

GDI Reportdesigner

FastReport 2.5

Inhaltsverzeichnis

Einführung	5
Einleitung	6
Bemerkungen über FastReport	7
Die Geschichte von FastReport	8
Reporte erstellen	10
Daten	10
Parameter	11
Formulare	12
Verarbeitung	13
Reporte vorbereiten	14
Beispiel: einfacher Report erstellen	15
FastReport Objekte	17
Das "Rechteckobjekt" Objekt	18
Das "Bereich" Objekt	22
Das Bild Objekt	24
Das Unterreport Objekt	25
Das Linien Objekt	26
Das RTF Objekt	26
Das OLE Objekt	27
Das Diagrammobjekt	28
Das Figur Objekt	30
Das Barcode Objekt	30
Den Report erstellen	32
Bänder in FastReport	32
Einfacher Report (Liste)	36
Haupt-Detail Report	36
Haupt-Detail-Unterdetail Report	36
Kreuztabellen Report	36
Dynamische Reporte	37
Gebrochene Bänder	38
Mehrspalten Report	39
Report mit Titelseite	39
Verschachtelte Reporte (Unterreporte)	40
Haupt-Detail-Detail Report	40
Zusammengesetzte Reporte	Fehler! Textmarke nicht definiert.
Report mit BLOb Feldern	40
Report ohne Bänder	40
Report mit Gruppen	41
Report mit Diagrammen	41
Der Designer	43
Verwendung der Maus	44
Report Optionen	45
Seitenoptionen	46
Der "Papier" Karteireiter	46
Der "Quelle" Karteireiter	46
Der "Ränder" Karteireiter	47
Der "Optionen" Karteireiter	47

Designer Optionen.....	48
Der Objekt Inspektor.....	50
Das "DB-Feld einfügen" Fenster.....	51
Der „Variableneditor“	52
Der Ausdruckeditor.....	57
„DB-Feld einfügen“ Dialog	57
„Variable einfügen“ Dialog	58
„Funktion einfügen" Dialog	58
Symbolleisten.....	60
Die "Standard" Symbolleiste	60
Die "Text" Symbolleiste	61
Die "Rechteck" Symbolleiste	62
Die "Ausrichtung" Symbolleiste	63
Endanwenderfunktionen.....	64
Einführung.....	65
Die Dialog-Formulare	66
Beschriftung	68
Eingabefeld	69
Memofeld.....	69
Druckknopf	70
Häkchenbox	71
Radioknopf	71
Liste.....	72
Kombinationsfeld.....	73
Übergeben der Informationen an den Report	74
Datenzugriffskomponenten	76
Beschreibung der FastReport DB-aware Komponenten.....	77
TfrDBLookupComboBox.....	78
TfrIBXTable	79
TfrIBXQuery	81
TfrIBXDataBase	83
Reporte erstellen	Fehler! Textmarke nicht definiert.
Einfacher "Tabellarischer" Report.....	84
Report mit Parametern	84
Feldeditor	86
Lookupfeld erstellen	86
Abfrageparametereditor.....	87
Daten verbinden	87
Parameterdialog	88
Parameterdialogdesigner	Fehler! Textmarke nicht definiert.
Eingebaute Sprache.....	89
Scripte und Objekte	90
Code schreiben	91
Objekt Abänderung.....	95
Eingebaute Funktionen	96
Eingebaute Funktionen	96
Angesammelte Funktionen.....	96
String Funktionen	96
Eigenschaften und Methoden von Objekten.....	99

Standardobjekte	101
Add-in Objekte	103
Dialog Bedienungselemente	104
Datenzugriffskomponenten	108
Verwendung des Interpreters	111
Reportbeispiele	Fehler! Textmarke nicht definiert.
Reportbeispiele	112
Einfügen von Grafiken und Diagrammen in ein Report	112
Kontrollieren der Logik der Reportgestaltung mittels dem OnManualBuild Event	Fehler! Textmarke nicht definiert.
Manuelle Reportgestaltung zur Laufzeit mittels Code ...	Fehler! Textmarke nicht definiert.
Drucken von Spaltenreporten mit variabler oder unbekannter Anzahl von Spalten	114

Einführung

Einleitung

Bemerkungen über FastReport

Die Geschichte von FastReport

Stufen der Reporterstellung

Erstellen eines einfachen Reports

Einleitung

Diese Anleitung setzt voraus, dass Sie mit Reportdesignern vertraut sind und die Grundkonzeption dieser Programme verstehen (z.B. Bänder, Datenquellen, zwei-Durchlauf Reporte, usw.). Diese Anleitung wird Ihnen helfen Reporte mit FastReport zu erstellen, aber Sie wird Ihnen nicht mit den Grundkonzepten von Reportdesignern helfen.

Wenn Sie mit Reportdesignern nicht vertraut sind, empfehlen wir Ihnen die QuickReports Hilfe aufzurufen. Eine Anleitung für QuickReports wird bei Ihrem Delphi mitgeliefert. Die Grundkonzepte von QuickReports kann man in den meisten Fällen auf FastReport anwenden, dennoch, FastReport bietet viel mehr Flexibilität und Endanwender-Anpassung.

Bemerkungen über FastReport

FastReport ist ein höchst flexibler Reportdesigner, wobei die Daten für den Report von fast jeder Datenquelle bezogen werden können, einschließlich Stringlisten, BDE Datenbanken, ADO Datenquellen (ohne BDE-Nutzung), Interbase (mittels IBO), Pascal Arrays und Records, um nur einige zu nennen!

Das gesamte FastReport System wurde in Delphi Pascal geschrieben. Das FastReport System benötigt keine DLL's und fügt dem leeren Projekt ungefähr 400kb (Delphi 5) hinzu. Wenn Sie Endanwender Designfähigkeit zulassen wollen, wird Ihrer .EXE weitere 500kb hinzugefügt. Auch wenn dies groß erscheint ist es nur ein Bruchteil gegenüber den anderen Reportdesignern. Sie sollten auch beachten dass FastReport die Möglichkeit bietet, dass der Endanwender nicht nur das Design des Reports ändern kann sondern auch die Abfragen und Datenbanken woher der Report seine Daten holt. FastReport besitzt auch eine eigene Scriptsprache um zu gewährleisten dass jeder Report leicht von Programmierern und Endanwendern geändert werden kann. Wenn mehrere Programme von Ihnen FastReports verwenden, können Sie einfach die FastReport BPL (ca. 1400kb) einsetzen und Ihre Programme bleiben klein.

FastReport besitzt eine der attraktivsten Benutzeroberflächen mit den neuesten Oberflächenkomponenten wie z.B. andockbare Symbolleisten. Der Endanwender wird den Designer sicherlich befürworten da fast alle Reports nur mithilfe der Maus erstellt werden können.

FastReport hat **FastReport** nicht umsonst so genannt: vergleichen Sie ihn mit anderen Delphi Reportdesignern werden Sie sehen dass in Sachen Geschwindigkeit nichts rankommt. Auch das Reportvorschaufenster lässt die meisten Reportdesigner im Regen stehen, mit seinem glatten, hochpolierten Look gibt es Ihrem Programm ein hochprofessionelles Aussehen.

FastReport ist ein erprobter Reportdesigner welcher schon über 3 Jahre erhältlich ist und immer mehr Features anbietet, mit denen kein anderer Delphi Reportdesigner mithalten kann.

Die Geschichte von FastReport

FastReport wurde unter Notwendigkeit gegründet. Während ich in 1997 eine Gehaltsabrechnung entwickelte, habe ich mich nach einem Reportdesigner umgeschaut welcher mir die Möglichkeit gab, einfach Reporte zu erstellen und die Reporte während der Laufzeit zu ändern. Jedoch gab es zur Zeit keine frei erhältlichen Komponenten welche meinen Anforderungen entsprachen. Also entwickelte ich mein eigenes Reporttool und FastReport war geboren.

Das Grundkonzept von FastReport wurde aus "1S-Bokkeeping" 6.0 für Windows entnommen, wobei das Hauptreportelement ein umrahmtes Rechteckobjekt mit mehreren Zeilen Text ist. Dieser Text kann auf der einen Seite Standardtext enthalten und auf der anderen Seite Variablen. Variablen, wie Datenfelder, werden durch eckige Klammern gekennzeichnet. Die erste Version des Reportgenerators unterstützte nur ein Band, aber man konnte mehrstufige Reporte erstellen. Es war auch keine Komponente sondern ein bloß ein Ansammlung von Units.

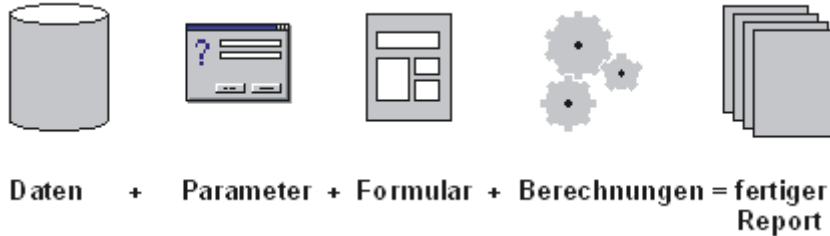
Später, im Jahre 1998 wurde der Reportgenerator eine Delphi Komponente. Zu diesem Zeitpunkt bekam er den Namen "FastReport" and er bot erhöhte Funktionalität an. Heutzutage ist FastReport ein ausgewachsener, gewerblicher und visueller Reportgenerator. Einige seiner Fähigkeiten:

- Eingebauter Reportdesigner welcher auch zur Laufzeit verfügbar ist (perfekt für Endanwender Reportänderungen!).
- Vorschau ähnlich der MS Word Druckvorschau.
- Kompiliert direkt in die Delphi EXE, keine DLL's nötig.
- JPEG (mithilfe Delphi Bibliothek) und GIF (mithilfe der RX Bibliothek) Unterstützung.
- Gute Performance vergleichbar mit QuickReport1 bloß mit mehr Features.
- Kompakter, purer Delphi Code – ohne dem Designer ist die Basisfläche nicht größer wie in QR3!
- Mächtiger bandorientierter Reportgenerator wie QuickReports und ReportBuilder.
- Sammlung nützlicher Komponenten einschließlich: Rechteckobjekt, Linie, Bild, Figur, OLE Objekt, RTF-Objekt, Diagramm, Barcode, schattiertes Rechteck.
- Unbegrenzte Anzahl von Seiten innerhalb des Reports.
- mehrseitige Reporte; zusammengesetzte Reporte; Unterreports; Gruppen; mehrspaltige Reporte; Haupt-Detail-Detail Reporte; Kreuztabellen Reporte; 2 Durchgänge Reporte.
- Volle Kontrolle über den Druckvorgang; unterstützt alle Papiergrößen.
- TXT, RTF (mit Graphiken), CSV, HTML (mit Graphiken) Export Filter.
- Textsuche in der Reportvorschau.
- Mit der zusätzlichen Komponente TfrDataStorage können Sie Tabellen erstellen und Abfragen während der Laufzeit starten. Dies ist besonders für Endanwenderreporte nützlich um ihnen die volle Kontrolle über den Reportvorgang zu geben.
- Die Seiten des vorbereiteten Reports können geändert werden.
- Ein eingebauter Pascal-ähnlicher Interpreter wird für äußerst flexible Reporterstellung mitgeliefert! Syntaxhervorhebung wird mithilfe einer frei erhältlichen Freewarebibliothek unterstützt.

- Die Reportdaten können in der Delphi DFM Datei, einer externen Datei oder einem BLOB Feld in einer DB Tabelle abgelegt werden oder sie können gestreamt werden.
- FastReport kann einfach durch Erstellung Ihrer eigenen Reportkomponenten, Assistenten und Funktionsbibliotheken erweitert werden.
- IBObjekte im Reporterkern unterstützt (ohne der BDE).
- Interbase Express (IBX) Unterstützung.
- ActiveX Data Objects (ADO) Unterstützung im Reporterkern.
- Komplette Datenmanager Funktionalität (ohne der BDE).

Reporte erstellen

Reporte erstellen besteht aus den folgenden Phasen:



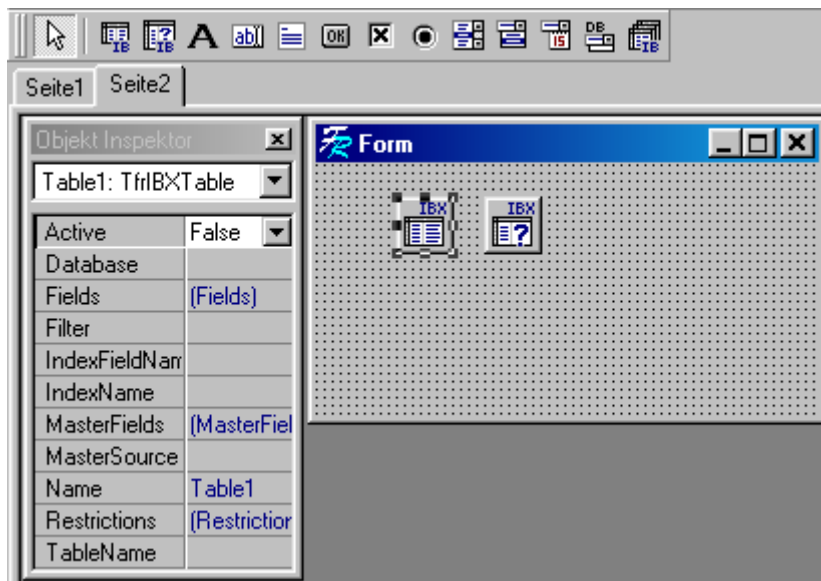
Wir werden jede dieser Phasen und ihre Ausführung in FastReport anschauen.

Daten

Die meisten Reporte basieren auf Daten einer Datenbank. Delphi selber besitzt wirksame Mechanismen für den Zugriff auf Datenbanken. Diese Mechanismen werden auch in FastReport verwendet. Die Ttable und Tquery Komponenten können als Quellen für ein Report verwendet werden. Gewöhnlich ist es möglich jeden Vorläufer einer Tdataset Komponente zu verwenden. Der Zugriff auf die Daten wird innerhalb des FastReport Kernels ohne die Beeinflussung eines Programmierers realisiert.

Bis auf Daten die in einer Datenbank gespeichert sind, kann FastReport praktisch alle Quellen verwenden (Arrays, Dateien, StringGrids usw.). In solchen Fällen muss der Programmierer aufpassen auf die „Nicht-Datenbank“ Quelle zuzugreifen. In FastReport gibt es einige Events die dem Programmierer erlauben Daten an den FastReport Kernel zu übergeben.

Datenzugriff ist in etwa allen Reportgeneratoren gleich. Alle Reportgeneratoren können mit DB- Komponenten die auf dem Dialogformular platziert werden, umgehen. Außer Datenzugriff erlauben FastReport, ReportBuilder und QR+QR Designer die Erstellung neuer Komponenten zur Laufzeit. In FastReport ist die Erstellung von DB- Komponenten vergleichbar mit der Delphi IDE. Ebenso wie in Delphi platzieren Sie Komponenten auf ein Formular und ändern die Eigenschaften im Objekt Inspektor. Die Komponentenideologie ist sehr flexibel: es ist möglich einfach neue Komponenten mit der Unterstützung verschiedener Datenbankengines zu erstellen.

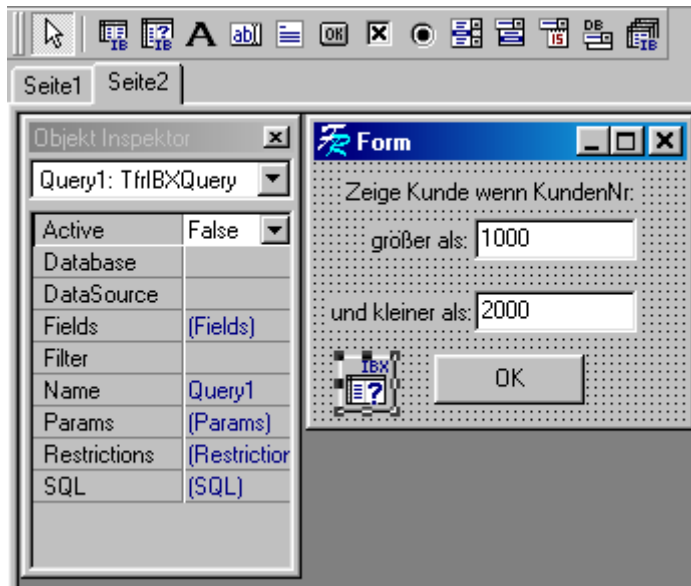


Parameter

Benutzer können nach Eingaben gefragt werden (Parameter). Beispiele für Benutzereingaben: Datum, Firmenname, Rechnungsnummer, usw. Manche Reporte verwenden gar keine oder feste Parameter (ohne deren Werte in einem Fenster anzufordern).

Die Handhabung der Parameter ist in den Reportgeneratoren verschieden umgesetzt. In ReportBuilder und QR+QRDesigner gibt es die Möglichkeit Parameter abzurufen wenn der Report Daten einer Abfrage verwendet. Für den Dialog mit einem Benutzer wird ein normales Dialogfenster verwendet. Ansonsten können Sie auch Formulare verwenden welche in Delphi erstellt wurden. Dies erfordert aber einige Änderungen in der Programmierlogik und Sie müssen Ihr Projekt neu kompilieren.

Im Gegenteil ermöglicht FastReport dem Endanwender die Dialog-Formulare zu entwickeln. Dieser Vorgang ist wie bei den Erstellungsformularen in Delphi: es gibt Standardbedienelemente welche auf dem Dialog-Formular platziert werden können. Wenn nötig können Sie dann ihre Eigenschaften ändern. Außerdem ermöglicht Ihnen die in FastReport eingebaute Sprache, Dialoge mithilfe der pascal-ähnlichen Programmierlogik zu erstellen und die eingegebenen Werte in den Kernel zu übergeben.

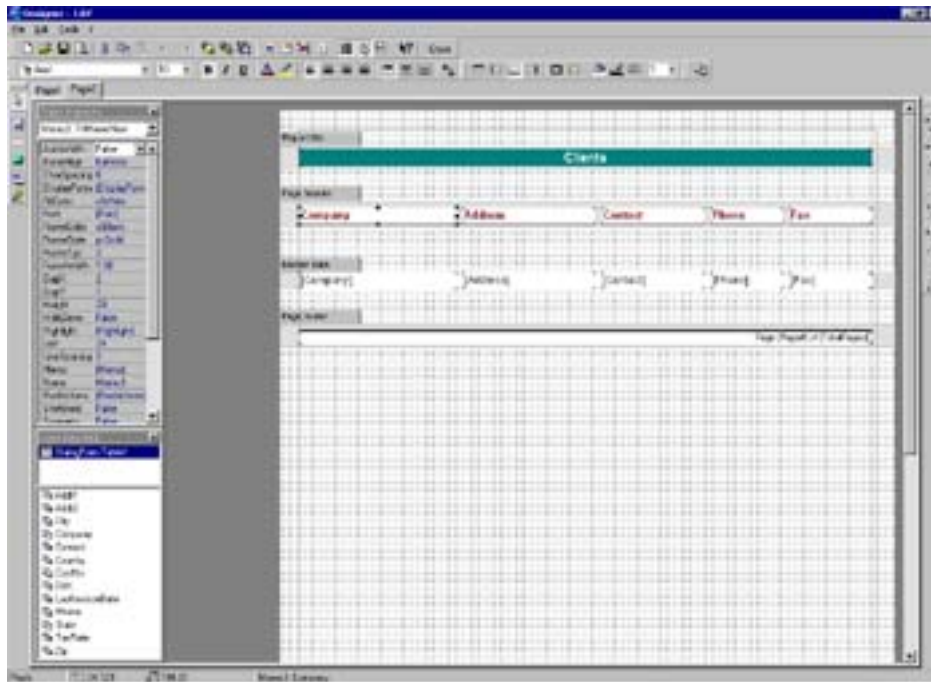


Die Möglichkeit Ihre eigenen Dialoge zu erstellen ist sehr nützlich. Andere Möglichkeiten (Datenquellen erstellen, Nutzung der eingebauten Sprache, usw.) ermöglichen Ihnen universelle Reporte zu erstellen z.B. Reporte die unabhängig des Programms arbeiten (Kompiliertes und gebautes Projekt). Dies ermöglicht Ihnen neue Reporte zu erstellen und alte bestehende ohne neu bauen oder neu kompilieren zu ändern.

Formulare

Das Reportformular selbst zeigt eine Sammlung von Elementen auf, die beschreiben wie ein Report aussehen muss. Um Elemente an ihrem Ort zu gruppieren verwendet FR Bänder. Es gibt 2 Arten von Bändern: Servicebänder (Reporttitel, Seitennummerierung, usw.) und Bänder welche mehrzeilige Teile eines Reports darstellen (z.B., Datenbänder). Datenbänder sind mit den Datenquellen verbunden und der Inhalt wird so lange angezeigt wie etwas in der Datenquelle steht.

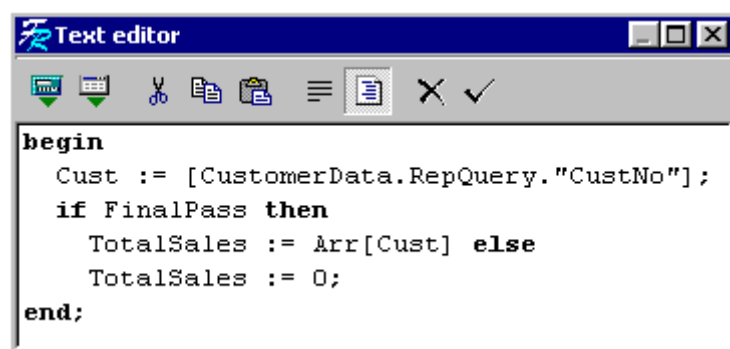
Die sichtbare Umgebung der Reportentwicklung –der Designer- wird zur Erstellung des Reports verwendet. Im FastReport Designer können Sie Reporte entwickeln, Leistung, Einfachheit und Bedienungskomfort vereinen. Die Oberfläche des Designers besteht aus Instrumententafeln (Symbolleisten). Natürlich können Sie die Symbolleisten platzieren wie Sie wollen. Um die Objekteigenschaften zu ändern, können Sie ein Objekt Inspektor ähnlich dem in Delphi verwenden.



Verarbeitung

Eingabedaten verarbeiten, verändern der Reportformulare oder der Komponenten sind Phasen (Vorgänge) in der Erstellung eines Reports. Ein Beispiel solcher Art wäre die negativen Beträge in Rot darstellen. Ein komplexeres Beispiel eines Vorgangs wäre den ganzen Betrag der verkauften Produkte einem Kunden auszudrucken welcher im Gruppenfuß dargestellt wird(oder einem Gruppenkopf).

Solch ein Vorgang realisieren Sie indem Sie eine Ereignisbehandlung in Delphi schreiben und genauso können Sie es in FastReport, QR oder ReportBuilder tun. Dieser Weg ist nicht allgemein da er es nicht ermöglicht neue Reporte außerhalb von Delphi zu erstellen ohne das Projekt zu erneuern. Deshalb ist in FastReport eine Sprache, ähnlich zu Pascal aber vereinfacht, eingebaut. Scripte die in dieser Sprache geschrieben werden sind Ereignisse die vor der Verarbeitung von Objekten gestartet werden. Dies ermöglicht Ihnen komplexe Vorgänge zu erstellen ohne Code in Delphi zu schreiben und dementsprechend den Report mit einem Projekt zu verknüpfen.

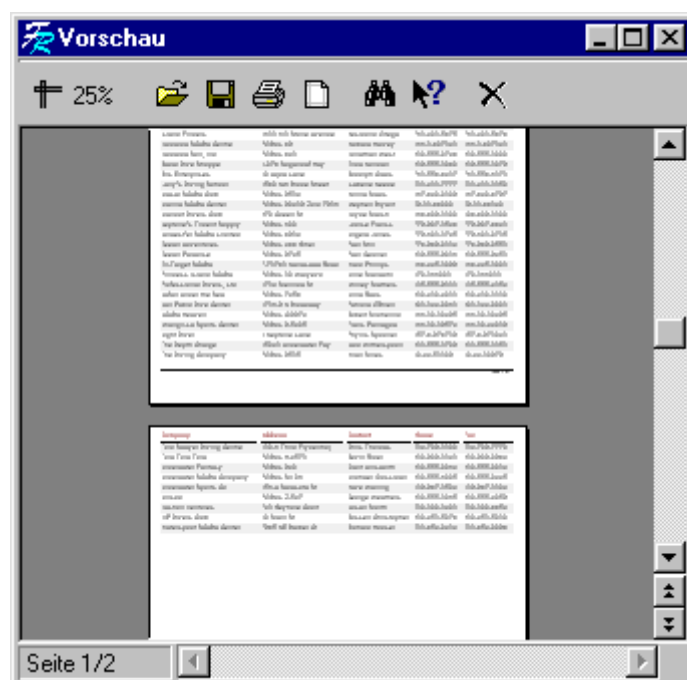


Die Möglichkeiten der eingebauten Sprache in FastReport ist sehr umfangreich. In Skripten können Sie alle Eigenschaften und Methoden eines Objekts wie auch die Variablen und Datenfelder verwenden. In Skripten ist es auch möglich Variablen und Arrays zu erstellen, welche dann, wenn sie dran sind, in allen Reports verfügbar sind.

