

## Kurzanleitung für die Komponente:

### TBBDBGridColorControl



Copyright by  
Peter Schrade  
Moorkampstraße 6  
D-27755 Delmenhorst  
eMail: [peter@jasocul.de](mailto:peter@jasocul.de)

#### Aufgabe der Komponente:

Die Komponente dient zur besseren Darstellung von Datenbankfeldern in einem Datenbank-Gitter (DBGrid oder dessen Nachkommen). Es können die typischen Formatierungsvorgaben gemacht werden, die man aus dem Feld-Editor von Delphi kennt. Zusätzlich können Farb-Bedingungen an ein Datenbankfeld geknüpft werden. Für das Datenbank-Gitter kann eine Farbe bestimmt werden, die in jeder zweiten Zeile dargestellt wird.

Über einen Methoden-Aufruf kann man sogar dem Endanwender einen Zugriff auf die Darstellung gewähren.

#### Eigenschaften:

- **Active:**  
Mögliche Werte: True und False. Voreingestellt ist True.  
Bei True ist die Komponente aktiv, bei False nicht.
- **DataSource:**  
Hier muss die TDataSource (oder ein Nachkomme) eingetragen werden. Es wurde bewusst kein TDataSet genommen, da es um eine Visualisierung der Daten geht. Das macht aber nur Sinn, wenn eine TDataSource verwendet wird.
- **Grid:**  
Hier muss die TDBGrid (oder ein Nachkomme) eingetragen werden.
- **INIFile:**  
Soll zum Speichern der Einstellungen eine INI-Datei verwendet werden, so ist hier der Name einzutragen. Wird kein Pfad vorgegeben, so wird der Pfad der Anwendung verwendet. Ich empfehle beim Einsatz von INI-Dateien, möglichst für jede Tabelle eine eigene Datei zu verwenden. Dies ist ein erheblicher Geschwindigkeitsvorteil.
- **Name:**  
Name der Komponente.
- **RegistryMainKey:**  
Bei Verwendung der Registry zum Speichern der Einstellungen.
- **RegistrySubKey:**  
Bei Verwendung der Registry zum Speichern der Einstellungen.
- **SelectIdentifier:**  
Werden in einem TDataSet, das über die DataSource mit der Komponente verbunden ist, mehrere verschiedene Tabellen verwendet, kann hier ein wechselnder Wert für die jeweilig zugehörige Tabelle eingetragen werden. Dadurch wird vermieden, dass die falschen Einstellungen geladen werden. Voreingestellt ist „0“.
- **Tag:**  
Der Standard-Tag aller Komponenten.
- **UseAutoMultiSelect**  
Bezieht sich direkt auf SelectIdentifier. Steht diese Eigenschaft auf True, so wird als SelectIdentifier die Anzahl der Felder einer Tabelle eingetragen. Bei Einstellung False,

hat diese Eigenschaft keine Auswirkung. Voreinstellung ist True.

