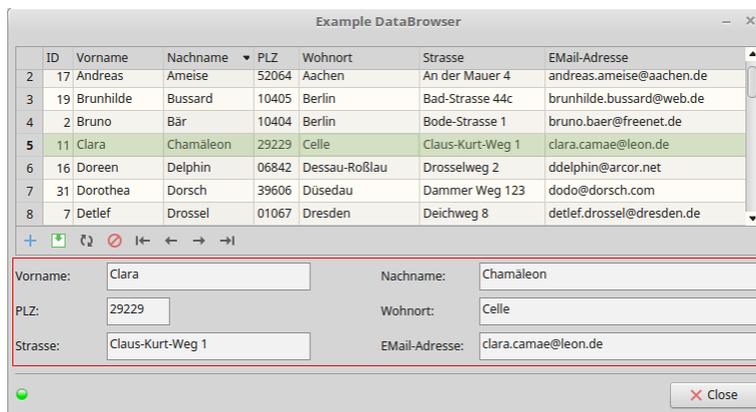


Abbildung 22.5.5.1: Sechs DataControls im markierten Bereich

22.5.5.1 Eigenschaften

Die Klasse *DataControl* verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Control	Control	Gibt den speziellen Editor zum Editieren eines bestimmten Feldes zurück, denn es hängt vom Feldtyp ab, welcher Editor im DataControl verwendet wird. Beispiele: Datum: DateBox oder Text: TextBox.
Field	String	Gibt das Feld an, dessen Daten mit dem DataControl angezeigt und bearbeitet werden sollen. Die DB-Daten kommen von der ersten, übergeordneten Datenquelle (DataSource). Sie können den Wert der Eigenschaft auch auslesen.
Modified	Boolean	Der Wert ist True, wenn der Inhalt im DataControl geändert wurde. Die Eigenschaft kann nur gelesen werden.
Valid	Boolean	Ist True, wenn der Inhalt des DataControls gültig ist.
Value	Variant	Setzt den Wert für das DataControl. Sie können die Eigenschaft auch auslesen.

Tabelle 22.5.5.1.1 : Eigenschaften der Klasse DataControl

22.5.5.2 Methoden

Die Klasse *DataControl* verfügt nur über eine relevante Methode: Update(). Die Prozedur lädt den (aktuellen) Daten-Inhalt erneut in das DataControl.

22.5.5.3 Ereignisse

Die Klasse *DataControl* hat nur dieses Ereignis, das für das Steuerelement DataControl von Belang ist:

- **Validate (Value As Variant)**
 Das Ereignis wird ausgelöst, wenn der Inhalt der DataControl validiert werden muss. `Value` ist der zu prüfende Wert der DataControl-Liste. Dieses Ereignis ermöglicht es dem Benutzer, spezifische Validierungsanforderungen hinzuzufügen. Um die Daten zu invalidieren, stoppen Sie das Ereignis einfach mit STOP EVENT.

```
Public Sub DataControl1_Validate(Value As Variant)

    If IsNull(Value) Then Return
    If Not IsInteger(Value) Then Return

    ' Valid range: [0,100]
    If Value < 0 Or Value > 100 Then ' Diese Zeile wird vom Typ des Feldes und von den Werten bestimmt
        Stop Event
    Endif

End
```

22.5.5.4 Beispiel

Das Beispiel bezieht sich auf ein Feld, in dem Datumswerte – wie zum Beispiel ein Geburtstag – abgespeichert werden. Der Editor ist in diesem Fall eine DateBox, deren Mode-Eigenschaft aber auf den Wert *DateTime* gesetzt ist. Auch wenn Sie den Feldtyp auf Date gesetzt haben, wird nach dem korrekten Datum stets auch automatisch der leere Zeitanteil angezeigt wie bei `12.12.1978 00:00`. Eine Änderung des Ausgabeformats erreichen Sie mit dem rot markierten Quelltext, durch den die Mode-Eigenschaft der DateBox auf *Date* (=0) gesetzt wird. Die Idee stammt von Gianluigi (bagonergi@gmail.com).

```
Public Sub Form_Open()

    Dim hDateBox As Datebox

    '--- DB-Verbindung (SQLite) konfigurieren
    hDBConnection.Type = "sqlite3"
    hDBConnection.Host = User.Home & "DatabasesSQLite"
    hDBConnection.Name = "Adressen"

    '--- DB-Verbindung öffnen
    hDBConnection.Open()

    '--- DB-Tabelle festlegen
    DataSource1.Table = "adressliste"
    DataSource1.MoveFirst()

    '--- Die einzelnen DataControls – Prefix `dc` – mit den entsprechenden Datenbank-Feldern verbinden
    dcNachname.Field = "Nachname"
    dcVorname.Field = "Vorname"
    cdWohnort.Field = "Wohnort"
    dcPLZ.Field = "PLZ"
    dcStrasse.Field = "Strasse"
    dcEmailAdresse.Field = "EmailAdresse"
    dcWebAdresse.Field = "WebAdresse"
    dcHinweise.Field = "Hinweise"
    dcTelefonFestnetz.Field = "TelefonFestnetz"
    dcTelefonMobil.Field = "TelefonMobil"
    dcGebDatum.Field = "GebDatum"

    '--- Format der (internen) Date-Box von `dcGebDatum` über deren Mode-Eigenschaft ändern
    hDateBox = dcGebDatum.Children[0]
    hDateBox.Mode = 0

End
```

Damit erfolgt eine korrekte Anzeige des Datums – ohne Zeitanteil:



Abbildung 22.5.5.4.1: Anzeige Datum