Reporte vorbereiten


Vorbereitete Reports sind Produkte des FastReport Kernels, sie können durch klicken auf den „Reportvorschau“ Button angeschaut werden. Nicht wie viele andere Reportgeneratoren welche den Inhalt der Reportdateien in Metadateien halten (z.B. Bilder im EMF Format), stellt der in FastReport vorbereitete Report eine Sammlung von Objekten dar, die den Inhalt jeder Seite des verarbeiteten Reports beschreiben. Dadurch können Sie den vorbereiteten Report ändern indem Sie die benötigte Seite in den Designer laden. Außerdem ist es möglich die Reaktion der Objekte im Vorschaufenster auf Mausklicks darzustellen. Dies ermöglicht Ihnen Ihre Arbeit leicht zu organisieren (Sie klicken auf ein Report und dadurch wird ein neuer Report mit genaueren Informationen erstellt).

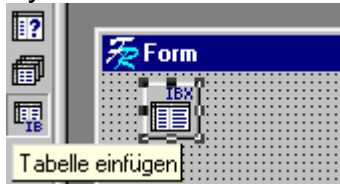
Das Vorschaufenster von FastReport unterscheidet sich von anderen Reportgeneratoren. Dokumente in FastReport anzusehen ist wie in Microsoft Word: in einem Fenster kann man mehrere Seiten betrachten. Außerdem können Sie den ganzen Text in allen Dokumenten durchsuchen.



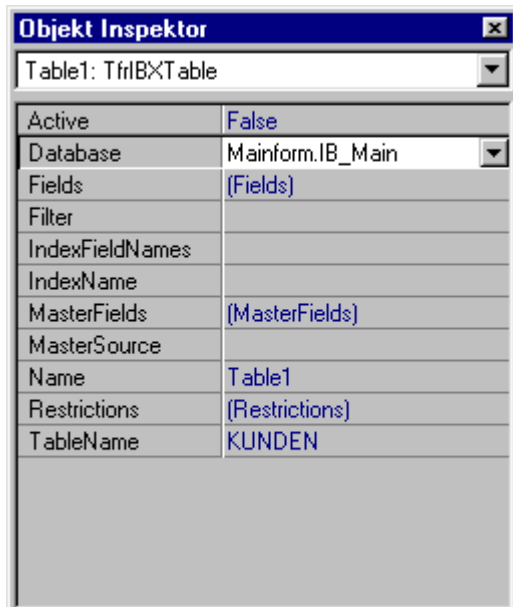
Beispiel: einfacher Report erstellen


Lassen Sie uns einen der einfachsten Reporte erstellen:

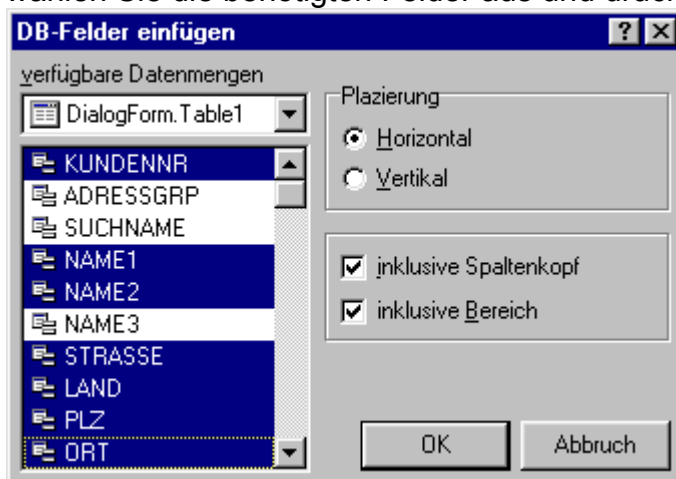
1. rufen Sie den Report Designer auf.
2. wählen Sie Menü Bearbeitet|Dialog-Formular hinzufügen.
3. Fügen Sie eine TIBXTable Komponente ein. Dazu klicken Sie in der Objektpalette Symbol  an. Der zweite Mausklick auf der Form gemacht, z.B. so:



4. Wählen Sie Menü Werkzeuge|Symbolleiste|Objekt Inspektor an oder drücken Sie F11 Taste um Objekt Inspektor sichtbar zu machen.
5. Wählen Sie Databasename = Mainform.IB_Main und Tablename = KUNDEN aus.




6. wechseln Sie auf die Reportseite „Seite1“ und klicken Sie auf den Button "DB-Feld einfügen"  auf der Symbolleiste;
7. wählen Sie die benötigten Felder aus und drücken Sie OK.



Jetzt enthält der Report alle Felder die Sie ausgewählt haben:

Seitenkopf						
KUNDENNR	NAME1	NAME2	STRASSE	LAND	PLZ	ORT
Hauptdaten						
{KUNDENNR}	{Table1."NAME1"}	{Table1."NAME2"}	{Table1."STRASSE"}	{Table1."LAND"}	{Table1."PLZ"}	{Table1."ORT"}

Es ist möglich automatisch generierte Reporte als Vorlage zu verwenden, welche auch Graphiken, Überschriften, Füße oder Anzahl der Seiten, usw. enthalten.

Um den Report auszuführen, drücken Sie den "Reportvorschau" Button . FastReport erstellt nun den Report und zeigt die Inhalte im Vorschaufenster an:

Vorschau				
KUNDENNR	NAME1	NAME2	STRASSE	LAND
10000	Computerstudio 300		In den Neuwiesen 3	
11000	Weiss Computer		Lindenstr. 11	
12000	Katharina Glaser	Systemhaus	Rodalber Str. 12 c	
13000	Müller & Co	Softwarebüro	Talstr. 111	
14000	Kolbenschmidt AG	Systemhaus	Heinestr. 1-5	
15000	Steiner GmbH	Sägewerk	Bachgasse 4	
16000	Cappel GmbH	Fahrzeugbau	Fürther Allee 121-1	
17000	Edgar Ziegler	Rolladenbau	Mittelstr. 101a	
18000	Edgar Kranz oHG	Hardware	Klausewitzstr. 1	
19000	Anton Winkler	Fleischgroßhandel	Schlachthof 8	
20000	Thomas Wolnitz	Bäckerei	Flemlinger Str. 12	
21000	Joseph Kaiser	Kfz-Reparaturwerks	In den Neugärten 1	
22000	Reifen Diener	Großhandel	Neusser Allee 113	

Seite 1/1

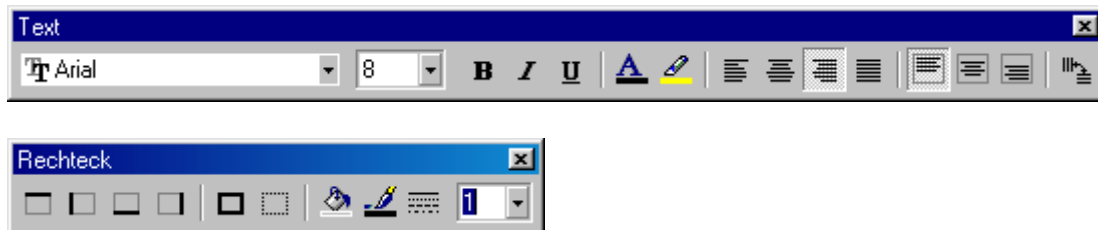
FastReport Objekte



Symbol	Name	Beschreibung
	Rechteckobjekt	Stellt ein umrahmtes Rechteck mit mehrzeiligem Text bereit. Der Text kann auch Variablen beinhalten.
	Bereich	Definiert wo der Bereichsinhalt im endgültigen Report platziert wird.
	Bild	Zeigt BMP, ICO, WMF, EMF und JPG Bildformate an. Die Quelle kann ein BLOB Feld sein.
	Unterreport	Gedacht zur Erstellung von Unterreporten. Wenn Sie dieses Objekt in ein Report einfügen wird eine neue Seite Ihrem Report hinzugefügt..
	Linie	Malt horizontale oder vertikale Linien auf den Report.
	Barcode	Zeigt Daten im Barcode Format an.
	Figur	Erlaubt das Einfügen von geometrischen Figuren (Rechteck, abgerundetes Rechteck, Ellipse, Dreieck).
	CheckBox	Häkchenbox zeigt True/False (Wahr/Falsch) zustände in Form eines Häkchens oder „X“ Zeichens.
	RTF-Objekt	Entworfen um RTF(Rich Text Format) Dokumente in den Report einzufügen. Quelle kann ein BLOB Feld sein.
	OLE	Einfügen eines OLE Objekts in den Report.
	Diagramm	Einfügen von Diagrammen in den Report. Die TeeChart Komponente muss in Delphi installiert sein damit es funktioniert.
	Kreuztabellenobjekt	Erstellt eine Kreuztabelle.

Das “Rechteckobjekt” Objekt

Dieses Objekt ist zweifellos das mächtigste und flexibelste Objekt in FastReport. Grundsätzlich stellt es Ihnen ein umrahmtes Rechteck aus mehrzeiligem Text bereit. Sie können den Rahmentyp, die Farbe und die Breite einstellen; ebenso wie die Schriftartenattribute, Textausrichtung und Schriftartenrotation (horizontal oder vertikal). Um die Objekt Attribute einzustellen, benutzen Sie die „Text“ und „Rechteck“ Symbolleiste:

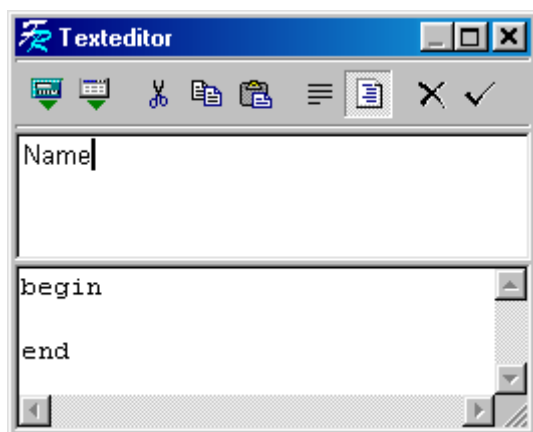


Die Inhalte eines Text Objekts bestehen grundsätzlich aus einem Memotypobjekt, welches Folgendes beinhalten kann: Text, Variablen, Datenfelder oder irgendeine Kombination von diesen Objekten.








Einige Beispiele zur Verwendung der Text Objekte:

- *Breite, cm: [Table1."Länge_cm"]* - Zeigt einigen **fixierten Text** (Breite, CM) gefolgt von einem **Datenfeld** ([Table1."Länge_cm"])
- *Breite, cm: [[Länge_inch] * 2.54]* - Zeigt einigen **fixierten Text** (Breite, CM) gefolgt von einer **Variable die eine Formel nutzt**, ([[Variable] * Wert]). Es ist wichtig die zusätzlichen eckigen Klammern bei der Benutzung von Formeln mit Variablen zu beachten
- *Breite, cm: [Table1."Länge_in" * 2.54]* - Zeigt einigen **fixierten Text** (Breite, CM) gefolgt von einem **Datenfeld in einer Formel** ([Datenfeld * Wert]). Beachten Sie das die eckigen Klammern für Datenfelder und Formeln benötigt werden.

Der schnellste Weg ein Textobjekt zu bearbeiten ist das gewünschte Rechteck auszuwählen und dann doppelklicken. Dies bringt ein Texteditor hervor:



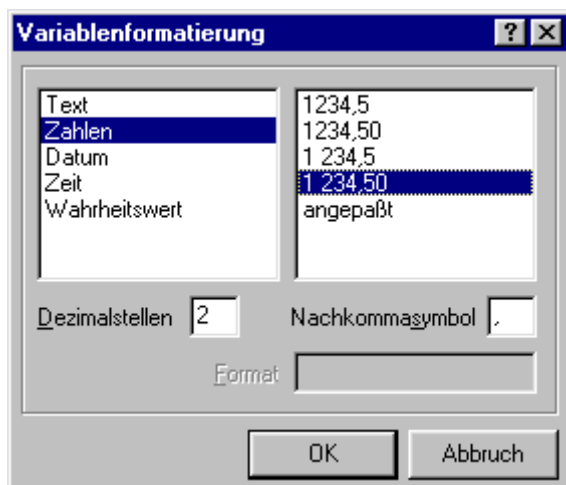
Hier ist eine kurze Beschreibung der Symbolleisten Buttons:

-  Ausdruck einfügen;
-  Datenfeld einfügen;
-  Zwischenablage Funktionen;
-  Zeilenumbruch ein/aus stellen;
-  Script ein/aus stellen;
-  Abbruch Schaltfläche;
-  OK Schaltfläche.

Sie können in dem Editor Text eingeben, Variablen, Datenfelder oder Ausdrücke einfügen. Wenn Sie im Memoeditordialog auf „Datenfeld“ oder „Ausdruck“ klicken, werden die aktuellen verfügbaren Datenfelder oder Variablen angezeigt. Sie können auch die folgenden Tasten benutzen:

- Einfg** “Variable” Button – bringt den Variablen Dialog hervor.
- Strg+Enter** “OK” Button – akzeptiert und schließt den Dialog.
- Esc** “Abbruch” Button – schließt und bricht die Bearbeitung ab.

Jeder Text kann auch seine eigene Formatierung besitzen. Sie können das Format bearbeiten indem Sie auf die „Display Format“ Eigenschaft des Text Objekts im Objekt Inspektor klicken. Formatierungsoptionen beinhalten: kein Formatieren, als einen numerischen Wert anzeigen, Datum, Zeit oder boolesche Formatierung.



Für jede Formatierungskategorie können Sie eine oder mehrere vordefinierte Formatierungsoptionen wählen oder Sie führen benutzerdefiniertes Formatieren durch (z. B., ###0.000;-###0.0;# für einen numerischen Wert). Die Formatierung wird auf die selbe Weise, wie Delphi Strings formatiert. Boolesche Werte können durch die folgende Formatierung formatiert werden: False_string;True_string (d.h. um Ja oder Nein zu zeigen, benutzen Sie Nein;Ja als Format).

Die Formatierung wird auf alle Variablen in der Memo angewandt. Wenn eine Variable nicht formatiert werden kann wird sie als Plain Text angezeigt. Wenn Sie mehrere Variablen innerhalb eines Textobjekts nutzen, aber für jede Variable verschiedene Formatierungen nutzen wollen, können Sie die Standardformatierung durch den <#> Tag gefolgt durch den Formatierungsstil aufheben. Setzen Sie den Formatierungstag und den Formatierungsstring in die Variablenklammern, d.h.:

[Variable #Format], wo Format eine der folgenden Werte ist

- x.x oder x.x oder Nyyyyy – numerisches Formatieren. x.x – Länge der Nummer/Nummer der Zahlen im Bruchstrich; yyyyy – String wie #,##0.00 (beschrieben in dem Delphi Online Hilfe System, siehe „Formatiere Strings“ Thema). Wenn der x.x or yyyyy String irgendein «.», «,», «-» Charakter enthält, wird er als dezimal/Bruch Trennzeichen benutzt.
- Dxxxxx, Txxxxx – Datum und Zeit. xxxxx – ein String wie dd.mm.yy.
- Bxxxxx;yyyyy – boolesche Formatierung. Ist der Wert False, wird der xxxxx String angezeigt; andernfalls wird der yyyyy String angezeigt.


Hier sind einige Beispiele zur Verwendung des «#» Tags:

[Table1.«N1» #9.2] [Table1.«N2» #N9-2] [Table1.«N3» #N#,##0.00] – numerische Formatierung

[Table1.«Date1» #Ddd.mm.yyyy] [Table1.«Time» #Thh:mm:ss] – Datum/Zeit Formatierung

[Table1.«Bool1» #BFalse;True] [Table1.«Bool2» #BNo;Yes] – boolesche Formatierung

Sie können den Formatierungstag nicht in Ausdrücken benutzen die mit dem Variableneditor erstellt worden sind (siehe unten).

Für Reporte wo Sie die Fontfarbe, Hintergrundfarbe, usw. von Objekten ändern wollen, die auf eine bestimmte Bedingung oder Ausdruck basieren, klicken Sie auf den  Button in der Text Formatierungssymbolleiste. Dies bringt den Dialog "Syntaxhervorhebung" vor, wo Sie die Bedingungen in die Text Box eintragen können und verschiedene Optionen auswählen können.



Zum Beispiel, nehmen Sie an es ist nötig Bestellungen von mehr als 1000 € auszuwählen, Demo Report „3- level“ ist ein Beispiel dafür. Um dies zu erreichen, wählen Sie das

Durch Rechtsklicken auf das “Text” Objekt können Sie folgende Optionen einstellen:

- **Gedehnt** – das Objekt besitzt eine variable Höhe, abhängig von der eigentlichen Anzahl der Text Strings. Sie sollten diese Option auch im Ausgangsband einschalten. Wenn dieses Band gedruckt wird, berechnet er die maximale Höhe aller Objekte mit der Gedehnt Option und gedehnt das Band selbst.
- **Zeilenumbruch** – lange Strings werden umgebrochen um mehrere Zeilen Text zu ermöglichen.
- **Autom. Größe** – das Objekt berechnet seine aktuelle Breite.
- **Nur Text** – es werden keine Variablen im Objekt verarbeitet..
- **Keine doppelten Werte** – wiederholte Werte werden nicht angezeigt.

Im Objekt Inspektor sind einige zusätzliche Eigenschaften vorhanden :

- **ZeichenAbstand** – zusätzlich Platz zwischen den Buchstaben;
- **GapX, GapY** – linker und oberer Abstand des Textes zu dem Rahmen;
- **ZeilenAbstand** – zusätzlich Platz zwischen den Zeilen des Textes.



Das “Bereich” Objekt

Wie alle anderen visuellen Report Designer ist FastReport ein bandbasierender Report Generator. Das bedeutet dass Sie Bänder auf dem Report platzieren und dann Daten in das Band setzen. Die untere Tabelle listet die Typen der Bänder welche FastReport zur Zeit unterstützt auf und wo Sie im endgültigen Report erscheinen werden.

BEACHTEN SIE: Die Bänder müssen im Reportdesigner nicht in der korrekten Anordnung erscheinen. Der Bandtyp steuert wo Sie sich im endgültigen Report befinden werden. Jedoch, Bänder in der richtigen Reihenfolge zu platzieren macht es viel einfacher sie später zu bearbeiten.

Name	Wo und wann
Report Titel	Wird am Beginn des Reports gedruckt
Report Zusammenfassung	Wird am Ende des Reports gedruckt
Seitenkopf	Wird oben auf jeder Seite gedruckt
Seitenfuß	Wird unten auf jeder Seite gedruckt
Hauptkopf	Wird am Anfang der 1. Datenebene gedruckt
Hauptdaten	Daten der 1. Datenebene– wird für jeden Hauptdatenbeleg wiederholt
Hauptfuß	Wird am Ende der 1. Datenebene gedruckt
Detaillkopf	Wird am Anfang der 2. Datenebene gedruckt
Detailldaten	Daten der 2. Datenebene – wird für jeden Detailbeleg wiederholt
Detaillfuß	Wird am Ende der 2. Datenebene gedruckt
Unterdetail Kopf	Wird am Anfang der 3. Datenebene gedruckt
Unterdetail Daten	Daten der 3. Datenebene – wird für jeden Unterdetail Beleg wiederholt
Unterdetail Fuß	Wird am Ende der 3. Datenebene gedruckt
Überlagerung	Wird durch eine untere Seiten Ebene gedruckt (nützlich um Wasserzeichen zu drucken)
Spaltenkopf	Wird am Anfang der Spalte gedruckt
Spaltenfuß	Wird am Ende der Spalte gedruckt

Gruppenkopf	Gruppentitel wird am Anfang der Gruppe gedruckt
Gruppenfuß	Wird nach der Gruppe gedruckt
Kreuzkopf Kreuzdaten Kreuzfuß	Diese Gruppen von Bändern ist entworfen worden um Tabellenvergleichsreporte zu erstellen die einen variablen Spaltenumfang auf der Seite haben.
Unterformular	Dieses Band kann zu einem anderen Band jeglichen Typs (außer Kreuz und Seiten Füßen)hinzugefügt werden. Das Unterformular wird nach dem Ausgangsband angezeigt.

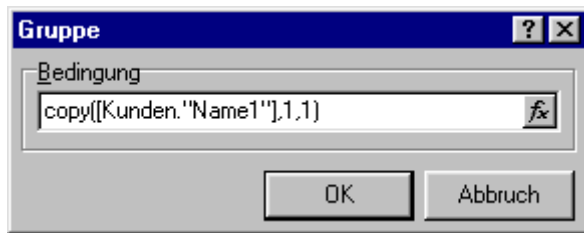
Wie Sie sehen können sind die Bänder verschieden von denen die allgemein akzeptiert werden. In dem klassischen Schema (besonders im ReportBuilder) gibt es nur ein Detaildatenband.

Abhängig vom Bandtyp zeigt FastReport den dazugehörigen Editor an. Bei Bändern welche Daten von einer Datenquelle anzeigen, (Detaildatenband, Hauptdatenband, usw.) werden Sie gefragt woher das Band seine Daten bezieht. Ihnen wird eine Liste aller verfügbaren Datenmengen angezeigt oder Sie können die „virtuelle“ Datenmengenoption wählen.



Wenn Sie die virtuelle Datenmengenoption auswählen müssen Sie die Anzahl der Einträge die in der „virtuellen Datenmenge“ beinhaltet sind, angeben. Das Band verhält sich dann als wenn die angegebene Nummer von Datenmengen wirklich existiert. Virtuelle Datenmengen können Ihnen helfen Formulare auszudrucken wo mehrere Zeilen ausgedruckt werden müssen aber wo Sie nicht manuell jede einzelne Zeile einfügen wollen. Sie können einfach eine Zeile entwerfen und sie die benötigte Anzahl durch virtuelle Datenmengen wiederholen.

Wenn Sie den Gruppenkopfbandtyp auswählen wird Ihnen der Gruppenkopfeditor vorgelegt. Dieser Editor erlaubt Ihnen den Gruppierungszustand zu definieren basierend auf den Feldern im Table oder in jedem anderen Ausdruck (d.h. gruppieren Sie die Daten auf dem 1. Buchstaben des Nachnamens basierend, wie es in Telefonbüchern verwendet wird)



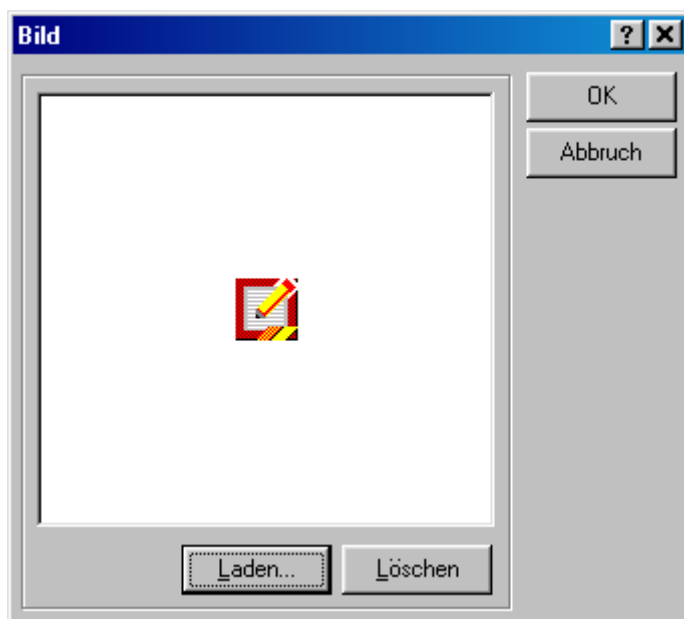
Wenn Sie auf ein Band rechts klicken, werden Sie eine Anzahl an Optionen im Kontextmenü sehen. Diese beinhalten:

- **Gedehnt** – Die Höhe der Bänder wird durch das größte Objekt im Band festgelegt. „Text“ Objekte werden, wie benötigt, gedehnt und der Zeilenumbruch sollte auch aktiviert sein. Dies erlaubt eine automatische Einstellung der Bandhöhe, wie benötigt, so das alle Daten in eine Reihe passen.
- **Unterbrochen** – Dies versucht jeden ungenutzten Platz auf der Seite zu füllen bevor eine neue Seite erstellt wird.
- **Immer neue Seite** – Dies zwingt den Bandinhalt auf der Seite auf eine neue Seite zu drucken.
- **Zeige alle Seiten** – Diese Option ist nur für Hauptkopf, Detailkopf, Unterdetail Kopf, Gruppenkopf und Kreuzkopf Bereiche verfügbar. Wenn die Anzahl der Daten unter den Köpfen eine neue Seite erzwingt werden die Köpfe auf dieser Seite miteinbezogen.



Das Bild Objekt

Das Bild Objekt wird zum Einfügen von Grafiken in einen Report benutzt. Unterstützte Bildformate einschließlich BMP/WMF/ICO. JPG und GIF Bilder können auch eingefügt werden aber sie werden nicht direkt unterstützt da sie zusätzliche Delphi Bibliotheken benötigen. Diese Bibliotheken sind frei verfügbar.