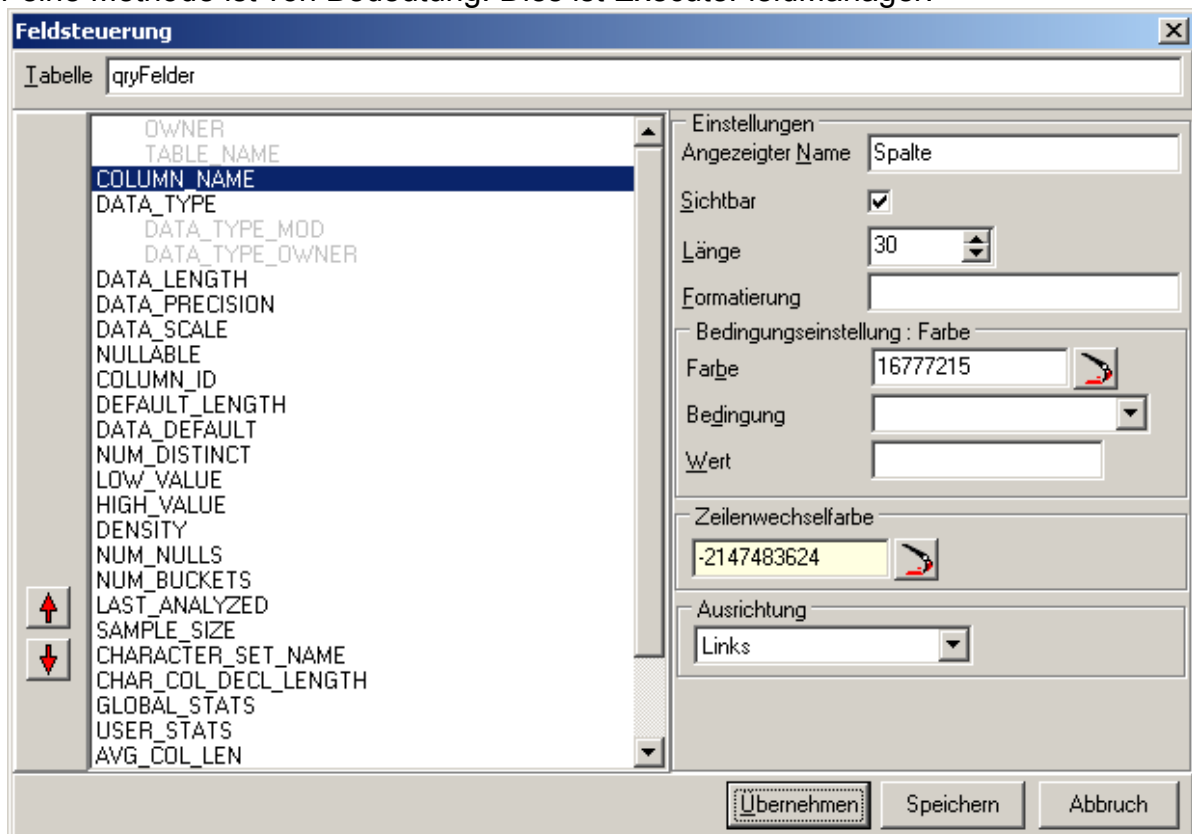
- UseINIFile:  
Wenn eine INI-Datei zur Speicherung der Einstellungen verwendet werden soll, muss hier True eingetragen sein. Voreingestellt ist False.
- Zeilenfarbe:  
Die Farbe jeder zweiten Zeile im eingetragenen Grid.
- Zeilenfarbwechsel:  
Bestimmt, ob die eingestellte Zeilenfarbe auch aktiv sein soll.

#### Ereignisse:

keine

#### Methoden:

Nur eine Methode ist von Bedeutung. Dies ist ExecuteFieldManager.



Der Aufruf führt zu folgendem Fenster: Links ist die Spalte der verfügbaren Felder (Multi-Select ist möglich).

Die roten Pfeile dienen der Veränderung der Spalten-Reihenfolge.

Angezeigter Name: verändert die Spaltenüberschrift.

Sichtbar: schaltet die Anzeige einer Spalte ein, bzw. aus.

Länge: stellt die Anzahl der angezeigten Spaltenbreite ein.

Formatierung: verändert die Anzeige. (#,###0.00 stellt numerische Daten mit Tausender-Trennzeichen und 2 Nachkommastellen ein)

Bedingungeinstellung: kann eine Farbe in Abhängigkeit vom Feld-Inhalt darstellen.

Zeilenwechselfarbe: stellt jede zweite Zeile in der angegebenen Farbe dar.

Ausrichtung: bestimmt die Ausrichtung des Feld-Inhalts.

### Hinweise und Tipps:

Ob man zur Speicherung der Einstellungen die Registry oder eine INI-Datei verwenden soll, hängt von verschiedenen Faktoren ab. Für die Geschwindigkeit ist die Benutzung der Registry dringend zu empfehlen. Es ist aber zu beachten, dass die Datenmenge sehr schnell anwächst. Zu jedem Tabellen-Feld werden mehrere Einträge gemacht.

Ist das nicht gewünscht, sollte man INI-Dateien verwenden. Meine Empfehlung ist, viele INI-Dateien zu verwenden. In einer Anwendung kann es sehr schnell passieren, dass die Datenmenge schnell die 1MB-Barriere überschreitet. Bei einer solchen Dateigröße muss beim ersten Aufruf mit Ladezeiten über 5 Sekunden gerechnet werden.

Die Voreinstellung bei UseAutoMultiSelect ist True. Dadurch wird bei der Nutzung verschiedener Tabellen in einem DataSet immer ein neuer SelectIdentifier gesetzt. Dies ist auf der einen Seite sehr praktisch. Auf der anderen kann es aber auch ärgerlich sein. Angenommen UseAutoMultiSelect ist aktiviert und in der Datenbank-Tabelle wird ein weiteres Feld angefügt. Dann könnten die bisherigen Einstellungen nicht mehr abgerufen werden. Es wäre natürlich möglich, die Registry- oder INI-Einträge manuell zu korrigieren, aber wäre UseAutoMultiSelect nicht aktiviert, gäbe es dies Problem nicht.

Es ist also vom Einsatzfall abhängig, was bevorzugt werden sollte. Ich deaktiviere es in den meisten Fällen.