Das Bildobjekt wird benutzt um ein Bild auszuwählen, ein Bild zu löschen oder ein Bild aus einem BLOB Feld aus einer Datenbanktabelle auszuwählen. Doppelklicken Sie auf das Bildobjekt um ein Bild aus einer Datei einzufügen. Um ein Bild aus einem BLOB Feld einzufügen, wählen Sie das Bild Objekt, drücken Strg+Enter um den Texteditor aufzurufen, dann beziehen Sie den Feldnamen auf das BLOB Feld in dem Memo Editor, zum Beispiel: [Table1.GrafikFeld]. Dieser Vorgang kann auch durch Auswählen des „DB-Feld einfügen“ in den Texteditor durchgeführt werden.

Wenn in einer Datenbank verschlüsselte Bilder direkt gedruckt werden sollen benutzen Sie den OnBeforePrint Event um die Bilder zu entschlüsseln und Sie zur Laufzeit in den Report zu laden. Ein Beispiel mit ADO und SQL Servern gibt es später in der Anleitung.

Wenn auf der Festplatte gespeicherte Bilder gedruckt werden sollen benutzen Sie TfrReport.OnBeforePrint um das Bild vor dem Drucken zu laden. Ein Beispiel um Thumbnails für ein vorhandenes Verzeichnis zu drucken gibt es am Ende der Anleitung.

Das Kontext Menü des Bild Objekts erlaubt Ihnen folgende Optionen einzustellen:

- **Gedehnt** – das Bild wird gedehnt um das Rechteck zu füllen.
- **Erhalte Proportionen** – die Bildproportionen bleiben erhalten.
- **Bild zentrieren** – das Bild wird im Rechteck zentriert.
- **BlobType** – Art des Blob Streams: BMP, WMF, ICO or JPG. Dies ist erforderlich denn Delphi kann die Streamart nicht automatisch handhaben. Verwenden Sie den Objekt Inspektor um diese Eigenschaft einzustellen



Das Unterreport Objekt

Ein Unterreportobjekt wird als Platzhalter um zusätzliche Reporte zu einem bestimmten Punkt einzufügen, verwendet. Der Report der innerhalb des Unterreports gedruckt werden soll, muss auf einer gesonderten Seite innerhalb des Hauptreports platziert werden. Ein Beispiel wann das Unterreportobjekt zu benutzen ist, ist ein Diagramm unterhalb oder neben einigen Verkaufszahlen einzufügen.

Es wird eine zusätzliche Seite automatisch in den Hauptreport eingefügt wenn das Unterreportobjekt in ein Report eingefügt wird. Das Unterreport Objekt zeigt automatisch auf die gerade erzeugte Seite. Alles was der neuen Seite hinzugefügt wird, wird innerhalb des Unterreportobjekts gedruckt, anstatt im Hauptreport angezeigt zu werden.

Unterreportobjekte können unter oder neben dem Hauptreport platziert werden. Fügen Sie die Unterreport Objekte in einzelne Bänder ein, wenn mehrere Unterreport Objekte untereinander platziert werden sollen.

Es gibt einige Einschränkungen bei der Benutzung von Unterreporten:

- Es können keine Spalten benutzt werden
- Die folgenden Bänder können in einem Unterreport benutzt werden (aber nicht in dem zugehörigen Band)
- Gebrochene Bänder können nicht genutzt werden
- Es können keine Gruppen innerhalb eines Unterreports genutzt werden.



Das Linien Objekt

Das Linienobjekt wird benutzt um horizontale und vertikale Linien in einen Report einzufügen. Linien können benutzt werden um Details innerhalb eines Reports zu trennen, dies macht ihn leichter zu lesen. Die Breite und Farbe der Linie können über die Rechtecksymbolleiste eingestellt werden

Um eine Linie zu zeichnen, klicken Sie auf den Liniensymbolleistenbutton, bewegen Sie die Maus über die aktive Seite und der Cursor ändert sich in ein Stift. Bewegen Sie die Maus an den Punkt an dem die Linie anfangen soll. Drücken und halten Sie den linken Maus Button und bewegen Sie die Maus an den Punkt an dem die Linie enden soll. Lassen Sie den Mausbutton los und die Linie wird gezeichnet. Die Linie kann, wenn benötigt, zum Bearbeiten ausgewählt werden.



Das Häkchenbox Objekt

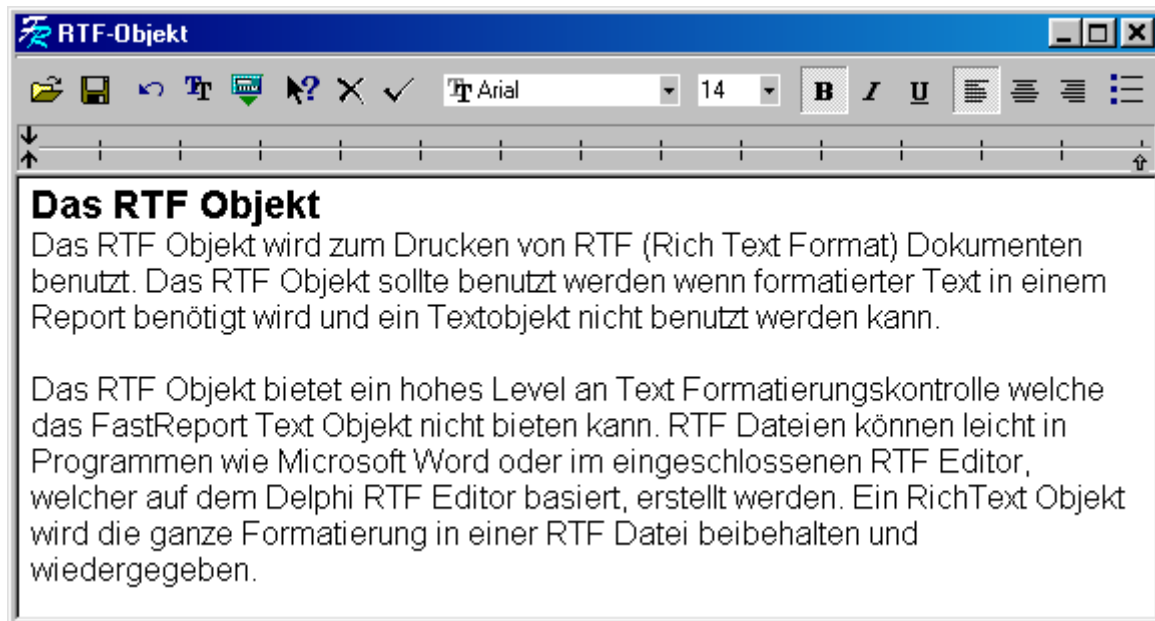
Das Häkchenbox wird benutzt, um Boolesche Daten graphisch zu zeigen und sollte nur mit einem Booleschen Variablen- oder Datenfeld verbunden werden. Wenn der Wert der Variable oder des Feldes zutreffend ist, wird das Kreuz angezeigt. Wenn der Wert falsch ist, wird nichts gezeigt.



Das RTF Objekt

Das RTF Objekt wird zum Drucken von RTF (Rich Text Format) Dokumenten benutzt. Das RTF Objekt sollte verwendet werden wenn formatierter Text in einem Report benötigt wird und ein Textobjekt nicht benutzt werden kann.

Das RTF Objekt bietet ein hohes Level an Textformatierungskontrolle welche das FastReport Textobjekt nicht bieten kann. RTF Dateien können leicht in Programmen wie Microsoft Word oder im eingeschlossenen RTF Editor, welcher auf dem Delphi RTF Editor basiert, erstellt werden. Ein RichText Objekt wird die ganze Formatierung in einer RTF Datei beibehalten und wiedergeben.



Der RTF Editor der mit FastReport geliefert wird bietet alle wesentlichen RTF Befehle am Text. Es können Variablen in den formatierten Text eingefügt werden. Diese Variablen müssen in eckigen Klammern eingeschlossen werden (wie im Textobjekt).

Das RTF Objekt kann auch benutzt werden um formatierten Text aus einem Datenbank-BLOB Feld zu drucken. Um die benötigten Felder auszuwählen drücken Sie Strg+Enter um den Texteditor aufzurufen dann geben Sie entweder das Tabellenfeld in eckigen Klammern ein, z.B.: [Table1."RichField"] oder Sie klicken auf DB-Feld einfügen und wählen die benötigte Datenquelle oder Feld.



Das OLE Objekt

Das OLE Objekt wird verwendet um ein OLE Objekt in ein Report einzufügen. Der OLE Objekteditor ermöglicht Ihnen das Einfügen eines OLE Objekts.



Das OLE Objekt kann verwendet werden um OLE Objekte die in einem Blob Feld sind, zu nutzen. Doppelklicken Sie auf das OLE Objekt, wählen Sie Einfügen und suchen Sie das erforderliche OLE Objekt.

Note 1. Die “gedehnt” Option im Kontextmenü verbessert manchmal die Ansicht von Excel Daten.



Das Diagrammobjekt

Das Diagrammobjekt wird verwendet um Diagramme in ein Report einzufügen. Dies ist besonders nützlich um Verkaufszahlen auszudrucken.

Der Art Karteireiter des Diagrammeditors wird verwendet um aus sechs Diagrammart und ihren Anzeigoptionen zu wählen:

- 3D – zeigt das Diagramm in 3 Dimensionen an;
- Einfach – sollte eingestellt sein (reserviert für ferneren Gebrauch);
- Legende anzeigen – zeigt in der Nähe des Diagramms ein Legende an;
- Achsen zeigen – zeigt die Achsen an (sollte abgeschaltet sein wenn Sie das Kuchendiagramm verwenden);
- Markierungen zeigen – Zeigt Markierungen;
- Mehrfarbig – Zeigt Diagrammwerte mit verschiedenen Farben.



Um ein Diagramm mit Datenfeldern zu verknüpfen sollten Sie die Namen zweier Textobjekte einstellen, d.h. 2 Datenfelder die bereits auf dem Report sind. Die Inhalte dieser Felder werden als Diagrammwerte und Legendenstrings verwendet. Die Legende ist normalerweise die X-Achse, der Wert die Y-Achse des Diagramms. Die Inhalte des markierten Textobjekts werden in der Memo des Diagramms angehäuft. Klicken Sie rechts auf das Diagramm, wählen Sie Bearbeiten, Daten aus und dann können Sie die Legenden/Daten auswählen, d.h. Legende könnte „memo1“ sein und die Werte könnten „memo2“ sein.



Das Diagrammobjekt ermöglicht die Erstellung von „Top10“ Diagrammen. Um dies zu tun stellen Sie die Anzahl der Topwerte auf dem Datenkarteireiter und den Legendenamen für nichteingeschlossene Werte ein (‘Sonstige’ wird normalerweise verwendet).

Der Markierungskarteireiter ermöglicht Ihnen die Art der Markierung festzulegen. “Wert” ist die Standardeinstellung. Wenn irgendeine andere Auswahl gemacht wird muss die Markierungsoption des Anzeigekarteireiter eingestellt sein.



Wenn das Textobjekt welches als Diagrammwert verwendet wird ein formatierten Wert enthält (z. B., 10 000.00 oder \$100.00) versucht das Diagrammobjekt den numerischen Wert dieses Strings zu erweitern. Alle "Nichtziffern" am Anfang und Ende des Strings werden übersprungen dann werden alle Symbole übersprungen – z.B. Zifferntrennzeichen. Wenn eine aufwendigere Formatierung verwendet wird (z.B., 10000km²) können Sie diese nicht als Diagrammwert verwenden. Nicht sichtbare, nicht formatierte Objekte sollten für diese Werte erstellt werden und mit Diagrammwerten gefüllt werden. Objekte können durch Einstellen der "Visible" Eigenschaft versteckt werden.

Diagrammwerte werden in der Memo des Diagramms angehäuft:

Header1;Header2;Header3
Value1;Value2;Value3.

Ein Diagramm kann auch durch Verwendung des Texteditors erstellt werden(Ctrl+Enter). Fügen Sie die entsprechenden Werte in die Memo des Objekts.



Das Figur Objekt

Das Figur Objekt wird verwendet um geometrische Figuren in einen Report einzufügen (d.h. Rechteck, Ellipse, Dreieck).

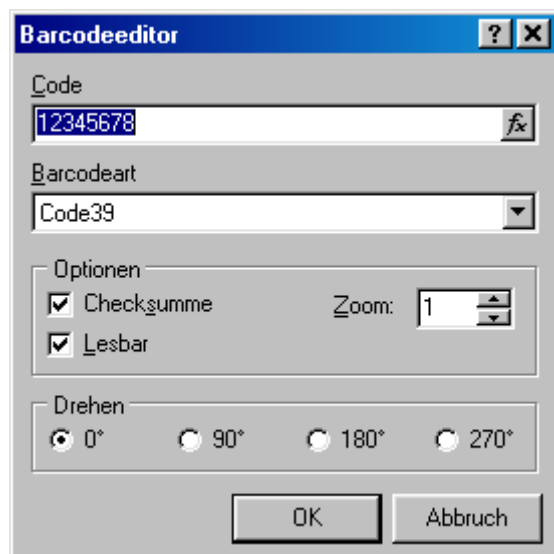


Das Barcode Objekt

Das Barcode Objekt wird verwendet um Barcodes auszudrucken. Die Breite des Barcodes wird automatisch durch die Anzahl an Daten festgelegt. Die folgenden Barcode Formate werden unterstützt:

- 2 of 5 interleaved
- Code39

- Code39 Extended
- Code128A-C
- Code93
- Code93 Extended
- MSI
- PostNet
- Codebar
- EAN8
- EAN13
- EAN128A-C
- UPC A, E0, E1, Supp2, Supp5



Hinweis: Wenn Sie weißen Platz um den Barcode benötigen, setzen Sie die FrameTyp Eigenschaft auf 15 (dies zeichnet einen Rahmen um das ganze Objekt), die FrameWidth auf 6 und die FramColor auf clWhite. Dies kann auch mit den Rechtecksymbolleistenbuttons erreicht werden.

Den Report erstellen

Bänder in FastReport



FastReport besitzt 22 Bänder. Hier ist eine Auflistung.

Name	Wo und Wann
Report Titel	Wird am Anfang des Reports gedruckt
Report Zusammenfassung	Wird am Ende des Reports gedruckt
Seitenkopf	Wird am Anfang jeder Seite gedruckt
Seitenfuß	Wird am Ende jeder Seite gedruckt
Hauptkopf	Wird am Anfang der ersten Datenstufe gedruckt
Hauptdaten	Daten der ersten Datenstufe – wird für jeden Hauptdateneintrag wiederholt
Hauptfuß	Wird am Ende der ersten Datenstufe gedruckt
Detailkopf	Wird am Anfang der zweiten Datenstufe gedruckt
Detaildaten	Daten der zweiten Datenstufe – wird für jeden Detaildateneintrag wiederholt
Detailfuß	Wird am Ende der zweiten Datenstufe gedruckt
Unterdetail Kopf	Wird am Anfang der dritten Datenstufe gedruckt
Unterdetail Daten	Daten der dritten Datenstufe – wird für jeden Unterdetaildateneintrag wiederholt
Unterdetail Fuß	Wird am Ende der dritten Datenstufe gedruckt
Überlagerung	Wird auf jeder Seite mithilfe einer unteren Seitenebene gedruckt (nützlich um Wasserzeichen zu drucken)
Spaltenkopf	Wird am Anfang der Spalte gedruckt
Spaltenfuß	Wird am Ende der Spalte gedruckt
Gruppenkopf	Gruppentitel wird am Anfang der Gruppe gedruckt
Gruppenfuß	Wird nach der Gruppe gedruckt
Kreuzkopf Kreuzdaten Kreuzfuß	Diese Bandgruppe wurde entworfen um Kreuztabellenreporte welche eine veränderbare Menge an Spalten auf der Seite haben zu erstellen.
Unterformular	Dieses Band kann zu jeder Art von Band hinzugefügt werden (außer Kreuz und Seitenfüße). Unterformular wird nach dem Vorgängerband angezeigt.

Wie Sie sehen unterscheiden sich die Bänder von dem klassischen Schema welches in

ReportBuilder verwendet wird. In diesem Schema besitzt der Report selbst ein Datensatz und es stellt die Erstdatenstufe dar. Die zweite Datenstufe wird von dem Detailband erstellt. Mit anderen Worten, das klassische Schema ermöglicht Ihnen Haupt-Detail Reporte zu erstellen. Wenn Sie mehr Detailstufen möchten, verwenden Sie ein Unterreport welches eigentlich ein Report auf einer separaten Seite ist. Unterreporte werden auch für die Haupt-Detail-Detail Reporte verwendet.

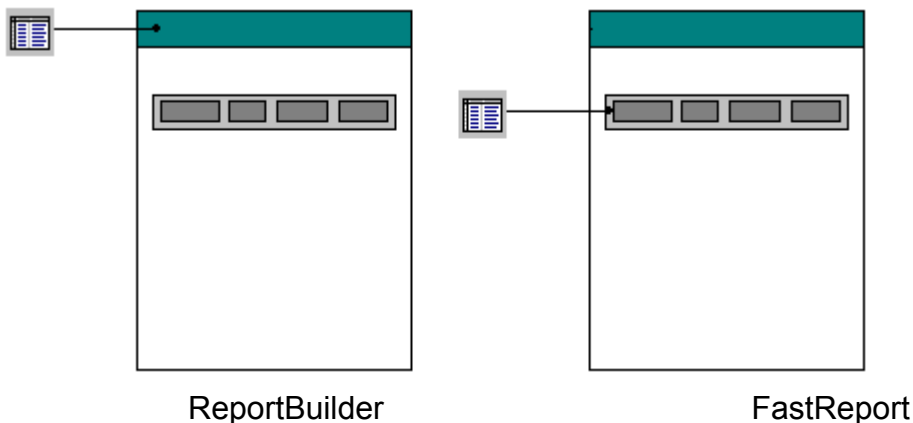
Mit dem Schema welches in FastReport verwendet wird, dürfen Sie keine Unterreporte verwenden. Wie Sie in der Tabelle sehen können Sie bis zu drei Datenstufen in einem Report verwenden (Hauptdaten des Bandes, Detaildaten und Unterdetaildaten). Jedes Datenband welches eine Datenstufe darstellt, benötigt ein zugeordnetes Verbindungsteil

„Datenquelle“, das kann eine Abfrage oder Tabelle sein (TfrIBXQuery , TfrIBXTable ). Wenn Sie mehr Stufen benötigen, können Sie Unterreporte verwenden.

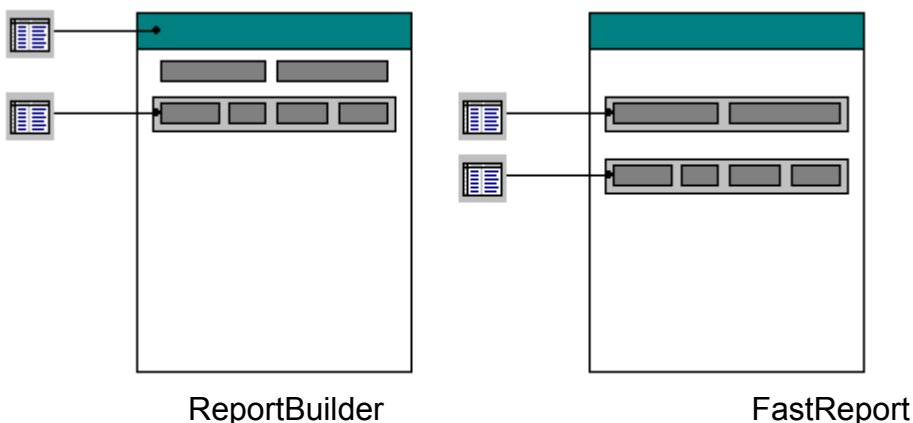
Sie können mehrere Datenbänder in einem Report haben. Z.B., Sie können zwei Hauptdatenbänder auf der selben Seite verwenden – dies ermöglicht Ihnen einfach ein “Haupt-Haupt” Report zu erstellen.

Zur Veranschaulichung wird Ihnen ein Beispiel wie einige Reporte in FastReport und ReportBuilder erstellt werden gezeigt.

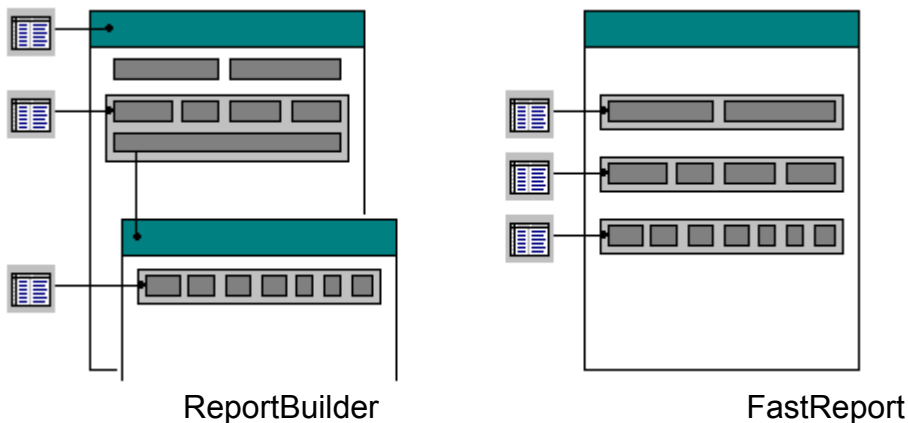
Beispiel 1. Report mit einer Stufe Daten.



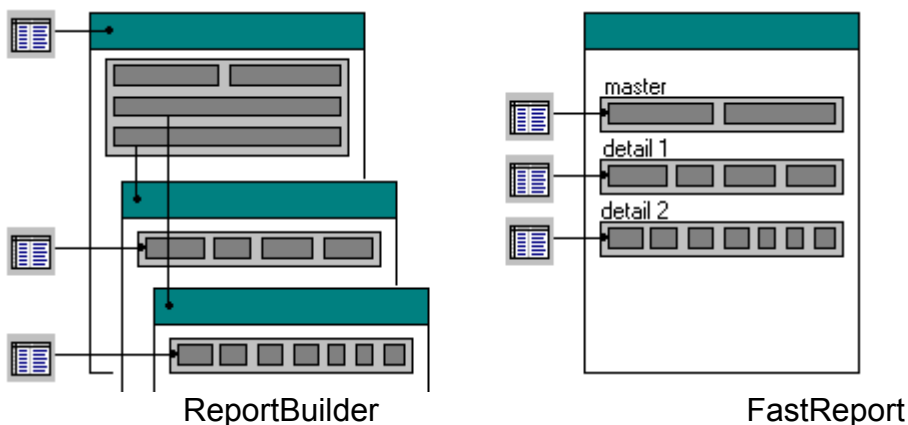
Beispiel 2. Report mit zwei Stufen Daten.



Beispiel 3. Report mit drei Stufen Daten.



Beispiel 4. Report der "Haupt-Detail-Detail" Art.



Das Schema welches in FastReport verwendet wird bietet mehr Flexibilität in der Formatierung der Reporte. Außerdem ermöglicht FastReport sich dem Objekt „Report“ Datenquelle zuzuordnen. Das Objekt TfrReport bietet einige einschlägige Eigenschaften: ReportType (rtSimple,rtMultiple) und DataSet: TfrDataSet. Wenn der ReportType = rtSimple (Vorgabe), dürfen Sie die Datenquelle nicht den Hauptbanddaten zuweisen – sie wird von der DataSet Eigenschaft oder dem TfrReport Objekt genommen. Andernfalls (ReportType = rtMultiple) würde der Report so oft erstellt wie die Anzahl der Einträge in dem Datensatz, der an der DataSet Eigenschaft angeschlossen ist. Dies ist praktisch wenn Sie mehrere Exemplare eines Reports drucken und markieren wollen (z.B. auf das erste Exemplar „Anfang“ und auf die restlichen „Zur Kenntnis“).

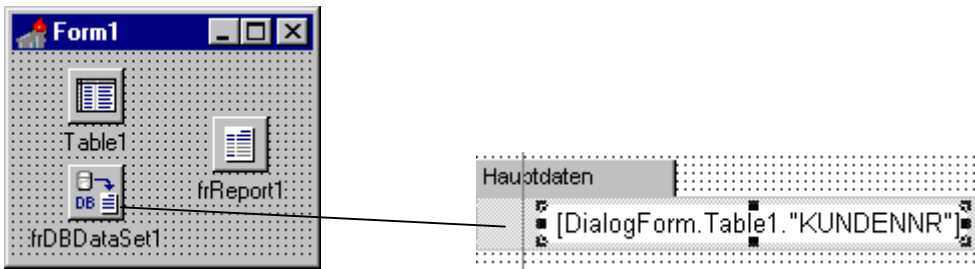
Zusätzlich zu den Hauptbändern gibt es noch zwei Nebebänder welche nicht im klassischen Schema eingebunden sind - Überlagerung und Unterformular. Diese Bänder wurden von QuickReport ausgeliehen. Überlagerung wird verwendet um Hintergrundbilder zu drucken (oder andere Objekte); Das Unterformularband kann an jedes Band außer dem Seitenfuß angeschlossen werden. Sie können auch ein Unterformular an ein anderes Unterformular anschließen. Hauptsächlich ist dies vorgesehen um mehrfach

zusammengesetzte Bänder(d.h. Bänder die aus mehreren Abschnitten bestehen) auszudrucken. Jeder Abschnitt wird von einem Unterformular gebildet und jedes Unterformular kann eine andere Höhe haben abhängig von der Höhe der Objekte in dem Abschnitt.

Hauptdaten	[Firma]
Unterformular	[Adresse]
Unterformular	[Kontakt]

Einfacher Report (Liste)

Dies ist der einfachste Report. Um ihn zu erstellen sollten Sie ein «Hauptdaten» Band auf der Seite platzieren und die erforderlichen Objekte:



Sie sollten das «Hauptdaten» Band auch noch mit einem entsprechenden Datensatz verbinden (TfrDBDataset oder TfrUserDataset Komponente). Beispiele für diesen und andere Reporte finden Sie in der Demo.

Haupt-Detail Report

Um diesen Report zu erstellen, platzieren Sie die «Hauptdaten» und «Detaildaten» Bänder auf der Seite dann platzieren Sie Objekte mit entsprechenden Daten auf diesen Bändern. Tabellen die in diesem Report verwendet werden müssen über die Haupt-Detail Beziehung verknüpft werden.

Es ist egal in welcher Reihenfolge Sie die Bänder platzieren – das «Hauptdaten» Band wird immer zuerst gedruckt. Wenn die entsprechende Detailliste leer ist wird der Haupteintrag übersprungen. Wenn Sie dies nicht möchten müssen Sie die Option «PrintIfSybsetEmpty» Drucke wenn Detailmenge leer im «Hauptdaten» Band aktivieren.

Sie können neue Hauptdaten auf einer neuen Seite drucken – aktivieren Sie einfach die «FormNewPage» Option.

Haupt-Detail-Unterdetail Report

Um diesen Report zu erstellen, platzieren Sie die «Hauptdaten», «Detaildaten» und «Unterdetaildaten» Bänder auf der Seite, dann platzieren Sie Objekte mit entsprechenden Daten auf diesen Bändern. Sonst ist es das selbe wie bei dem Haupt-Detail Report.

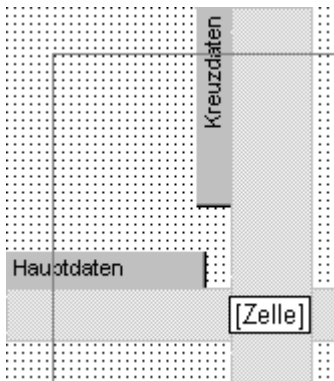
Kreuztabellen Report

Dieser Report ist beabsichtigt Tabellen mit einer variablen Anzahl von Spalten zu drucken. Während der Reporterstellung werden alle Spalten die über die Seite gehen auf die nächste

Seite gedruckt (wie in MS Excel).

Um diesen Report zu erstellen, platzieren Sie die «Hauptdaten» und «Kreuzdaten» Bänder auf der Seite. Platzieren Sie ein Objekt im Kreuz dieser Bänder.

Dieses Objekt wird als Zelle der Kreuztabelle gedruckt.

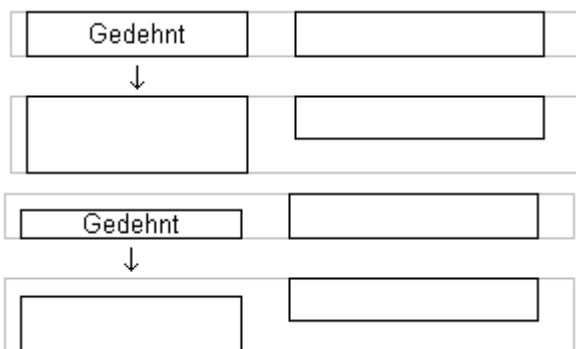


Das ist alles was Sie benötigen um Basis Reporte zu erstellen. Datenquellen sollten Sie nur "Hauptdaten" und "Kreuzdaten" zuweisen. Wenn "Kreuzdaten" mehrere normale Bänder kreuzt müssen Sie jedem Band eine Datenquelle zuweisen. Wie Sie sehen brauchen Sie mindestens zwei Datensätze die untereinander über die Haupt-Detail Beziehung verbunden sind

Diese Art von Report kann eine variable Höhe der Datenzeilen haben. Sie müssen die "gedehnt" Option im "Hauptdatenband" aktivieren. Die Kreuztabelle wird in zwei Durchläufen erstellt: maximale Zeilenhöhe wird im ersten Durchlauf festgelegt, Datenausgabe – im zweiten. Die aktuelle FastReport Version unterstützt nicht die Erstellung von Kreuztabellenreporten mit komplizierten Köpfen (wie in Report Builder). In einer der nächsten Versionen wird es möglich sein.

Dynamische Reporte

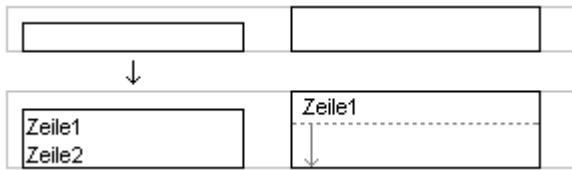
Die Bandhöhe hängt von den Objekten die in den Bändern enthalten sind ab. Die Objekte die gedehnt werden können sind die folgenden: "Rechteckobjekt" und "RTF-Objekt". Das "gedehnt" Flag muss aktiviert sein damit die Objekte gedehnt werden. Außerdem müssen Sie die Option auch im Band aktivieren indem Sie das Objekt platziert haben. Wenn Sie ein gedehntes Band drucken, liest FastReport die maximale Höhe des Objekts welches sich im Band befindet aus und setzt die Höhe gleich der maximalen Höhe der Objekte.



gedehntes Objekt und Objekt mit fester Höhe.

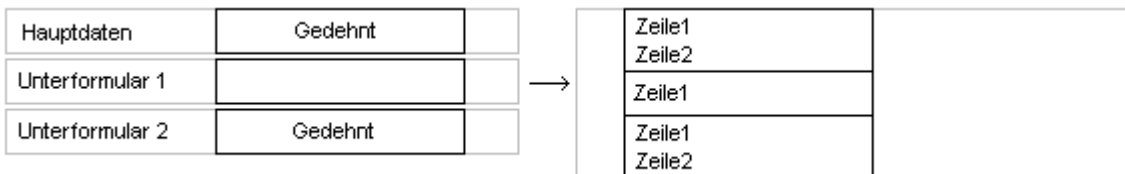
Wenn mehrere gedehnte Objekte nebeneinander stehen, würden sich die unteren Ränder

aller Objekte (trotz ihrer eigentlichen Höhe) mit dem unteren Rand des Bandes überschneiden.



Und was würden Sie tun wenn gedehnte Objekte untereinander oder nicht gedehnte Objekte unter Gedehnten platziert werden sollen?

Objekte, die untereinander platziert wurden, eines oder mehrere sind gedehnt, müssen in verschiedenen Bändern angeordnet werden.

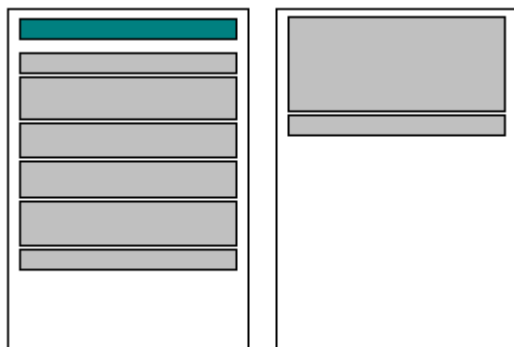


Für alle Unterformulare müssen Sie die “gedehnt” Option setzen. Die ChildBand Eigenschaft im Objekt Inspektor dient der Bindung an die Bänder. In dem oben erwähnten Beispiel müssen Sie die Hauptdaten mit Child1 (Unterformular1) verbinden und Child1 mit Child2 (Unterformular2).

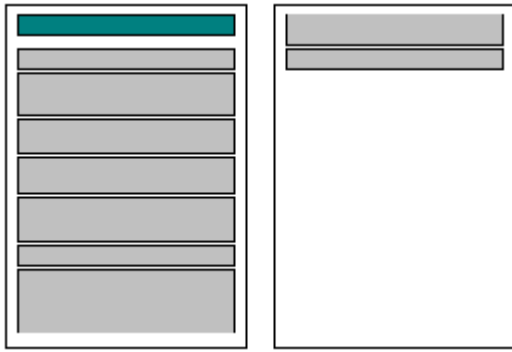
Sie sollten sich merken dass alle Objekte Rahmen besitzen. Indem Sie Rahmen zuordnen können Sie leicht eine tabellenähnliche Ansicht erlangen. Wenn ein Objekt gedehnt wird dehnt sich der Rahmen mit. Das ist sehr praktisch.

Gebrochene Bänder

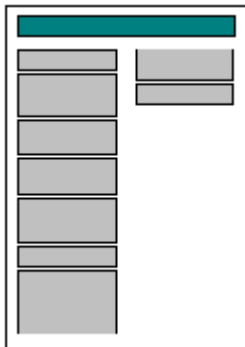
Während ein Band gedruckt wird prüft FastReport ob genug Platz auf der Seite ist und wenn es nicht genug ist erstellt er eine neue Seite und druckt das Band dorthin. Darüber hinaus wird der Platz auf der Seite nicht vergeudet, insbesondere wenn die Bandhöhe groß genug ist:



Um so viele Textzeilen wie möglich auf einer Seite zu drucken muss das Band nicht nur die „gedehnt“ Option sondern auch die „unterbrochen“ (Broken) Option aktiviert haben. Danach sollte der Report ungefähr so aussehen:



Wenn der Report mehrere Spalten hat werden die Inhalte der Objekte in die nächste Spalte übergeben:



“Rechteckobjekt” und “RTF-Objekt” Objekte können die Inhalte brechen. Alle anderen Objekte werden auf neuer Seite gedruckt, wo genug Platz verfügbar ist.

Mehrspalten Report

In einem gewöhnlichen Report fährt die Erstellung zur nächsten Seite nach der Fertigstellung der vorherigen.

In einem mehrspaltigen Report wird auf der gleichen Seite fortgeführt aber neben der ersten Spalte, d.h. in mehreren benachbarten Spalten. Die Anzahl der Spalten wird in den Seitenoptionen festgelegt. Sie können ein gewöhnlichen Report in ein mehrspaltigen umwandeln nur indem Sie die Spaltenanzahl in den Seitenoptionen einstellen. “Spaltenkopf” und “Spaltenfuß” stattet jedes Band mit einem Fuß und einem Kopf aus.


Außerdem können Sie die Spaltenanzahl für jedes Datenband einzeln in FR2.5 festlegen. Wenn wir die Column >1 Eigenschaft für ein Band setzen läuft die Datenausgabe von rechts nach links und von oben nach unten. Sie können auch die Spaltenbreite und die Spaltenlücke einstellen.

Report mit Titelseite

Es ist leicht mit FR mehrseitige Reporte vorzubereiten (d.h., die erste Seite - Titelseite, der Rest - andere Informationen). Im Wesentlichen repräsentieren mehrseitige Reporte mehrere Reporte in einem. Jede Seite kann ihren eigenen Report mit seinen eigenen Bändern beinhalten.

Seiteninhalte können auf den freien Platz der vorherigen Seite platziert werden, wenn die “PrintToPrevPage” Option (PrintToPrevPage Eigenschaft im Objekt Inspektor) aktiviert ist.

Verschachtelte Reporte (Unterreporte)

Verschachtelte Reporte sind Reporte mit einem «Unterreport» Objekt . «Unterreport» Objekte sind Verknüpfungen mit anderen Reporten die auf anderen Seiten platziert sind. Während der Reporterstellung wird anstatt des «Unterreport» Objekts der passende Report gedruckt. Wenn Sie ein Unterreport Objekt einfügen wird eine neue Seite hinzugefügt.

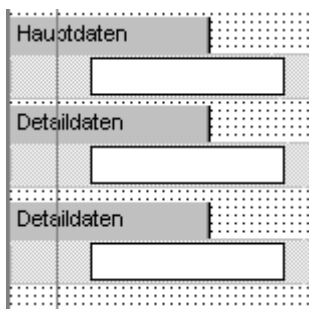
Unterreport Objekte können nebeneinander platziert werden. Wenn Sie Unterreport Objekte eins unter dem anderen platzieren wollen, verteilen Sie die Objekte auf mehrere Bänder, (z.B., verwenden Sie das Unterformular). Das selbe Unterreport Objekt kann durch kopieren aus mehreren Bänder gedruckt werden.

Es gibt einige Einschränkungen in Bezug auf Unterreporte:

- Verwenden Sie in Unterreporten keine Spalten.
- Verwenden Sie keine Reporttitel, Reportzusammenfassung, Seitenkopf, Seitenfuß, SpalteXXX Bänder.
- Verwenden Sie keine gebrochenen Bänder.
- Verwenden Sie keine Gruppen.

Haupt-Detail-Detail Report

Um diese Art von Report zu erstellen platzieren Sie «Hauptdaten» und zwei «Detaildaten» Bänder auf der Seite. Verknüpfen Sie die Bänder mit den entsprechenden Datensätzen.



Sie können auch «Haupt-Detail-Detail-Detail», «Haupt-Haupt», «Haupt-Detail-Unterdetail-Unterdetail» Reporte usw. erstellen, bis auf "Haupt-Detail- Haupt-Detail" Reporte. Solch ein Report muss mehrseitig sein – Sie müssen die Haupt-Detail Sätze auf separate Seiten legen.

Report mit BLOb Feldern

Wenn Sie BLOb Daten in einem Objekt anzeigen wollen fügen Sie eine Verknüpfung des entsprechenden Feldes in die Memo des Objekts ein (oder tragen Sie in die "DataField" Eigenschaft im Objekt Inspektor ein). Wenn Sie BLOb Daten aus einer nicht-DB Quelle anzeigen wollen, müssen Sie dies im OnBeforePrint Event Handler tun. So zum Beispiel kann ein Bild geladen werden: `Picture1.LoadFromFile('C:\Bitmap.BMP');`

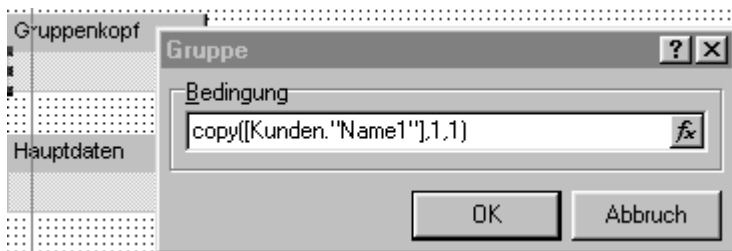
Report ohne Bänder

Wenn Sie ein Report drucken wollen, welcher Daten aus nur einem Eintrag enthält können Sie keine Bänder verwenden. Alle Objekte werden gedruckt wie sie platziert wurden.

Report mit Gruppen

Gruppen werden verwendet um Daten nach einigen Merkmalen zu gruppieren. Sie können jeden FR Ausdruck als Gruppenbedingung verwenden (normalerweise verwendet man Ausdrücke die auf DB-Feldern basieren). Wenn sich der Ausdruck ändert bildet FR eine neue Gruppe.

Um diese Art von Report zu erstellen, platzieren Sie die «Gruppenkopf» und «Hauptdaten» Bänder auf der Seite. Ordnen Sie dem «Hauptdaten» Band ein entsprechenden Datensatz zu. Geben Sie im Editor des «Gruppenkopf» Bandes den Gruppenausdruck ein. Zum Beispiel, um die Kundenliste die nach dem ersten Buchstaben des Kundennamens geordnet ist auszudrucken, geben Sie folgenden Ausdruck ein: `copy([Kunden."Name1"],1,1)`. `[Kunden."Name1"]` ist Datenbankfeld.



Reportformular

Gruppenkopf	[A]
Hauptdaten	[Firma]

Vorbereiteter Report

V		
Vashon Ventures	743	↓
VIP Divers Club	32	M
W		
Waterspout SCUBA Center	7865	

Bei der Verwendung von Gruppen gibt es eine Einschränkung: in Unterreporten können Sie keine Gruppen verwenden.

Hinweis: Der Datensatz der für das «Hauptdaten» Band verwendet wird, sollte nach der Gruppenbedingung geordnet werden. Dies können Sie mithilfe von Abfragen und dem ORDER BY Statement tun.

Report mit Diagrammen

Lassen Sie uns annehmen wir haben die folgende Tabelle:

Action Club	\$1000
Action Diver Supply	\$12000
Adventure Undersea	\$5000

Um ein Report zu erstellen der Diagramme anzeigt ist es nötig ein Datenband zu haben d.h. Hauptdaten an ein entsprechende Abfrage oder Tabelle angeschlossen. In unserem Fall werden zwei Objekte mit "Name" und "Amount" Feldern darauf platziert. Diese Felder werden z.B., Memo1 und Memo2 genannt. Platzieren Sie das Diagrammobjekt in dem Band welches angezeigt wird nachdem die ganzen Hauptdaten angezeigt wurde – in unserem Fall können Sie das Report Zusammenfassungsband verwenden.

The screenshot shows a Windows-style dialog box titled "Diagramm Objekt". It has three tabs: "Art", "Daten", and "Markierungen". The "Daten" tab is selected. Inside the dialog, there are two main sections. The first section, labeled "Objektname", contains two text input fields: "für Legenden" with the value "Memo1" and "für Werte" with the value "Memo2". The second section, labeled "Top10 Gruppe", contains two sub-sections. The first sub-section, "Zeige Top", has a text input field with the value "10" followed by the text "Werte". The second sub-section, "Bezeichnung", has a text input field with the value "Sonstige". At the bottom of the dialog are two buttons: "OK" and "Abbruch".

Im Diagramm Objekteditor wählen Sie die Art des Diagramms und die Optionen, auf dem Datenkarteireiter geben Sie den Namen der Memo ein der für die Legende verwendet wird (in unserem Fall Memo1), dann geben Sie den Namen der Memo ein welche die numerischen Daten enthält die Sie als Diagramm anzeigen wollen (Memo2).

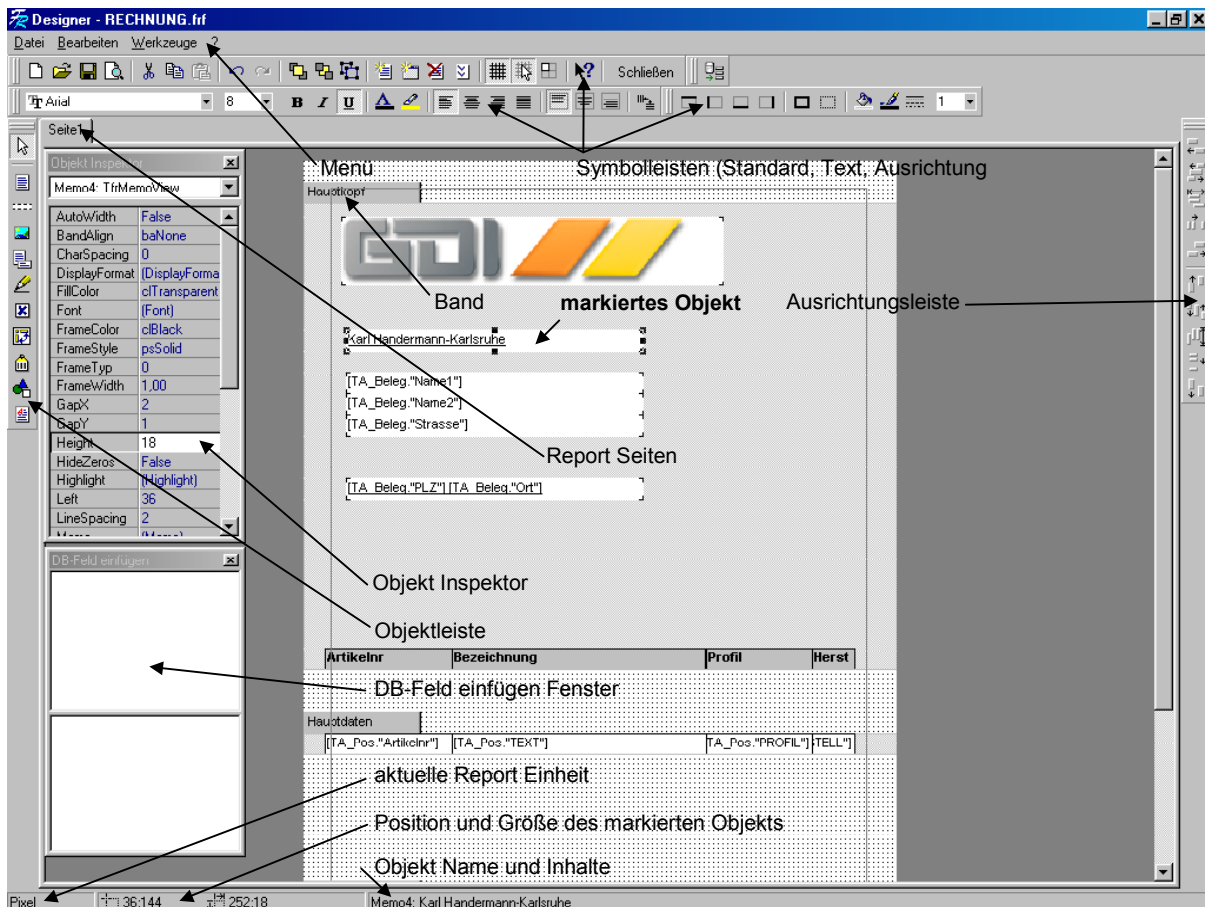
Wenn Sie eine große Anzahl von Eintragungen haben, wollen Sie vielleicht den "top ten" Gruppenwert auf einen kleineren Wert setzen. Dies schränkt die Anzahl der Elemente die im Diagramm angezeigt werden ein. Stellen Sie die Größe des Diagrammrechtecks vernünftig ein um es korrekt darzustellen. Machen Sie eine Vorschau des Reports und schauen Sie sich das Resultat an. Dies ist eine kleine Zusammenfassung, um ein Diagramm darzustellen müssen Sie Variablen verwenden und in manchen Fällen kommt es auf die Art der Werte an die von den grundlegenden Daten kommen.

Der Designer

FastReport wird mit seinem eigenen Reportdesigner geliefert.

Der Designer erlaubt einen schnellen und leichten Zugriff auf das Reportdesign und auch auf die Livevorschau des Reports während Sie in bearbeiten.

Der Designer enthält andockbare Symbolleisten (Toolbars), die nach Ihren besonderen Bedürfnissen bewegt und geändert werden können. Die Position und Sichtbarkeit aller Panels werden in der Windows Registry gespeichert wenn der Designer geschlossen wird und ihre Position und Sichtbarkeit werden beim nächsten Mal wenn Sie den Designer starten, wiederhergestellt.



Verwendung der Tastatur

- Pfeiltasten – bewege zum nächsten Objekt .
- Strg + Pfeiltasten – bewegt ausgewählte Objekte in die Richtung des Pfeils
- Shift + Pfeiltasten – vergrößert oder verkleinert die ausgewählten Objekte in Pfeilrichtung.
- Enter – bringt den Editor für das ausgewählte Objekt hervor.
- Entf - löscht gewähltes Objekt.
- Einfg - zeigt "DB-Feld einfügen" Symbolleiste an.
- Strg + Enter – bringt den Memoeditor für das ausgewählte Objekt hervor.
- Strg + 1..9 – stellt die Rahmendicke für das ausgewählte Objekt ein.
- Strg + Z – letzte Aktion rückgängig.
- Strg + Y – wiederholt Aktion.
- Strg + G – stellt Raster ein/aus.
- Strg + B, Strg + I, Strg + U - stellt fett/kursiv/unterstrichen Schriftartattribute ein.
- Strg + F – stellt Objekturnrahmung aus.
- Strg + D - stellt Objekturnrahmung an.
- Strg + X – ausschneiden in die Zwischenablage
- Strg + V – einfügen aus der Zwischenablage.
- Strg + C – kopiert in die Zwischenablage.
- Strg + A – wählt alle Objekte auf der Seite aus.
- Strg + N – erstellt neuen leeren Report.
- Strg + O – öffnet Reportdatei.
- Strg + S – speichert Reportdatei.
- Strg + P – Reportvorschau.

Verwendung der Maus

- Linksklick - markiert ein Objekt im Seitenfenster; markiert in der visuellen Komponentenpalette ein Objekt zum Einfügen – nach der Auswahl der Linksklick über dem Seiten Fenster fügt ein neues Objekt ein.
- Rechtsklick – bringt das Kontext Menü des ausgewählten Objekts hervor.
- Doppelklick – bringt den Standardeditor des ausgewählten Objekts hervor. Doppelklick auf eine freie Stelle auf der Seite bringt den Seitenoptionsdialog hervor, wo Sie die Seitenoptionen wählen können z.B. Seitenrand, Größe, usw.
- Shift + Linksklick – stellt die Auswahl des Objekts ein.
- Strg + Linksklick – zeichnet ein Auswahlrechteck. Alle Objekte die in dieses Rechteck passen werden ausgewählt nachdem Sie die Maustaste loslassen.
- Um mehrere Objekte zu skalieren, ziehen Sie das rote Quadrat auf der rechten unteren Ecke der Objektgruppe.

Report Optionen

Um die Report Optionen einzustellen, wählen Sie “Datei|Report Optionen...” aus dem Designer Menü.

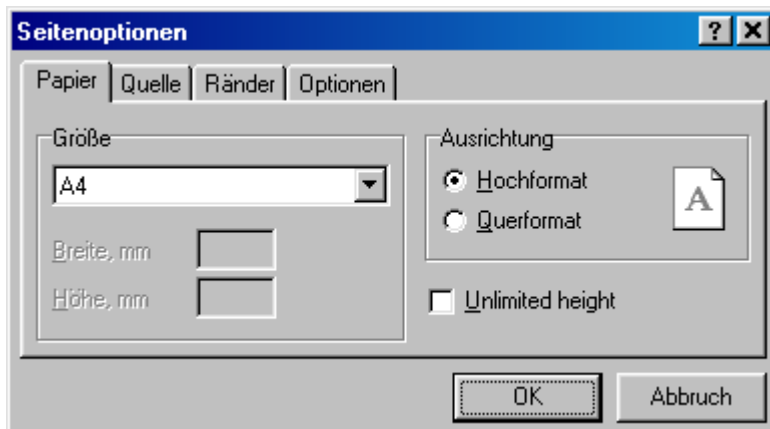


- Die Liste in der Dialogbox listet alle verfügbaren Drucker des Systems auf. Es besteht nur die Möglichkeit mit dem Report im Designermodus zu arbeiten und die Vorschau des Reports zu betrachten.
- Sie können aber auch die Standarddrucken Option zum Entwerfen eines Reports für einen Drucker benutzen, der nicht auf ihrem PC installiert ist (z. B., Sie möchten einen Report für A3 Papier entwerfen, aber ihr Drucker (A4) erlaubt es nicht.)
- Wenn die “Auswählen beim Report laden” Option ausgewählt ist, wird die Druckerinformation in dem Report gespeichert und der Drucker wird automatisch ausgewählt wenn der Report wieder geladen wird. Wird dieser Drucker nicht im System gefunden dann wird der Standardwindowsdrucker ausgewählt.
- Die “2 Durchläufe pro Report” Option muss angewählt sein wenn Sie die „gesamte Seiten“ Funktion in Ihrem Report benutzen möchten, d.h. drucke „Seite xx von yy“. Wenn Sie die „gesamte Seiten“ Funktion benutzen, aber vergessen haben die Option einzustellen, werden Sie eine 0 anstatt die Anzahl der Seiten in ihrem Report bekommen.
- Noch ein Vorteil Reporte „2 Durchläufe pro Report“ zu erstellen, ist einige Berechnungen während des ersten Durchlaufs zu erledigen und die Resultate im letzten Durchgang anzuzeigen. Eine dieser Berechnungen – zeige Gruppengesamtbetrag im Gruppenkopf – finden Sie bei den Demoreporten.
- Das Seitenfenster in dem Reportdesigner zeigt die verfügbare druckbare Fläche für die ausgewählte Seitengröße und den ausgewählten Drucker, nach der Auswahl eines Druckers, an.

Seitenoptionen

Um die Seitenoptionen für die aktuelle Seite des Reports einzustellen, wählen Sie „Datei|Seitenoptionen...“ im Designer Menü oder doppelklicken Sie auf eine leere Stelle auf der Seite. Der Dialog hat vier Karteireiter.

Der “Papier” Karteireiter



- Sie können das Papierformat aus der drop-down Liste von Papierformaten die den aktuellen Drucker unterstützt, wählen.
- Wenn der aktuelle Drucker benutzerdefinierte Papiergrößen unterstützt und Sie wählen „benutzerdefiniert“ aus der Liste aus, werden Sie in der Lage sein die Breite und Höhe der benutzerdefinierten Größe einzugeben. Jede andere Auswahl benutzt Vorgabegrößen.
- Sie können auch die Ausrichtung für die Seite auswählen und die „Unlimited Height“ Option einstellen – dies vergrößert die Höhe der Seite (falls Sie mehr Platz benötigen um mehr Bänder auf der Seite zu platzieren).

Beachten Sie Nicht alle Drucker unterstützen benutzerdefinierte Papiergrößen, (z.B., Druckertreiber „HP LaserJet 6L“ unterstützt keine Seitendimensionen kleiner als 76 * 127 mm; Drucker Treiber „HP LaserJet 4L“ unterstützt überhaupt keine benutzerdefinierten Seitengrößen).

Der “Quelle” Karteireiter

Hier können Sie Papierschacht auswählen für diese Reportseite.



Der “Ränder” Karteireiter



- Auf der “Ränder” Seite des Dialogs können Sie auswählen ob Sie Ränder und die Größe der Ränder benutzen wollen. Wenn die „An Druckbereich anpassen“ Option ausgewählt ist, wird kein druckbarer Rand im Designerseitenfenster angezeigt und alle Seiteninhalte werden korrekt auf jedem Drucker ausgedruckt.
- Wenn Sie diese Option nicht auswählen und alle Ränder Einstellungen auf 0 sind, wird der Seitenrand die maximale druckbare Fläche für den ausgewählten Drucker widerspiegeln. Sie könnten etwas Kürzung feststellen wenn Sie zwischen den Reports von einem Drucker zu dem anderen wechseln, besonders zwischen Tintenstrahl und Nadeldrucker. Tintenstrahldrucker haben normalerweise eine kleinere bedruckbare Fläche als ein Nadeldrucker auf Grund der Art wie das Papier eingezogen wird.
- Wenn Seitenränder auf ungleich Null eingestellt werden, wird ein Rand im Report Designerseitenfenster angezeigt (im Hash Linien Stil). Wenn Sie ein Nadeldrucker verwenden, achten Sie auf die bedruckbare Fläche: manche Nadeldrucker drucken nicht wenn das zu druckende Objekt den druckbaren Bereich überschreitet, andere erzeugen Seiten mit dem Überlauf.

Der “Optionen” Karteireiter



Auf der "Optionen" Seite des Dialogs können Sie viele andere Optionen für die Seite einstellen. Sie können die Anzahl der Spalten über die Seitenbreite und den Abstand zwischen ihnen einstellen. Wenn Sie „Auf vorherige Seite drucken“ wählen, wird eine neue Seite auf dem unbenutzten Platz der vorherigen Seite gedruckt.

Designer Optionen

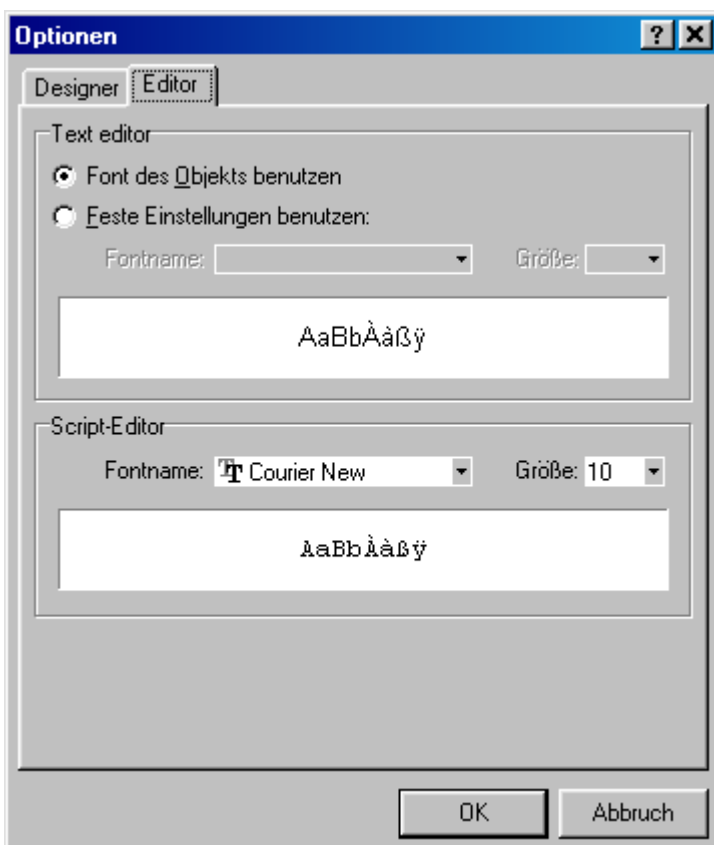
Um die Standard Optionen für den Report Designer einzustellen, wählen Sie den "Werkzeuge|Optionen..." Menübefehl.



- Hier können Sie die Gitter Größe, die Report Einheiten: Pixel, Millimeter und Inches einstellen. Gittergröße 18 Pixel entspricht 5mm.
- Sie können auch kontrollieren wie Objekte gezeichnet werden wenn Sie bewegt oder in

der Größe angepasst werden.

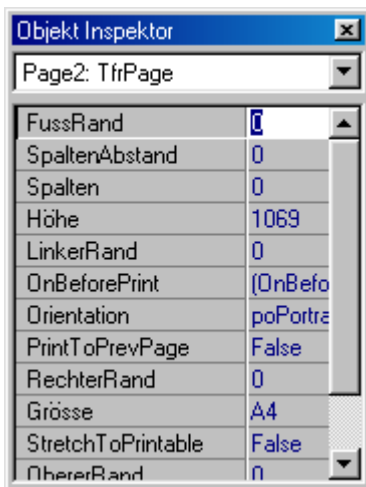
- Die “Seitenposition” Gruppe ermöglicht Ihnen die Seitenposition zu wählen (dieses wird vom Objekt Inspektor benötigt).
- Wenn “bunte Schaltflächen” ausgeschaltet ist, werden alle Buttons schwarz und weiß gefärbt.
- Die “Bearbeiten nach Einfügen” Option legt die Standard Aktion nach dem Einfügen von Objekten fest, ob der Standard Editor für Objekte nach dem Einfügen erscheint oder nicht. Wenn Sie eine große Anzahl von Rechtecken einfügen, stellen Sie diese Option aus.
- “Zeige Bereichsüberschriften” ermöglicht ihnen den Band Titel (Karteireiter) auszuschalten wenn Sie mehr Platz auf der Seite zum Entwerfen benötigen.
- “Eigenschaften in deutsch” deutsche Objekt Inspektor Eigenschaften. Da die Eigenschaften von Objekten in Befehlsinterpreter in English sind, sollte man gleich die Eigenschaften in English sich merken.
- “am Gitter ausrichten” rastert Objekte, wenn Sie bewegt werden, am nächsten Rasterpunkt ein.



- Auf dem “Editor” Karteireiter können Sie die Schriftart des Editor Fensters festlegen.

Der Objekt Inspektor

Der Objekt Inspektor ermöglicht Ihnen die Objekt Eigenschaften zu verändern.

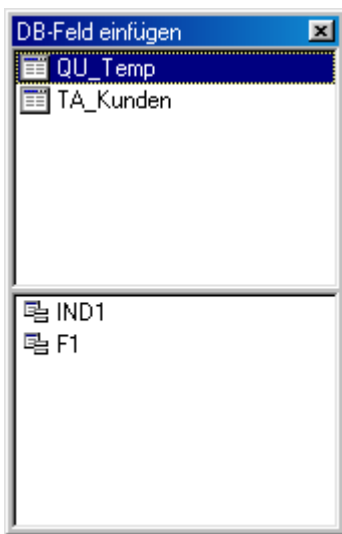


- Der Inspektor arbeitet auf die selbe Art und Weise wie der Delphi Objekt Inspektor. Wie die anderen Symbolleisten kann er angezeigt oder versteckt werden.
- Um den Objekt Inspektor anzuzeigen, wählen Sie das „Werkzeuge|Symbolleisten|Objekt Inspektor...“ Menüelement oder F11. Um den Objekt Inspektor zu verkleinern, doppelklicken Sie auf seine Titelleiste; Auf die Titelleiste doppelklicken führt dazu dass er wieder vergrößert wird.

Das „DB-Feld einfügen“ Fenster

Sie können ein DB Feld durch dieses Dialogfenster schnell in einen Report einfügen. Sie können es aus dem Menü „Bearbeiten|Symbolleiste|DB-Feld einfügen“ aufrufen oder durch Drücken der „Einfügen“ Taste.

- Das Fenster beinhaltet 2 Listen: Table (Abfrage) Liste am Anfang der Liste und Felder Liste am Ende.
- Das Feld wird mittels drag&drop eingefügt.
- Sie müssen ein Feld mit der Maus aus der unteren Liste auswählen, die Maustaste halten und es in die Reportseite ziehen. Wenn Sie die Maus loslassen, sollte ein neues Objekt „Text“ erscheinen mit einem Link auf das markierte DB Feld.



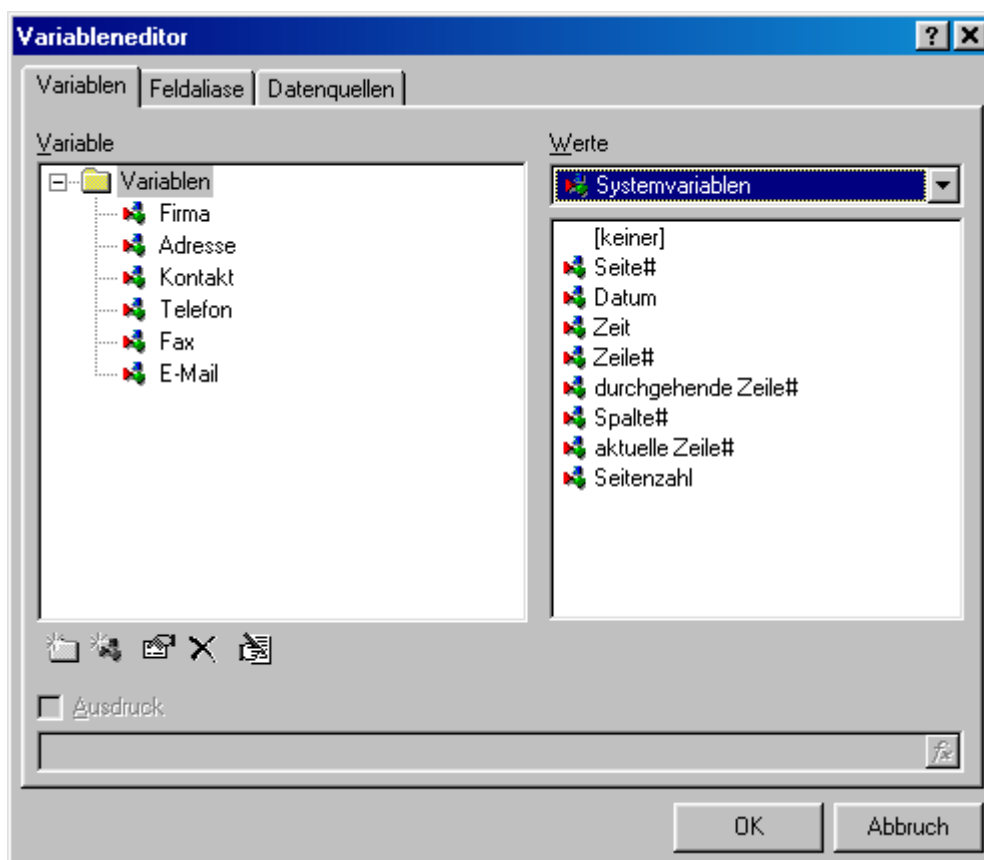
- Wenn der Dialog oft benutzt wird, ist es nützlich ihn unter oder über den Objekt Inspektor zu setzen.
- In diesem Fall doppelklicken Sie auf den Dialogfenstertitel und das Fenster wird minimiert und das Inspektor Fenster wird maximiert;
- Ein weiterer Doppelklick kehrt es um, so wird der Standardfensterstatus wiederhergestellt und die Größe des Inspektor Fensters wird reduziert.

Der „Variableneditor“

Der „Variableneditor“ Fenster kann aus dem Menü „Datei|Variablenliste...“ ausgewählt werden.

- Das Fenster sieht wie ein Notizbuch mit drei Karteireiter aus: „Variablen“, „Feldalias“ und „Datenquellen für die Bänder“.
- Der Variableneditor wird mit dem Report in der Datei gespeichert aber es gibt die Möglichkeit ihn in einer gesonderten Datei mit .FRD Endung zu speichern und wenn es nötig ist Daten zu dem existierenden Report zu lesen.
- Um dies zu tun, benutzen Sie die Befehle aus dem Designer Menü „Datei|Öffnen“ und „Datei|Speichern unter“ und wählen Sie den Datei Typ „FastReport Dictionary“ in dem Fenster Öffnen und Speichern.






„Variablen“ Karteireiter

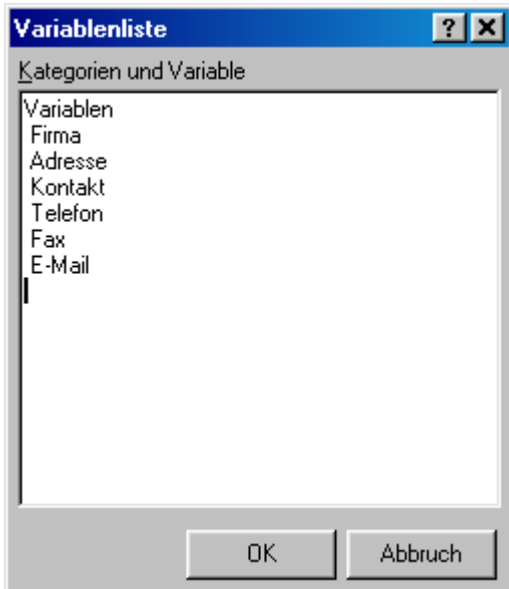


Dieser Karteireiter wird benutzt um mit einer Variablenliste zu arbeiten. Die Variablen Liste ist im rechten Teil des Fensters.


- Wie auf dem Bild gezeigt, enthält die Listen Struktur 2 Ebenen.
- Sie besteht aus Kategorien, es dürfen eine oder mehrere Variablen in jeder Kategorie sein.
- Die Kategorien werden nur zur visuellen Variablenanordnung verwendet, aber Sie werden nicht in den Report eingefügt.

Wenn Sie einen neuen Report erstellen ist die Liste leer, Sie können sie füllen indem Sie die Buttons unter der Liste verwenden:

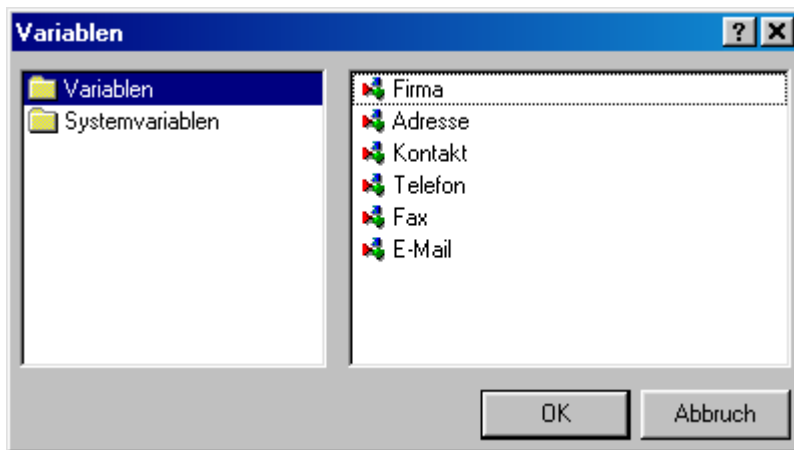
-  - fügt eine neue Kategorie ein
-  - fügt eine neue Variable zur aktuellen Kategorie hinzu. Diese Aktionen können Sie mit der Tastatur ausführen: „Einfügen“ Taste fügt eine neue Variable hinzu, „Strg+Einfügen“ fügt eine neue Kategorie hinzu.
-  - bearbeitet den Namen einer Variable oder Kategorie („Enter“ Taste),
-  - löscht eine Variable oder Kategorie („Entfernen“ Taste).
-  - ruft den Variablen Listen Editor auf, wo er als String Liste angezeigt wird. Hier können Sie mehrere Variablen aus der Zwischenablage einfügen und Variablen von einer Kategorie in die andere verschieben.



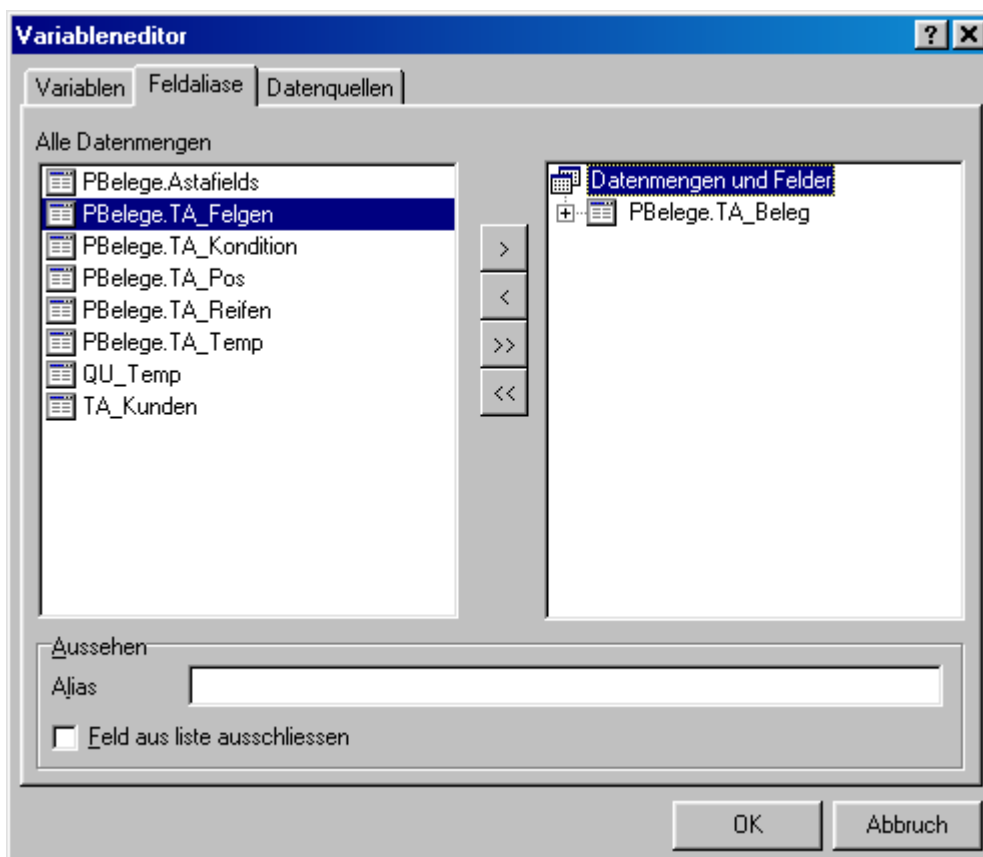
- Nachdem die Variablen Liste definiert ist, sollten Sie ein Wert für jede Variable festlegen.
- Um dies zu tun wählen Sie eine Variable links aus der Liste aus und ein Wert aus der rechten Liste.
- Alle Daten Quellen die in der rechten Liste angezeigt werden sind zu diesem Zeitpunkt abrufbar, so sind die Namen des Feldes auch abrufbar.
- Sie können den Wert „System Variablen“ aus der rechten List Box wählen und dann gibt es die Möglichkeit einer der folgenden Werte der Variablen zuzuweisen:
 - Seite – Nummer der aktuellen Seite, ist das selbe wie die Page# Funktion (siehe die Beschreibung interner Funktionen)
 - Datum – Druckdatum vom Beginn des Reports, entspricht Datum;
 - Zeit – Druckzeit vom Beginn des Reports, entspricht Zeit;
 - String, LineThrough#, Spalte, CurrentLine#, Gesamte Seiten – siehe die Beschreibung interner Funktionen

Ein Ausdruck kann ein Wert einer Variable sein – kreuzen Sie die “Wählen Sie eine Variable” Checkbox an und kreuzen Sie die „Ausdruck“ Checkbox am Ende des Fensters an .

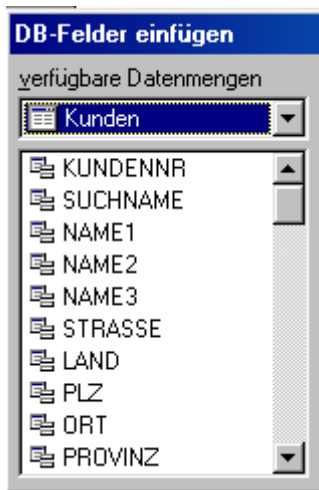
Das Variableneinfügefenster für hervorgehobene Beispiele wird wie hier gezeigt:



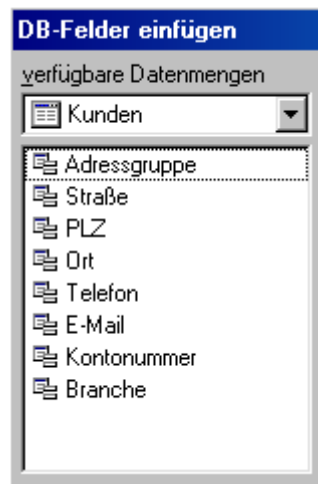
"Feldalias" Karteireiter



- Auf der linken Seite gibt es eine Liste aller verfügbaren Datensätze, Tabellen und Anfragen, die auf allen Formularen des Programms vorhanden sind.
- Dieses Dialogfenster wird verwendet um nicht benötigte Daten aus der Liste zu entfernen und um sie mit ausführlicheren Namen zu benennen (Aliase).
- Um eine Datenmenge zu verwenden sollten Sie ihn von der linken Liste in die rechte ziehen. Sie können die drag&drop Methode benutzen oder auf die benötigte Datenmenge doppelklicken oder die Buttons im mittleren Teil des Fensters benutzen.
- Um das Feld aus der Liste zu entfernen, wählen Sie eine Datenmenge und ihre Felder und kreuzen Sie die „Feld aus Liste ausschließen“ Box an. Der Befehl ist auch durch Drücken der „Space“ Taste verfügbar.
- Um ein Alias einzurichten wählen Sie ein Element und geben Sie einen neuen Namen im „Alias“ Feld ein. Löschen Sie alle Daten des Feldes, wenn kein Alias benötigt wird. Dieser Befehl ist auch über die „Enter“ Taste erreichbar.
- Wie Sie auf dem Bild sehen, macht es die Reporterstellung leichter wenn Sie Aliase für Datenquellen und seine Felder erstellen. Vergleichen Sie das Dialogfenster des Einfügen Feldes mit dem Report bevor und nachdem Sie Aliase angelegt haben:

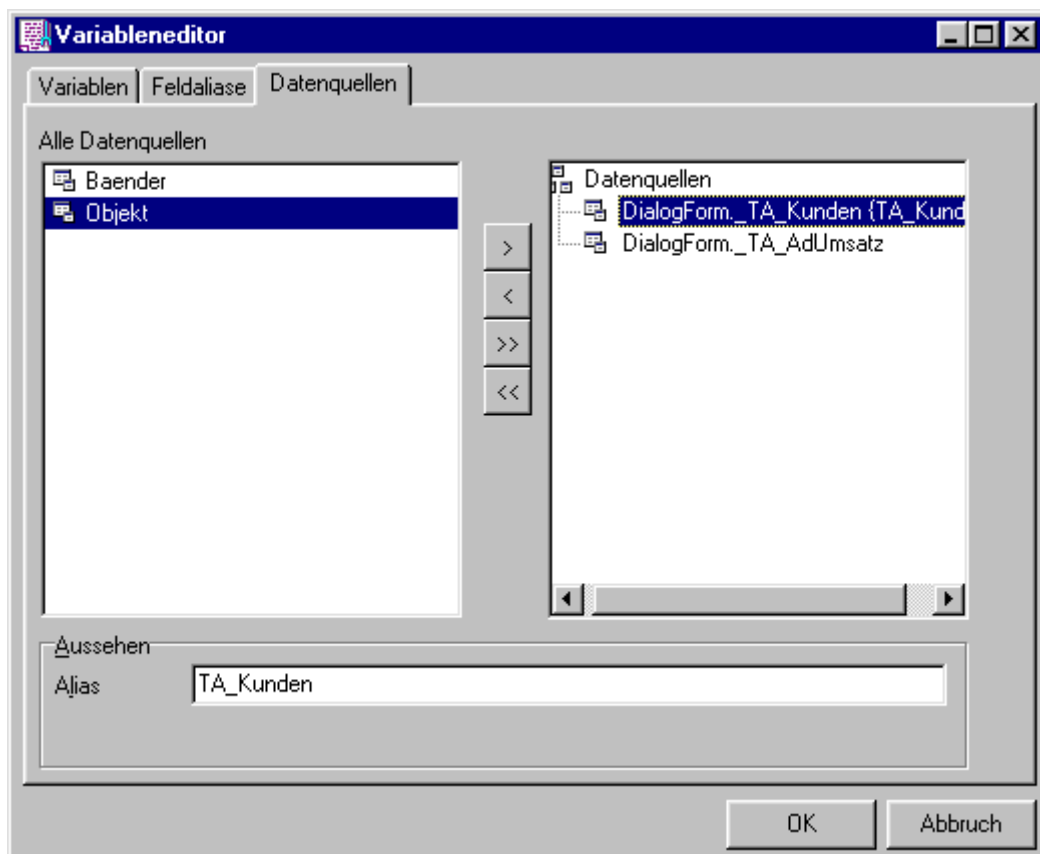


Ohne Alias



Mit Alias


"Datenquellen" Karteireiter

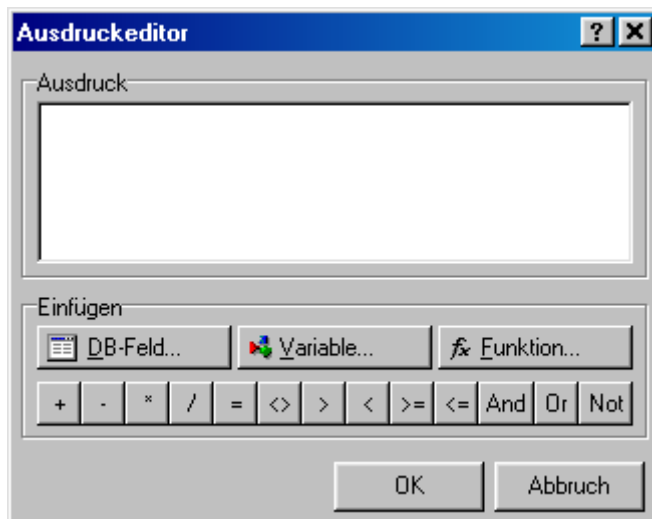


Dieser Karteireiter zeigt eine Liste aller verfügbaren Datenquellen an – Komponenten TfrDBDataSet, TfrUserDataSet, die auf allen Formularen des Projekts sind. Wie im vorherigen Dialog können Sie ausführlichere Namen verwenden (Aliase). Das Resultat wird im Dialog mit der Wahl der Datenquelle für das Datenband angezeigt:



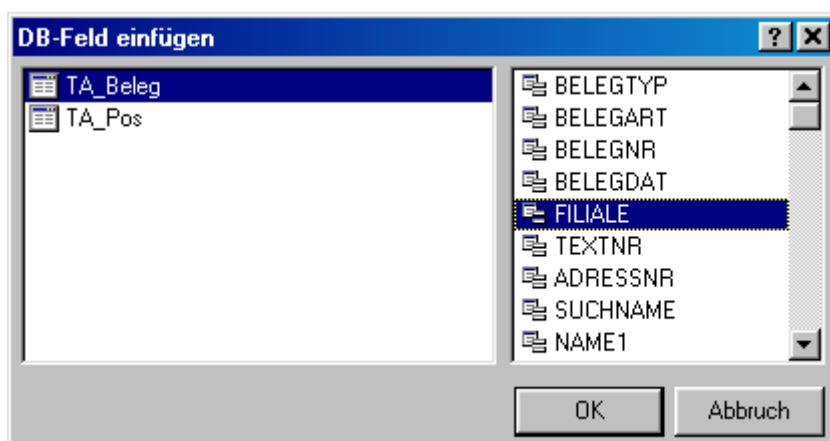
Der Ausdruckeditor

Das Ausdruckseditorfenster kann aus dem Texteditor heraus gestartet werden indem Sie auf  in der Symbolleiste klicken. Sie können ihn auch in einigen Dialogfenstern, wo es nötig ist ein Ausdruck zu erstellen, auswählen (z.B., im Editor der Bandgruppen Bedingungen, Gruppenkopf)



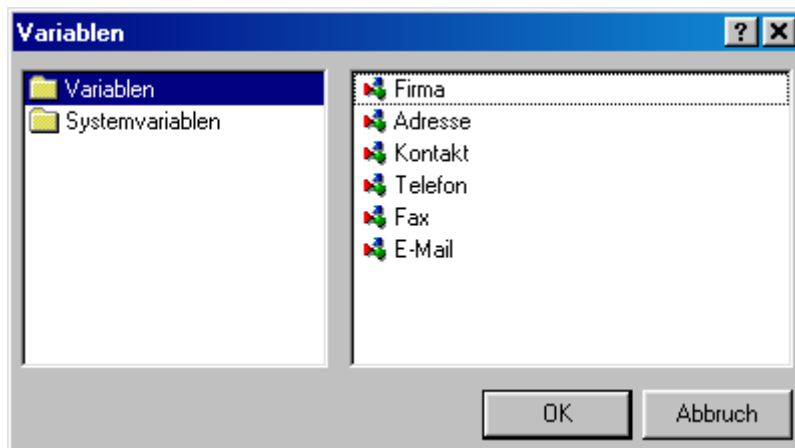
Das Fenster beinhaltet ein Feld in dem man ein Ausdruck eingeben kann und Buttons um Dialogfenster für Variableneinsetzung, DB Felder, Funktionen und Buttons für arithmetische und logische Operationszeicheneinsetzung, aufzurufen

“DB-Feld einfügen” Dialog



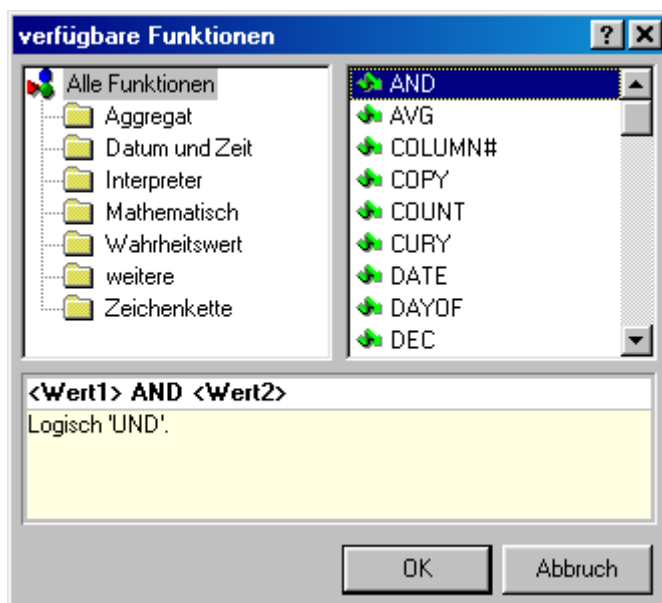
Dieser Dialog hilft Ihnen ein DB Feld zu wählen um es in den Ausdruck einzufügen. Auf der linken Seite gibt es eine Liste aller verfügbaren DB Tabellen. Auf der rechten Seite die Liste der Felder der markierten Tabelle. Um ein Feld in ein Ausdruck einzufügen, wählen Sie ein Feld und drücken „OK“ oder doppelklicken Sie auf das Feld.

„Variable einfügen“ Dialog



Der Dialog hilft Ihnen eine Variable zu wählen um es in den Ausdruck einzufügen. Auf der linken Seite gibt es eine Kategorie Liste. Auf der rechten Seite gibt es eine Variablen Liste in der ausgewählten Kategorie. Um eine Variable in den Ausdruck einzufügen wählen Sie eine Variable und drücken Sie „OK“ oder doppelklicken Sie es.

„Funktion einfügen“ Dialog



Sie können eine Funktion wählen um Sie in den Ausdruck des Dialogs einzufügen. Während die Funktion ausgewählt ist können Sie eine kurze Beschreibung der Funktion und die Liste der Argumente am Ende des Fensters sehen. Wenn die Funktion Argumente hat werden Sie gefragt ob Sie es nach dem „OK“ drücken, füllen wollen.

Funktionsargumente ? X

COUNT(<BereichName>)
Ergibt die Anzahl der Datenzeilen in <BereichName>.

Argument 1 fx

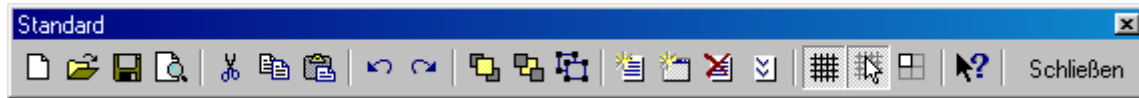
Argument 2 fx

Argument 3 fx

OK Abbruch

Symbolleisten










Die “Standard” Symbolleiste



Symbol	Name	Beschreibung
	Neuer Report	Erstellt einen neuen leeren Report
	Öffne Report	Öffnet einen bestehenden Report aus einer Datei mit der FRF Endung. Tastaturkürzel - Strg+O
	Report Speichern	Speichert den Report in eine Datei mit der FRF Endung. Tastaturkürzel - Strg+S
	Vorschau	Führt den Report aus und zeigt ihn im Vorschau Fenster an. Tastaturkürzel Strg+P
	Ausschneiden	Schneidet ausgewählte Objekte in die interne Zwischenablage aus. Tastaturkürzel Strg+X
	Kopieren	Kopiert markierte Objekte in die interne Zwischenablage. Tastaturkürzel Strg+C
	Einfügen	Fügt Objekte aus der internen Zwischenablage ein. Tastaturkürzel Strg+V
	Rückgängig	Macht letzte Aktion rückgängig. Anzahl der rückgängig Schritte – bis zu 100. Tastaturkürzel Strg+Z
	Wiederholen	Wiederholt die letzte Aktion die rückgängig gemacht wurde. Tastaturkürzel Strg+Y
	nach Vorne	Markiertes Objekt nach vorne setzen.
	nach Hinten	Markiertes Objekt nach hinten setzen.
	Alles auswählen	Markiert alle Objekte auf der Seite. Tastaturkürzel Strg+A
	Neue Seite	Erstellt eine neue leere Seite.
	Lösche Seite	Löscht die aktuelle Seite.
	Seitenoptionen	Zeigt den Seitenoptionen Dialog an.
	Zeige Gitter an	Zeige Raster auf der Seite an.
	am Gitter ausrichten	Wenn Sie bewegen oder in der Größe anpassen, werden die Koordinaten oder Größen im Verhältnis zur Gittergröße geändert.
	An Gitter anpassen	Verändert die Position und Größe der markierten Objekte so dass Sie in das Gitter passen.
	Hilfe	Zeigt die Direkthilfe an dem markierten Objekt an.
Schließen	Schließen	Schließt das Designerfenster.

Die "Text" Symbolleiste



Symbol	Name	Beschreibung
	Zeichensatz Name	Drop-down Liste mit allen Schriftarten die auf ihrem System installiert sind. Doppelklicken Sie auf dieses Feld und es erscheint ein Standard „Wähle Schriftart“ Dialog sehen.
	Zeichensatz Größe	Drop-down Liste aller verfügbaren Schriftartgrößen für die ausgewählte Schriftart. Wenn Sie eine Größe manuell eingeben möchten, klicken Sie auf dieses Feld, geben die entsprechende Größe an und drücken „Enter“
	Fett	Stellt das Schriftart Attribut „fett“ ein. Tastaturkürzel Strg+B
	Kursiv	Stellt das Schriftart Attribut „kursiv“ ein. Tastaturkürzel Strg+I
	Unterstrichen	Stellt das Schriftart Attribut „unterstrichen“ ein. Tastaturkürzel Strg+U
	Textfarbe	Wählen Sie die Textfarbe aus der Drop-Down Farbpalette.
	Hervorhebungseigenschaften	Ändere Hervorhebungseigenschaften.
	Linksbündig	Richtet den Text zur linken Seite des Objekts aus.
	Zentriert	Richtet den Text zentriert zur Breite des Objekts aus.
	Rechtsbündig	Richtet den Text zur rechten Seite des Objekts aus.
	Optimale Breite	Passt den Text an beide Seiten an.
	Oberer Rand	Richtet den Text am oberen Rand des Objekts aus.
	Vertikal zentrieren	Richtet Text zur Mitte der Objekthöhe aus.
	Unterer Rand	Richtet den Text am unteren Ende des Objekts aus.
	Normaler Text / 90 Grad	Ändert die Text Ausrichtung(Normal/90 Grad)

Die "Rechteck" Symbolleiste



Symbol	Name	Beschreibung
	Rahmenlinie oberer Rand	Schaltet die Rahmenlinie oben ein/aus.
	Rahmenlinie linker Rand	Schaltet die Rahmenlinie links ein/aus.
	Rahmenlinie unterer Rand	Schaltet die Rahmenlinie unten ein/aus.
	Rahmenlinie rechter Rand	Schaltet die Rahmenlinie rechts ein/aus.
	Komplett umrahmt	Schaltet alle Rahmenlinien ein.
	Kein Rahmen	Schaltet alle Rahmenlinie aus.
	Hintergrundfarbe	Wählt die Hintergrundfarbe des Objekts aus der drop-down Palette.
	Rahmenfarbe	Wählt die Rahmenfarbe aus der drop-down Palette.
	Linienarten	Wählt den Linien Stil aus der drop-down Liste.
	Rahmenbreite	Wählt die Rahmenbreite aus der drop-down Liste

Die “Ausrichtung” Symbolleiste



Symbol	Beschreibung
	Richtet die ausgewählte Komponente zur linken Ecke der als 1. ausgewählten Komponente aus. (Nicht anwendbar auf einzelne Komponenten)
	Bewegt die ausgewählten Komponenten horizontal bis ihr Mittelpunkt zur 1. ausgewählten Komponente ausgerichtet ist. (Nicht anwendbar auf einzelne Komponenten.)
	Richtet die ausgewählte(n) Komponente(n) am Mittelpunkt des Fensters entlang einer horizontalen Linie aus.
	Richtet 3 oder mehr ausgewählte Komponenten horizontal aus, so dass die Mittelkomponenten abstandsgetreu zwischen den äußeren Komponenten sind.
	Richtet die ausgewählten Komponenten zur rechten Ecke der als 1. ausgewählten Komponente aus. (Nicht anwendbar auf einzelne Komponenten.)
	Richtet die ausgewählten Komponenten zur oberen Ecke der als 1. ausgewählten Komponente aus. (Nicht anwendbar auf einzelne Komponenten.)
	Bewegt die ausgewählten Komponenten vertikal bis ihr Mittelpunkt zur 1. ausgewählten Komponente ausgerichtet ist. (Nicht anwendbar auf einzelne Komponenten.)
	Richtet die ausgewählten Komponenten zum Mittelpunkt des Formulars entlang einer vertikalen Linie aus.
	Richtet 3 oder mehr ausgewählte Komponenten vertikal aus, so dass die Mittelkomponenten abstandsgetreu zwischen den äußeren Komponenten sind.
	Richtet die ausgewählten Komponenten zur unteren Ecke der als 1. ausgewählten Komponente aus. (Nicht anwendbar auf einzelne Komponenten.)

Endanwenderfunktionen

Einführung

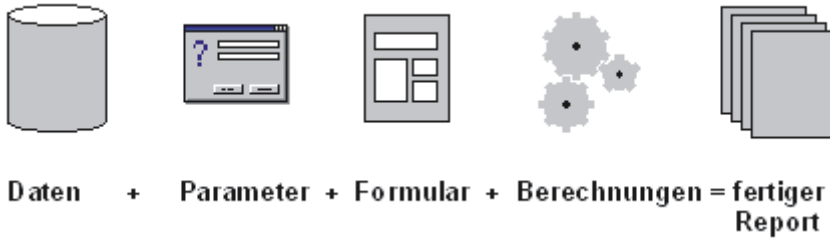
Die Dialog-Formulare

Datenzugriffskomponenten

Eingebaute Sprache

Einführung

Die Reporterstellungsstadien sind wie im Einführungskapitel beschrieben:




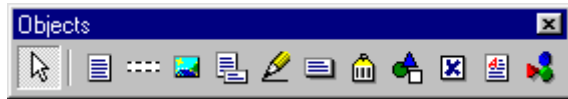
In den Reports, welche im Programm eingebaut sind, können alle Stadien mit Delphi Services realisiert werden. Der Datenzugriff wird über Datenbank-Komponenten bewerkstelligt: Ttable, Tquery... (Als Regel gilt, es gibt eine allgemeine Datenbank für das ganze Projekt und es ist nur möglich auf diese Tabellen zuzugreifen); für die Parameterabfrage können Sie Dialog-Formulare benutzen.

FastReport bietet Ihnen an:

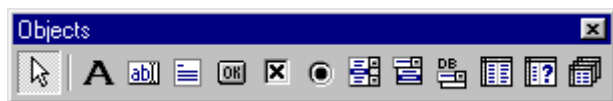
- Neue Reporte zu erstellen und zu verändern;
- Datenquellen zu erstellen die ein Report zur Erstellung benutzen;
- Dialog-Formulare für Reportparameter erstellen die vom Nutzer angefordert werden;
- Reportdaten zu verarbeiten und die Dialogfenster durch Verwendung der eingebauten Pascal-relativen Sprache zu handhaben.

Die Dialog-Formulare

Die Entwicklung von Dialog-Formularen verwendet den selben Designer wie die Reportseiten. Dieser Druckknopf  auf der Designer Symbolleiste erstellt ein neues Dialog-Formular. Die Objektsymbolleiste ändert sich automatisch: entweder in eine Report Seite oder ein Dialog-Formular.

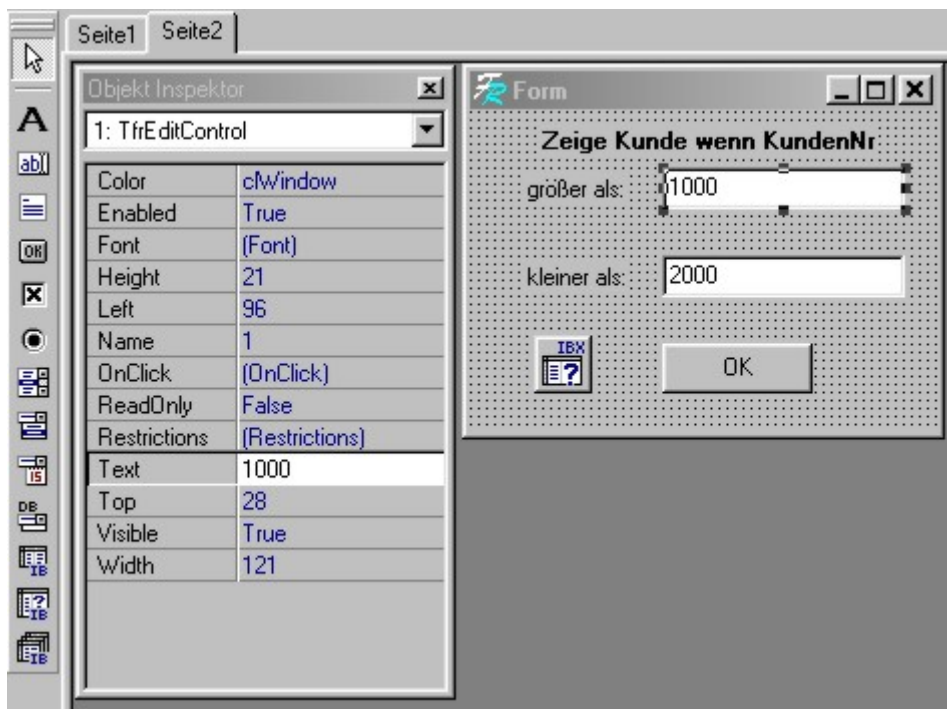


Die 'Objekte' Symbolleiste wenn eine 'Report Seite' aktiv ist.











Die 'Objekte' Symbolleiste wenn ein 'Dialog-Formular' aktiv ist.

Die Bedienungselemente des Designers ändern sich von den Reportseitenbedienungselementen zu den Dialogbedienungselementen, wenn ein Dialog-Formular aktiv wird.



Beachten Sie dass der Prozess ein Report Dialog-Formular zu erstellen sehr ähnlich ist, wie ein Standard Dialog-Formular in der Delphi IDE erstellt wird. Es können mehrere Dialog-Formulare in jedem Report erstellt werden. Diese Dialog-Formulare werden in der Reihenfolge ihrer Erstellung ausgeführt bis alle Dialog-Formulare ausgeführt und durch Drücken von OK akzeptiert werden. Dann wird der Hauptreport vorbereitet und ausgeführt.

Dialog-Formular Bedienungselemente

Icon	Name	Beschreibung
	Beschriftung	Feststehendes Textbedienungselement.
	Eingabefeld	Einzelzeilen Eingabefeld
	Memofeld	Mehrzeilen Eingabefeld
	Druckknopf	Einzelbefehlsschaltflächen Bedienungselement
	Häkchenbox	Boolesche Entscheidung Druckknopf Bedienungselement.
	Radioknopf	Set of mutually exclusive choices button control.
	Liste	Feste Größe, scrollbares Bedienungselement.
	Kombinationsfeld	Drop-down Bedienungselement angegliedert an ein Eingabefeld.

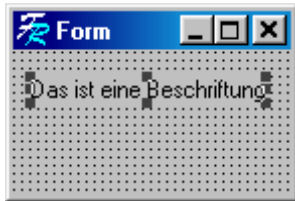
Es ist wichtig zu beachten dass alle Dialog Bedienungselemente in FastReport von der TfrStdControl Klasse abstammen und dass sie die folgenden gemeinsamen Eigenschaften und Methoden besitzen.

Eigenschaft	Standard	Beschreibung
Color	clBtnFace	Hintergrundfarbe des Objekts.
Enabled	True	Aktivieren oder deaktivieren Sie das Objekt.
Font	-	Gibt die Schriftart die mit dem Text des Objekt verbunden ist, an.
Height	-	Gibt die vertikale Größe des Objekts an.
Left	-	Gibt die Koordinate der linken Ecke des Objekts an.
Name	-	Gibt den Namen des Objekts an.
OnClick	-	Tritt auf, wenn der Benutzer auf ein Objekt klickt – ähnlich dem Delphi OnClick Event. Besondere Code Anweisungen können hier mit FastReport Pascal platziert werden.
Restrictions - Beschränkungen	-	Eine Reihe von Flags verbunden mit den Bedienungselementen die den Benutzer beschränken das Objekt zu verändern (bewegen, löschen, bearbeiten, usw.).
Top	-	Gibt die Koordinate der oberen Ecke des Objekts an.
Visible	True	Ermittelt ob das Objekt auf dem Bildschirm erscheint.
Width	-	Gibt die horizontale Größe (breite) des Objekts an.

Als nächstes nehmen wir einen eingehenden Blick auf jedes Bedienungselement.

A Beschriftung

Dieses Bedienungselement wird verwendet um statischen Text anzuzeigen, welcher nicht vom Benutzer geändert werden kann. Sein normaler Zweck ist es erklärenden Text über anderen Bedienungselementen oder großen Mengen von Informationen dem Benutzer anzuzeigen.



Das Beschriftungsbedienungselement hat folgende Eigenschaften (zusammen mit den obigen gemeinsamen Bedienungselementen):

Eigenschaft	Standard	Beschreibung
Alignment - Ausrichtung	taLeftJustify	Steuert die horizontale Platzierung des Texts innerhalb der Beschriftung.
AutoSize	True	Legt fest ob die Beschriftung automatisch die Größe verändert um den Text unterzubringen.
Caption	-	Legt den Textstring fest den die Beschriftung darstellt.
WordWrap - WortUmbruch	False	Legt fest ob der Beschriftungstext in eine neue Zeile umbricht, wenn er zu lang für die Breite der Beschriftung ist. Die AutoSize Eigenschaft muss auf ‚False‘ gesetzt werden, wenn WordWrap ‚True‘ ist.

So können Sie Text zur Laufzeit setzen:

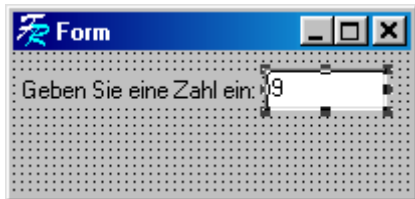
Begin

```
Label1.Caption := 'Das ist eine Beschriftung';
```

End

Eingabefeld

Dieses Bedienungselement wird verwendet um ein einzeligen Textstring, der von der Tastatur eingegeben wird, anzuzeigen und zu verändern.



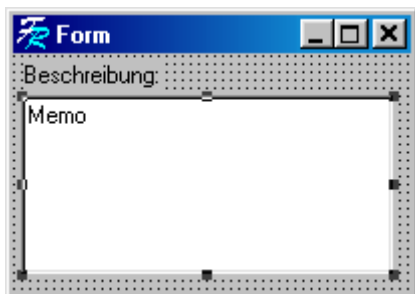
Das Bedienungselement hat folgende Eigenschaften:

Eigenschaft	Standard	Beschreibung
ReadOnly - Schreibgeschützt	False	Legt fest ob der Benutzer den Text des Eingabefeld ändern kann.
Text	-	Beinhaltet den Text als String dargestellt, welcher standardmäßig in dem Bedienungselement angezeigt wird.

```
begin
    Edit1.Text := variable_VonKundenNr;
end
```

Memofeld

Dieses Bedienungselement wird verwendet um einen mehrzeiligen Textstring, der von der Tastatur eingegeben wird, anzuzeigen und zu verändern.



Das Bedienungselement hat folgende Eigenschaften:

Eigenschaft	Standard	Beschreibung
-------------	----------	--------------

Lines - Zeilen	-	Beinhaltet die einzelnen Linien als String dargestellt. Diese Linien werden standardmäßig in dem Bedienungselement angezeigt.
ReadOnly Schreibgeschützt	False	Legt fest ob der Benutzer den Text des Memofelds ändern kann.

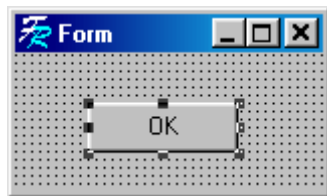
So können Sie Text zur Laufzeit setzen:

```
begin
    Memo1.Text := 'Kunden'+crlf+'Lieferanten';
end
```



Druckknopf

Dieses Bedienungselement wird verwendet um ein Befehlsschaltflächenbedienungselement darzustellen.



Das Bedienungselement hat folgende Eigenschaften:

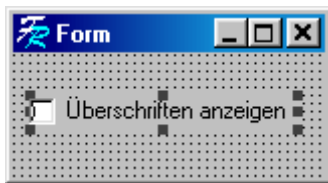
Eigenschaft	Standard	Beschreibung
Caption - Überschrift	-	Beinhaltet den Text der in dem Druckknopf angezeigt wird
ModalResult	mrNone	Legt fest ob und wie der Druckknopf das Ausgangsdialog-Formular schließt und welches Resultat zurückgegeben wird.

Die ModalResult Eigenschaft einzustellen ist ein leichter Weg dass der Druckknopf ein modales Dialog-Formular schließt. Die ModalResult Eigenschaft des Ausgangsformulars bekommt den selben Wert wie die ModalResult Eigenschaft des Druckknopfs, wenn ein Druckknopf geklickt wird. Üblicherweise hat ein „OK“ Druckknopf ein ModalResult = mrOk und ein „Abbrechen“ Druckknopf das ModalResult = mrCancel. So können Sie Überschrift des Druckknopfs ändern:

```
begin
    Button1.Caption := 'OK';
end
```

Häkchenbox

Dieses Bedienungselement wird verwendet um einen booleschen Optionsdruckknopf darzustellen. Der Benutzer kann das Kästchen ankreuzen um die Option auszuwählen oder er kann es nicht ankreuzen um die Option nicht auszuwählen. Beispiele sind Yes/NO, On/Off, True/False, usw.



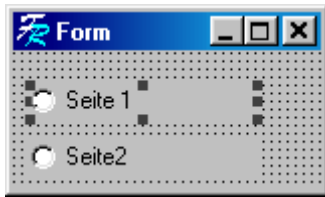
Das Bedienungselement hat folgende Eigenschaften:

Eigenschaft	Standard	Beschreibung
Alignment - Ausrichtung	taRightJustify	Steuert die Position der Kreuzkästchenbeschriftung.
Caption - Überschrift	-	Beinhaltet den Text der in dem Kreuzkästchen angezeigt wird.
Checked	False	Weist darauf hin ob das Kreuzkästchen markiert ist.

```
CheckBox1.Checked := True;  
CheckBox1.Caption := 'Vermerk setzen';
```

Radioknopf

Dieses Bedienungselement wird verwendet um eine Reihe von sich gegenseitig ausschließenden Wahlen als Druckknöpfe darzustellen. D.h., nur ein Druckknopf in einer Reihe von Druckknöpfen kann auf einmal ausgewählt werden. Ein Klick auf ein Druckknopf in der Reihe von Druckknöpfen markiert den Druckknopf und wählt alle anderen Druckknopfs in der Reihe ab.



Das Bedienungselement hat folgende Eigenschaften:

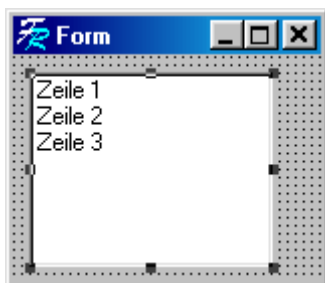
Eigenschaft	Standard	Beschreibung
Alignment - Ausrichtung	taRightJustify	Steuert die Position der Radioknopfbeschriftung.
Caption - Überschrift	-	Beinhaltet den Text der in dem Radioknopf angezeigt wird.
Checked - Markiert	False	Weist darauf hin ob der Radioknopf markiert ist.

```
RadioButton1.Checked := True;
RadioButton1.Caption := 'Kunden';
```



Liste

Dieses Bedienungselement wird verwendet um eine scrollbare Liste von Begriffen die der Benutzer auswählen kann, anzuzeigen.



Das Bedienungselement hat folgende Eigenschaften:

Eigenschaft	Standard	Beschreibung
Items - Elemente	-	Enthält die einzelnen Begriffe aus dem Anzeigetext als String dargestellt. Diese Begriffe werden standardmäßig im Bedienungselement angezeigt.

begin

```
ListBox1.items.Clear;
ListBox1.items.Add('Zeile1');
ListBox1.items.Add('Zeile2');
ListBox1.items.Delete(0);
ListBox1.Items.IndexOf('Zeile2'); // Erster Eintrag fängt mit 0 an
Nr := ListBox1.ItemIndex; // Liefert Nummer der Markierung, beginnend mit 0.
```

```

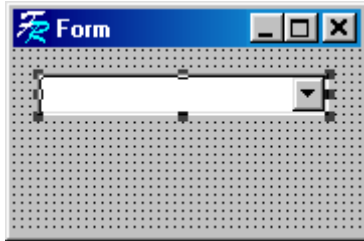
    ListBox1.ItemIndex := 0; // Setzt Markierung. (Erster Eintrag)
    Anzahl := ListBox1.Items.Count; // Anzahl der Einträge in ListBox.
end

```



Kombinationsfeld

Dieses Bedienungselement wird verwendet um eine scrollbare, drop-down Liste von Begriffen die der Benutzer auswählen kann ohne den Platz der für eine Liste benötigt wird, anzuzeigen. Ebenso ist es mit dem zugehörigen Eingabefeld möglich; dem Benutzer zu erlauben Begriffe einzugeben die nicht in der drop-down Liste vorhanden sind.



Das Bedienungselement hat folgende Eigenschaften:

Eigenschaft	Standard	Beschreibung
Items - Elemente	-	Beinhaltet die einzelnen Begriffe aus dem Anzeigetext als String dargestellt in der drop-down Listen Region der Combo Box
Style - Stil	csDropDown	Legt den Anzeige Stil der Combo Box fest.
Text	-	Beinhaltet den markierten Text in der Eingabefeld Region des Bedienungselements.

Dies sind die 3 Kombinationsfeldeigenschaftswerte:

- csDropDown – Erstellt eine drop-down Liste mit einem bearbeitbaren Eingabefeld. Der Benutzer kann selber ein Text Begriff eingeben der nicht innerhalb der Liste enthalten ist. Alle Begriffe sind Strings der selben Höhe;
- csDropDownList - Erstellt eine drop-down Liste mit einem nicht bearbeitbaren Eingabefeld. Der Benutzer kann selber kein Text Begriff eingeben und die einzigen Begriffe die in dem Eingabefeld zur Anzeige verfügbar sind, sind die Begriffe die schon innerhalb der Liste enthalten sind. Alle Begriffe sind Strings der selben Höhe;
- csLookup – Die Strings in der Liste werden in zwei Teilen dargestellt: String1, String2. Die Begriffseigenschaften, wie sie in der drop-down Liste angezeigt werden, werden im ersten Teil des Strings angezeigt: String1. Wenn einer der String1 Werte von der Liste gewählt wird, wird der Wert der String2 enthält dem Eingabefeld gezeigt. Z. B., durch Bestücken der Eigenschaft des Bedienungselements mit; „Januar, 1“, „Februar, 1“, usw.; können Sie die Monate des Jahres darstellen. Der markierte Wert, String 2, der in dem Eingabefeld angezeigt wird ist durch die Texteingenschaft abrufbar.

begin

```

    ComboBox1.Style := csDropDown; // csDropDown; csDropDownList; csLookup

```

```

Monat := ComboBox1.Text; // März
ComboBox1.items.Clear;
ComboBox1.items.Add('Zeile1');
ComboBox1.items.Add('Zeile2');
ComboBox1.items.Delete(0);
ComboBox1.Items.IndexOf('Zeile2'); // Erster Eintrag fängt mit 0 an
Nr := ComboBox1.ItemIndex; // Liefert Nummer der Markierung, beginnend
mit 0.
ComboBox1.ItemIndex := 0; // Setzt Markierung. (Erster Eintrag)
Anzahl := ComboBox1.Items.Count; // Anzahl der Einträge.
end

```

Das Dialog-Formular besitzt auch eine Reihe von Eigenschaften. Diese Eigenschaften können im Objektinspektor angezeigt werden indem Sie mit der Maus auf einen leeren Platz auf dem Formular klicken.

Eigenschaft	Standard	Beschreibung
BorderStyle - RandStil	bsDialog	Die Art des Dialog-Formulars, fixierte Größe oder änderbares Fenster.
Caption - Überschrift	-	Die Textanzeige in der Titel Symbolleiste des Fensters.
Color - Farbe	clBtnFace	Gibt die Hintergrundfarbe des Fensters an.
Height - Höhe	-	Gibt die vertikale Größe des Objekts an.
Left - Links	-	Gibt die Koordinate der linken Ecke des Objekts an.
OnActivate	-	Tritt nach der Initialisierung aller Elemente des Fensters auf.
Position	poScreenCenter	Stellt die Größe und Platzierung des Fenster dar.
Top - Oben	-	Gibt die Y Koordinate der oberen Ecke des Fensters an.
Type - Typ	ptDialog	Gibt die Art des Formulars an, ein Dialog-Formular oder ein Reportseiten Formular
Width - Breite	-	Gibt die horizontale Größe des Fenster an.

Die OnActivate Eigenschaft entspricht ungefähr dem selben Event in anderen Delphi Formularen. Sie wird ausgeführt nachdem alle Bedienungselemente die auf dem Dialog-Formular sind, initialisiert sind und gerade bevor das Formular bereit ist auf dem Bildschirm angezeigt zu werden. Benutzen Sie die OnActivate Eigenschaft zur Initialisierung von Werten in den Formular Bedienungselementen. Z.B., die Eigenschaftsstringliste einer Liste kann zu dieser Zeit mit Werten bestückt werden.

Übergeben der Informationen an den Report

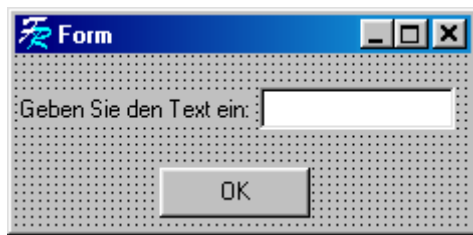
Häufig ist es nötig, Werte in den Report zu übertragen die in Bedienungselementen eingegeben wurden.

Es gibt zwei Wege dies zu tun

1. Einsatz von Variablen
2. Direkte Aufrufe

Einsatz von Variablen

Variablen können benutzt werden um Informationen von einem Bedienungselement zu einem Report zu übertragen. Z.B., wenn es nötig ist Text von einem Eingabefeld in den Kopf des Reports anzuzeigen.



Dies kann im OnClick Event Handler des OK Druckknopfs getan werden.

```
begin
  TitelText := Edit1.Text;
end
```

Im Kopf des Reports ist es erforderlich das "Text" Objekt [TitleText] wie gezeigt zu platzieren:



Direktaufruf

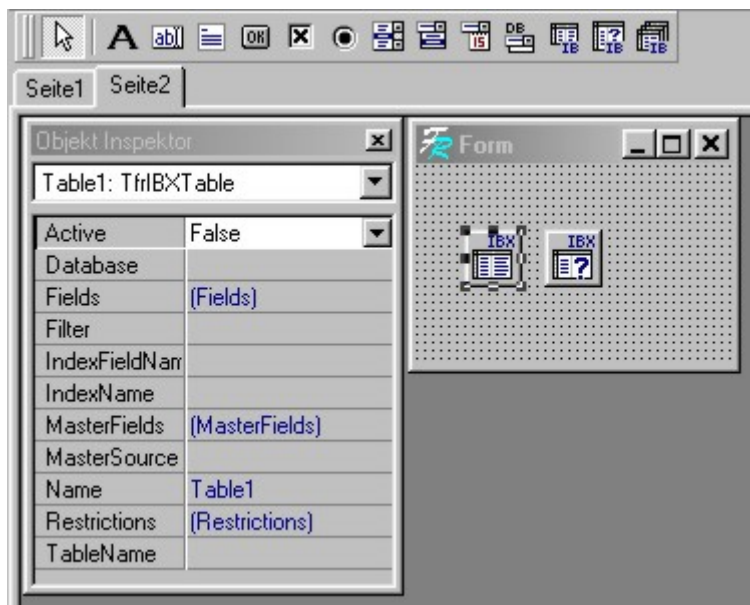
In vielen Fällen ist es einfacher ein Direktaufruf zu verwenden um Werte in den Report zu übergeben. In diesem Fall werden keine dazwischenliegenden Variablen benötigt.




Datenzugriffskomponenten

Die Mehrheit der Reporte greifen in der Regel auf Daten einer Datenbank(DB) zu. Um auf solche Daten zuzugreifen bietet Delphi effektive Mechanismen an, die in FastReport genutzt werden. Diese Datenzugriffskomponenten sind TTable und TQuery, die als Quelle der Daten für den Report genutzt werden können. Im Allgemeinen ist es möglich jede Komponente die auf TDataSet basiert für diesen Zweck zu nutzen.

Abgesehen von einem Datenzugriff, welcher zur Erstellungszeit erzeugt wird, erlaubt FastReport die Erstellung von neuen Komponenten zur Laufzeit. In FastReport ist der Prozess zur Erstellung von Komponenten für den Datenzugriff gleich zu dem, wenn Sie eine Datenbank Anwendung in der Delphi Umgebung erstellen. Eine erforderliche Datenzugriffskomponente wird auf dem Formular platziert und Eigenschaften werden im Objektinspektor angepasst.







Beschreibung der FastReport DB- Komponenten

Wir sollten den Nutzen von Komponenten für den Datenzugriff mit Hilfe von IBX erörtern. Diese Komponenten sind mit der TfrIBXComponents  Komponente der Palette FastReport verbunden und diese Komponenten sind TfrDBLookupComboBox, TfrIBXTable, TfrIBXQuery, TfrIBXDataBase.



Der Zweck dieser Komponenten ist gleich den Datenbank Komponenten TDBLookupComboBox, TTable, TQuery and TdataBase.

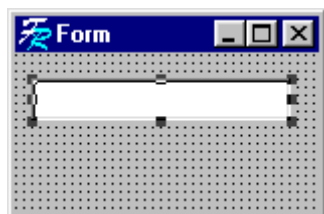
Icon	Name	Beschreibung
	TfrDBLookupComboBox	Drop-down Liste aus Suchbegriffen zum Ausfüllen von Feldern die Daten aus einer anderen Datenquelle benötigen.
	TfrIBXTable	Daten aus einer Tabelle abrufen
	TfrIBXQuery	Daten aus einer SQL Abfrage abrufen
	TfrIBXDataBase	Mit einer Datenbank verbinden

Lassen Sie uns jede Komponente betrachten.



TfrDBLookupComboBox

TfrDBLookupComboBox wird verwendet um dem Benutzer eine praktische drop-down Liste aus Suchbegriffen anzubieten um Felder auszufüllen die Daten aus einer anderen Feld benötigen. Zum Beispiel: Name1 auswählen und KundenNr auslesen.



Das Element hat folgende Eigenschaften:

Eigenschaft	Standard wert	Beschreibung
KeyField - Schlüsselfeld	-	Feldbezeichner des gewählten Werts.
ListField - ListenFeld	-	Felder deren Werte in der Suchsteuerung angezeigt werden.
ListSource - ListenQuelle	-	Datenquelle für die Daten die in der Suchsteuerung angezeigt werden.
Text	-	Gegenwärtig ausgewählter Suchwert

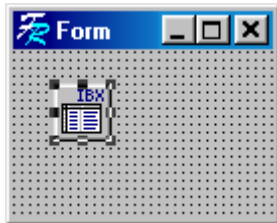
Wählen Sie in ListSource TA_Kunden aus, danach weisen Sie List Field = Name1 und Key Field = KundenNr zu. Der ausgewählte Wert kann über die Text Eigenschaft abgerufen werden.

```
Wert := DBLookupComboBox1.Text; // 1000
```



TfriBXTTable

Diese Komponente wird verwendet um Daten aus einer Tabelle abzurufen.



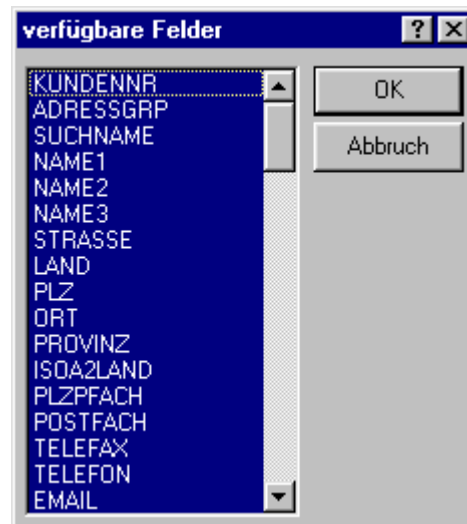
Die Komponente hat folgende Eigenschaften

Eigenschaft	Standard wert	Beschreibung
Active - Aktiv	False	Legt fest ob die Tabelle aktiv ist
DatabaseName	-	Alias für die Datenbank
Fields - Felder	-	Die Liste der abrufbaren Felder.
Filter	-	Nur die Einträge die dieser Bedingung entsprechen werden aktiv, aktiviert oder sichtbar.
IndexName	-	Name des zweiten Index
MasterFields - HauptFelder	-	Die Felder die mit einer Hauptdatenmenge verbunden sind.
MasterSource	-	Hauptdatenquelle
TableName	-	Name der Tabelle.

Um die Komponente mit der Tabelle zu verbinden reicht es die Eigenschaften DatabaseName und TableName auszufüllen. Die Tabelle wird geöffnet indem die Active Eigenschaft der oberen TfriBXTTable auf True gesetzt wird.

Eigenschaft Fields und MasterFields werden mit Hilfe des Editors angepasst. Für diesen Zweck ist es nötig den Druckknopf  in dem Objektinspektor zu drücken.

Der Editor der Eigenschaft Fields erlaubt Ihnen die Felder zu wählen welche während der Tabellensuche abrufbar sind.



Um der Liste Felder hinzuzufügen klicken Sie auf den „Feld hinzu“ Druckknopf. Es erscheint ein Fenster welches alle Felder anzeigt, von dort wählen Sie die Felder und klicken Sie den „OK“ Druckknopf. Um mehrere Felder oder Gruppen von Feldern auszuwählen, klicken Sie mit der Maus auf das erste Feld der Gruppe, dann während Sie „Shift“ halten, wählen Sie das letzte benötigte Feld aus.

Um eines dieser Felder aus einer Liste zu löschen, wählen Sie das Feld dass gelöscht werden soll und drücken Sie den „Löschen“ Druckknopf.

Um ein Lookupfeld hinzuzufügen benutzen Sie den „Lookup hinzu“ Druckknopf.

Folgendes Beispiel demonstriert wie Sie ein Lookupfeld einrichten.

Nehmen wir an, dass es zwei Datenbank Tabellen gibt: der Tabelle „Bestellungen“ mit den Feldern N, Datum, ClientID und Menge und die Tabelle „Kunden“ mit den Feldern ID, Name, Adresse. Die Tabelle „Bestellungen“ beinhaltet die Informationen der Aufträge (Nummer, Bestelldatum, Kennung des Kunden der die Bestellung aufgegeben hat und Summe der Bestellung). Die Tabelle „Kunden“ beinhaltet die Informationen der Kunden (Kennung des Kunden, Name, Adresse).

Um einen einfachen Report aus der Tabelle „Bestellungen“ wie

Bestellnummer - Datum - Name des Kunden - Summe,

zu machen ist es nötig beide Tabellen mit den Feldern ClientID -> ID zu verknüpfen. Dies wird getan indem Sie ein Lookupfeld erzeugen welches zur Tabelle „Bestellungen“ hinzugefügt wird und die Verknüpfung auf ein Feldnamen der Tabelle „Kunden“ darstellt.

Die Dialogbox zur Erstellung eines Lookupfeldes ist aus dem Feldeditor abrufbar.

Um ein Lookupfeld zu erstellen ist es nötig den Feld Namen, den Feldtyp und die Größe festzulegen (falls der Feldtyp ein String ist).

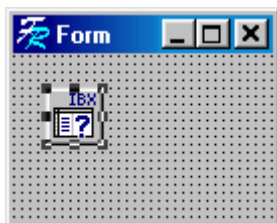
- Primäres Schlüsselfeld - Feld in Quelldatenmenge, welches wie die Verknüpfung auf ein Feld einer Suchdatenmenge handelt.
- Datenquelle - Suchdatenmenge.
- Suchschlüssel - das Schlüsselfeld in der Suchdatenmenge. In unserem Beispiel ist es ein ID Feld.
- Es ist wichtig ein resultierendes Feld auszuwählen, in unserem Beispiel ist es „Name“ Feld.

In der Tabelle „Bestellungen“ gibt es ein Dummy Feld welches das ClientID Feld beinhaltet. Sie können auf das Feld wie auf jedes andere gewöhnliche Feld zugreifen, aber es ist nur lesbar.



TfrIBXQuery

Diese Komponente wird für SQL Abfragen auf Datenbanken verwendet.



Die Komponente besitzt folgende Eigenschaften:

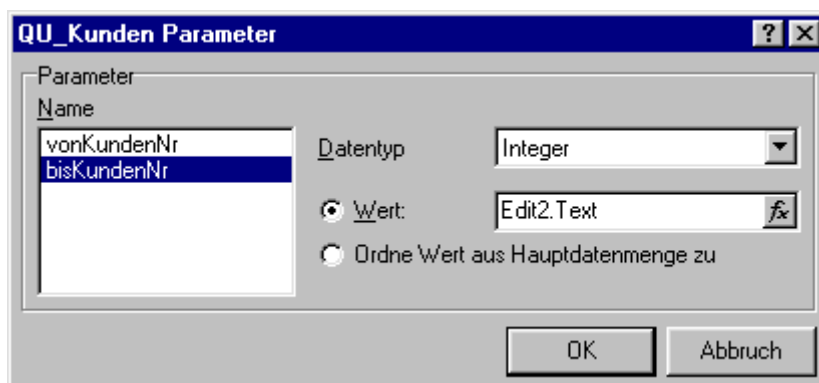
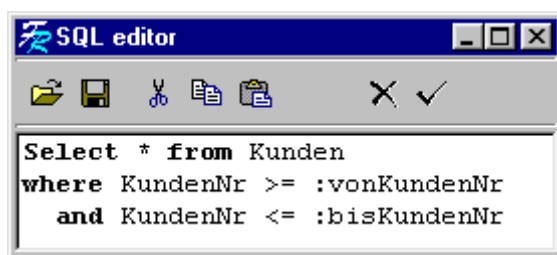
Eigenschaft	Standardwert	Beschreibung
Active - Aktiv	False	Legt fest ob die Suche aktiv ist.

DatabaseName	-	Alias der Datenbank.
DataSource	-	Hauptdatenquelle
Fields - Felder	-	Liste der erreichbaren Felder.
Filter	-	Nur die Aufzeichnungen die dieser Bedingung entsprechen werden aktiv, aktiviert oder sichtbar.
Params	-	Die Liste der Parameter des SQL Statements.
SQL	-	SQL Statement.

Eigenschaften aktiv, Datenbankname, Felder und Filter sind die selben wie in TfrlBXTTable. Die Eigenschaft SQL hat ein eigenen Editor um SQL Text einzugeben.



Die Eigenschaft Params besitzt auch ein Editor der aufrufbar ist wenn der SQL Text Parameter enthält.

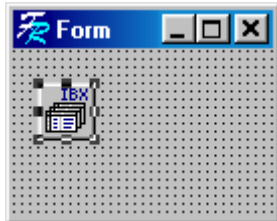


In dieser Dialogbox ist es möglich jedem Parameter ein Typ zuzuordnen und die Quelle des Parameterwerts zu konfigurieren, d.h. entweder von einer Hauptdatenmenge oder durch Zuordnung eines bestimmten Wertes. Falls der Parameterwert aus einer Hauptdatenmenge genommen wird sollte die Datenmenge ein Feld mit dem selben Namen wie der Parameternamen beinhalten. Ansonsten sollte dieses Feld in der Liste der zugänglichen Felder sein (siehe Feldeditor). Deshalb ist es optional den Typ des Parameters anzugeben.



TfrIBXDataBase

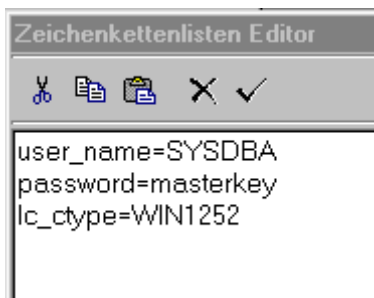
Diese Komponente verbindet mit einer Datenbank. Sie ist ähnlich wie die TDataBase Komponente in Delphi.



Eigenschaft	Standardwert	Beschreibung
AliasName	-	Alias welches im Projekt für die Datenbank verwendet wird.
Connected	False	Stellt die Verbindung her wenn auf True.
DatabaseName	-	Der Name welcher in die Liste der Alias hinzugefügt wird.
LoginPrompt	True	Legt fest ob der Benutzer aufgefordert wird ein Passwort einzugeben während er sich mit der Datenbank verbindet.
Params	-	Verbindungsparameter.

Mit dieser Komponente können Sie sich mit der Datenbank verbinden (üblicherweise wird es verwendet um sich mit dem Serverkontrollsystem der Datenbank zu verbinden).

Ebenso wie in Delphi müssen Sie entweder den Aliasnamen oder den Treibernamen eingeben. Um die Verbindungsparameter einzustellen wählen Sie im Objekt Inspektor die „Params“ Eigenschaft aus und starten Sie den Zeichenkettenlisteneditor.




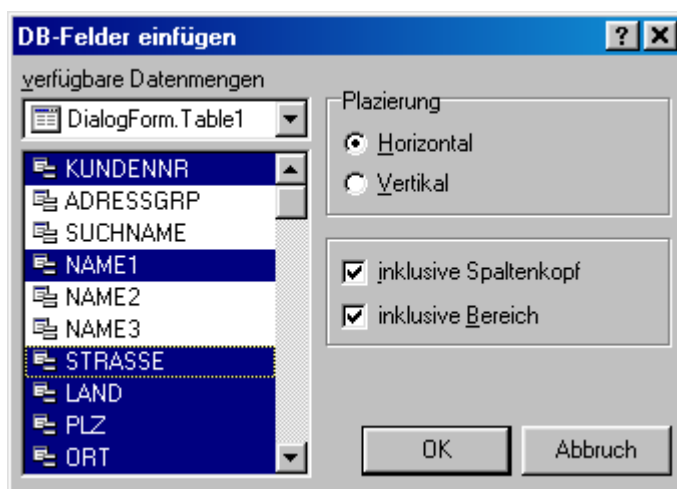
Wenn die „LoginPrompt“ Eigenschaft auf True gesetzt ist erscheint eine Datenbankloginaufforderung wenn Sie sich mit dem SQL Server verbinden wollen. Wenn Sie in den Datenbankparametern Benutzername und Passwort bereitstellen, wie im Dialog gezeigt, können Sie diese Eigenschaft auf False setzen.

Einfacher "Tabellarischer" Report

Dieser Report zeigt Daten einer Tabelle in einer Datenbank. Um den Report zu erstellen folgen Sie diesen Schritten:


Fügen Sie dem Report ein Dialog-Formular hinzu. Dieses Formular wird verwendet um die Datenzugriffskomponenten zu platzieren.

- Platzieren Sie eine TfrIBXTable Komponente auf dem Formular und ändern Sie seine Einstellungen wie beschrieben: Database = ,IB_MAIN', TabellenName = ,KUNDEN'
- Wechseln zum Report Formular und klicken Sie den "DB-Feld einfügen" Druckknopf  in der Wizard Symbolleiste des Designers. Wählen Sie In der Öffnen Dialogbox die benötigten Felder aus und klicken Sie OK.



Jetzt schaut das Report Formular in etwa so aus:

Seitenkopf						
KUNDENNR	NAME1	STRASSE	LAND	PLZ	ORT	
Hauptdaten						
[ENNR]	[AME1]	[ASSE]	[LAND]	[.PLZ]	[.ORT]	

Um sich den Report anzusehen klicken Sie den Vorschau Druckknopf  auf der Symbolleiste.


Report mit Parametern

Lassen Sie uns auf die komplexere Erstellung von Reporten eingehen wo der Benutzer Parameter eingeben muss bevor der Report erstellt wird. Folgen Sie diesen Schritten

- Fügen Sie dem Report ein Dialog-Formular hinzu.

- Platzieren Sie Query, Beschriftung, Eingabefeld und Knopf Komponenten auf dem Dialog-Formular.

- Ändern Sie die Query Komponenten Eigenschaften wie nachstehend beschrieben: Database = ‚IB_MAIN‘, SQL = ‚Select * from Kunden where KundenNr > :Kunde‘
- Öffnen Sie den Params Eigenschaften Editor der Query Komponente und ändern Sie die Parameter Eigenschaften.

- Wechseln zum Report Formular und klicken Sie den “DB-Feld einfügen” Druckknopf  in der Wizard Symbolleiste des Designers. Wählen Sie In der Öffnen Dialogbox die benötigten Felder aus und klicken Sie OK.

Wenn der Report erstellt wird werden Sie durch eine Dialogbox aufgefordert eine Kundennummer einzugeben. Sobald Sie die Nummer eingegeben und OK gedrückt haben wird der Report erstellt. Alle Kunden deren Nummer den Wert den Sie eingegeben haben übersteigt, werden gedruckt.

Feldeditor

In diesem Editor ist es möglich eine Liste von verfügbaren Feldern der Tabelle oder der Abfrage zu füllen. Wenn die Liste leer ist sind alle Felder verfügbar.



Klicken Sie auf den "Feld hinzu" Druckknopf um Felder hinzuzufügen. In dem darauffolgendem Fenster wählen Sie die Felder aus und klicken auf "OK". Um eine Gruppe von Feldern auszuwählen klicken Sie auf das erste Feld der Gruppe dann halten Sie die "Shift" Taste und wählen Sie das Feld welches als letztes ausgewählt werden soll. Um ein Lookupfeld hinzuzufügen verwenden Sie den "Lookup hinzu" Druckknopf. Siehe "Lookupfeld erstellen"

Um markierte Felder aus einer Liste zu löschen verwenden Sie den "Löschen" Druckknopf.

Lookupfeld erstellen

An einem Beispiel wird Ihnen erklärt welchen Nutzen die Lookupfelder haben. Nehmen wir an, dass es zwei Datenbank Tabellen gibt: der Tabelle "Bestellungen" mit den Feldern N, Datum, ClientID und Menge und die Tabelle „Kunden“ mit den Feldern ID, Name, Adresse. Die Tabelle „Bestellungen“ beinhaltet die Informationen der Aufträge (Nummer, Bestelldatum, Kennung des Kunden der die Bestellung aufgegeben hat und Summe der Bestellung). Die Tabelle „Kunden“ beinhaltet die Informationen der Kunden (Kennung des Kunden, Name, Adresse).

Um einen einfachen Report aus der Tabelle "Bestellungen " wie

Bestellnummer - Datum - Name des Kunden - Summe,

zu machen ist es nötig beide Tabellen mit den Feldern ClientID -> ID zu verknüpfen. Dies wird getan indem Sie ein Lookupfeld erzeugen welches zur Tabelle "Bestellungen" hinzugefügt wird und die Verknüpfung auf ein Feldnamen der Tabelle

„Kunden“ darstellt.

Die Dialogbox zur Erstellung eines Lookupfeldes ist aus dem Feldeditor abrufbar.

The image shows a Windows-style dialog box titled "Neues Lookupfeld". It has two main sections. The first section, labeled "Feld", contains three input fields: "Name" with the value "NAME", "Art" with a dropdown menu showing "String", and "Größe" with the value "50". The second section, labeled "Lookup Definition", contains four dropdown menus: "Schlüsselfeld" with "CLIENTID", "Datenmenge" with "TA_Kunden", "Lookup Schlüssel" with "ID", and "Ergebnisfeld" with "NAME". At the bottom of the dialog are two buttons: "OK" and "Abbruch".

Um ein Lookupfeld zu erstellen ist es nötig den Feld Namen, den Feldtyp und die Größe festzulegen (falls der Feldtyp ein String ist).

- Primäres Schlüsselfeld - Feld in Quelldatenmenge, welches wie die Verknüpfung auf ein Feld einer Suchdatenmenge handelt.
- Datenquelle - Suchdatenmenge.
- Suchschlüssel - das Schlüsselfeld in der Suchdatenmenge. In unserem Beispiel ist es ein ID Feld.
- Es ist wichtig ein resultierendes Feld auszuwählen, in unserem Beispiel ist es „Name“ Feld.


In der Tabelle „Bestellungen“ gibt es ein Dummy Feld welches das ClientID Feld beinhaltet. Sie können auf das Feld wie auf jedes andere gewöhnliche Feld zugreifen, aber es ist nur lesbar.

Daten verbinden

In dem „MasterFields“ Eigenschaftseditor können Sie Haupt und Detail Datensätze bildlich verknüpfen. Als Beispiel wähle ich Tabelle Beleg und Belegpos aus. Platzieren Sie auf dem Dialog Form 2 TibxTable Objekte und weisen der Eigenschaft Database „Mainform.IB_Main“ zu. Benennen Sie Tabelle 1 „TA_Beleg“ und weisen Tablename = Beleg und Indexname BE_Beleg zu. Tabelle 2 soll TA_Pos werden (sieh Bild unten)

Objekt Inspektor	
TA_Beleg: TfrlBXTTable	
Active	False
Database	Mainform.IB_Main
Fields	(Fields)
Filter	
IndexFieldNames	
IndexName	BE_BELEG
MasterFields	(MasterFields)
MasterSource	
Name	TA_Beleg
Restrictions	(Restrictions)
TableName	BELEG

Objekt Inspektor	
TA_Pos: TfrlBXTTable	
Active	False
Database	Mainform.IB_Main
Fields	(Fields)
Filter	
IndexFieldNames	
IndexName	BP_POSNR
MasterFields	(MasterFields)
MasterSource	TA_Beleg
Name	TA_Pos
Restrictions	(Restrictions)
TableName	BELEGPOS

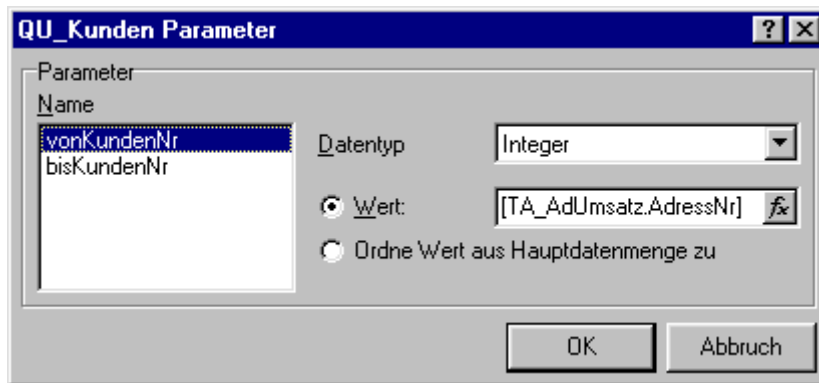
Wenn Sie Tabelle TA_Pos MasterSource TA_Beleg zugewiesen haben kann Belegpos Tabelle mit Tabelle Beleg verknüpft werden. Klicken Sie Druckknopf  in Eigenschaft MasterFields des Objekt Inspektors und Sie erhalten diesen Dialog.

TA_Pos -> TA_Beleg	
Detail Feld	Hauptfeld
<div>POSNR</div> <div></div>	<div>BELEGDAT</div> <div>FILIALE</div> <div>TEXTNR</div> <div>ADRESSGRP</div> <div>ADRESSNR</div> <div>SUCHNAME</div>
<div>Hinzu</div>	
Verbundene Felder	
<div>BELEGTYP -> BELEGTYP</div> <div>BELEGART -> BELEGART</div> <div>BELEGNR -> BELEGNR</div>	
<div>Löschen</div>	
<div>OK</div> <div>Abbruch</div>	

Um Felder von Datensätzen zu verknüpfen, wählen Sie ein Feld aus der linken Liste (Detaildatenmenge) dann wählen Sie ein Feld aus der rechten Liste (Hauptdatenmenge) und dann klicken Sie auf den „Hinzu“ Druckknopf. Dadurch hat sich die linke Liste in die untere Liste bewegt. Um die untere Liste zu löschen betätigen Sie den „Löschen“ Druckknopf. Die verknüpften Felder sollten den gleichen Typ haben und Indexiert sein.

Parameterdialog

Wenn der Report eine oder mehrere Abfragen mit Parametern beinhaltet, ist es notwendig die Parametern Quelle anzugeben.



- **Wert** – hier müssen Sie Datentyp und ein Datenbankfeld oder eine Variable eingeben. Auch komplizierte Ausdrücke sind hier möglich wie:
`copy([TA_Beleg."BELEGTyp"] + [TA_Beleg."BELEGART"] + Str([TA_Beleg."BelegNr"], 1, 7)`
- **Ordne Wert aus Hauptdatenmenge zu** – es ist dabei wichtig das Parameternamen der Abfrage gleich dem Feldnamen der DataSource Datenmenge entspricht. Auf Groß-Kleinschreibung wird nicht geachtet.

Wichtig: Bei mehr als 2 verketteten Datenmengen kann zu Problemen kommen das die übergeordnete Datenmenge noch nicht geöffnet ist. In diesem Fall muss man nachhelfen und die Datenmengen in der Abhängigkeit aktivieren.

Query1.Active := True;

Query2.Active := True;

Query3.Active := True;

Eingebaute Sprache

FastReport besitzt einen eingebauten Pascal-ähnlichen Sprachen Interpreter. Dieser Interpreter ist ein mächtiges Hilfsmittel um sprachunabhängige (Delphi oder C++ Builder) Scripte welche bei der Erstellung der Reports verwendet werden zu schreiben.

Die Sprache die im Interpreter ist ein Pascal Dialekt mit den folgenden Fähigkeiten:

- Operatoren: Zuordnungsoperator ; bedingte Statements, Schleifen Statements und unbedingte Sprünge: **if...then...else**, **while...do**, **repeat...until**, **for..to..do**, **goto**;
- Block **begin...end**;
- Variablen ohne Typ, Arrays;
- Hinweise zu Eigenschaften und Methoden der FastReport Objekte durch Punkt Notation.


Verglichen zu Object Pascal ist diese Sprache sehr vereinfacht. Die folgenden Vereinfachungen werden verwendet:

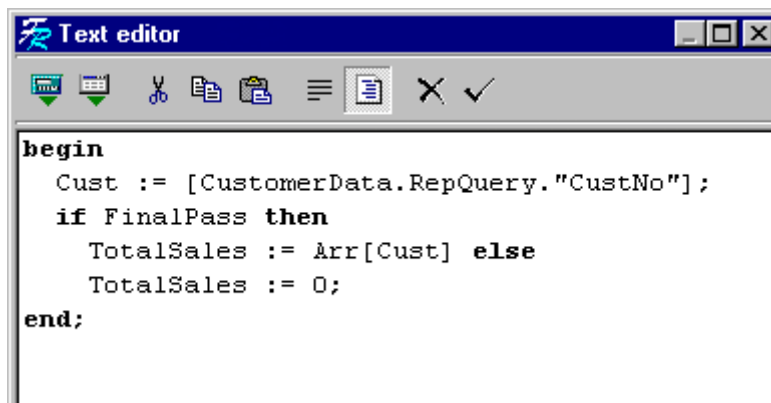
- alle Variablen sind Variant; es gibt keine Typdefinitionen für Variablen;
- alle Variablen sind global, es gibt keine lokalen Variablen;
- es gibt keine Datentypen wie class, record, enumeration type, usw.;
- Sie können nicht Ihre eigene Prozeduren oder Funktionen schreiben;
- Es gibt keine Schleifenunterbrechungszeichen (break, continue);

- Die Anzahl der Argumente die einer Prozedur oder Funktion zugeordnet sind kann nicht 3 überschreiten;
- Aufgrund der Tatsache dass Variablen keinen Datentyp besitzen, fehlt die Datentypkontrolle; dies sollte beachtet werden wenn Sie logische Ausdrücke schreiben;
- Arrays können nur ein-dimensional sein.

Trotz dieser beachtlichen Vereinfachungen ermöglicht Ihnen der Interpreter ziemlich komplexe Datenverarbeitungen durchzuführen. Von dem Script aus haben Sie Zugriff zu allen Methoden und Eigenschaften der Report Objekte, ebenso wie zu den Datenbank Feldern. Im Script können Sie Variablen und Arrays über den ganzen Report zugänglich machen. Die Fähigkeiten der eingebauten Sprache kann mit der Tatsache demonstriert werden, dass solch eine schwere Aufgabe wie Gruppensummen im Gruppenkopf (weil die Summen am Ende der Gruppe berechnet werden) zu drucken für FastReport einfach ist.

Scripte und Objekte

Jedes Objekt kann einen oder mehrere Codeblöcke besitzen. Scriptbearbeitung wird im Texteditor vorgenommen(um das Script zu sehen müssen Sie auf den  Druckknopf oben im Fenster klicken). Das Script wird jedes mal bevor ein Objekt gedruckt wird ausgeführt (Das Script ist der OnBeforePrint Eigenschaft des Objekts angehängt).



```

begin
  Cust := [CustomerData.RepQuery."CustNo"];
  if FinalPass then
    TotalSales := Arr[Cust] else
    TotalSales := 0;
end;

```

Nicht nur Objekte können Scripte verwenden. Bänder und Reportseiten sind auch „scriptbar“. Um das Bereichsskript aufzurufen müssen Sie den OnBeforePrint Editor des Bereichs öffnen (entweder über den Objekt Inspektor oder durch Auswahl des Bereichs und Drücken von Strg+Enter). Um das Script der Report Seite aufzurufen müssen Sie den OnBeforePrint Editor der Seite öffnen (klicken Sie auf der Seite auf einen leeren Platz oder rufen Sie den Editor über den Objekt Inspektor auf). Beide Dialog-Formulare und Report Seiten haben Scripte auf die OnActivate Eigenschaft angehängt. Auf alle anderen Objektskripte können Sie über die Memofeldeigenschaft oder über Strg+Enter zugreifen.

Code schreiben

In Scripten können Sie Eigenschaften und Methoden der Reportobjekte, Datenbanktabellenfelder und verschiedene Variablen benutzen. Sie können auch Variablen und Arrays erstellen die über den ganzen Report zugänglich sind. Sie können auch Prozeduren und Funktionen verwenden.

Variablen verwenden

Sie müssen den "Typ" der Variablen nicht bestimmen, weil Sie alle Variant sind. Als Variablennamen können Sie lateinische Buchstaben, Ziffern und unterstrichene Symbole benutzen. Variablen aus Scripten können in Objekten verwendet werden und Variablen aus der Liste von Variablen können in Scripten verwendet werden.

Ein Beispiel wie man eine dazwischenliegende Variable verwendet:

```
begin
  Cust := [CustomerData.RepQuery."CustNo"];
  if FinalPass then
    TotalSales := Arr[Cust] else
    TotalSales := 0;
end;
```

In diesem Beispiel erstellen wir eine Variable Cust und setzen Sie mit dem Datenbanktabellenfeldwert gleich.

Sie können auch Variablen aufrufen die im Variableneditor, in den Systemvariablen und den Benutzervariablen definiert sind. In diesem Fall könnte der Name der Variablen Symbole beinhalten, die normalerweise nicht aus Syntax Regeln her erlaubt sind(nehmen Sie die Page# Systemvariable als Beispiel). Benutzen Sie eckige Klammern um solche Variablen aufzurufen:

```
begin
  a := [Page#];
end
```

Datenbank Felder verknüpfen

Sie können Verweise zu Datenbanktabellen in Ihren Scripten verwenden. Hier ist die Syntax für solch einen Verweis:

[FormName.TalbeName."FieldName"]

Der volle Pfad wird benutzt wenn die Tabelle und der Report auf verschiedenen Formularen ist (oder Datenmodule). Wenn die Komponenten auf dem selben Formular sind können Sie die Komponenten so *[TableName."FieldName"]* adressieren.

Wenn Sie eine Tabelle adressieren auf der der Bereich den Sie benutzen basiert können Sie einfach *["FieldName"]* schreiben. Z.B., wenn Sie ein Hauptdatenbereich haben welcher über eine Datenquelle an die Customer.db Tabelle angeschlossen ist können Sie sich auf die Felder der Tabelle über den Report beziehen nur durch Verwendung des kurzen Pfades. Verwendung des vollständigen Pfades verlangsamt nicht - FastReport speichert die Datenbankfeldnamen im Cache.

Arrays

Abseits von Variablen können Sie auch Arrays in Ihren Scripten verwenden. Die Arrays dürfen nur eindimensional sein, aber Sie können ihre Elemente so benutzen dass Sie zweidimensional behandelt werden.

Beispiel zur Verwendung eines Arrays:

begin

```
MyArr[0] := 'a'; MyArr[1] := 'b'; MyArr[3] := 'd';
```

```
MyArr[2] := MyArr[0] + MyArr[1] + 'c' + MyArr[3];
```

end;

Tatsächlich werden die Array Elementwerte in der frVariables Liste hinterlegt im *Arr_array_name_index* Format. D.h. in dem oberen Beispiel sind die Inhalte der Variablen:

```
Arr_MyArr_0 := 'a'  
Arr_MyArr_1 := 'b'  
Arr_MyArr_2 := 'abcd'  
Arr_MyArr_3 := 'd'
```

Konstanten

Sie können Konstanten in Ihren Scripten benutzen. Ein einfaches Beispiel ist die Verwendung von numerischen, String und logischen Konstanten:

```
begin
  a := 0;
  b := 'abcd';
  c := True;
  d := 'That''s all!';
end;
```

Achten Sie darauf Einzelanführungszeichen innerhalb von String Konstanten zu verwenden – sie müssen wie in Pascal dupliziert werden: `d := 'That''s all!'`.

Abseits der einfachen Konstanten können Sie solche Konstanten als Farbnamen, Schriftartennamen usw. verwenden. Unterhalb ist die Liste der verfügbaren Konstanten:

- Farben: `clWhite`, `clBlack` etc. – alle Standard Farben + System Farben;
- dialog box response constants: `mrNone`, `mrOk`, `mrCancel`;
- System: `CRLF`, `Null`;
- Schriftschnitte: `fsBold`, `fsItalic`, `fsUnderline`;
- Objektrahmen: `frftNone`, `frftRight`, `frftBottom`, `frftLeft`, `frftTop`;
- Textausrichtung in einem "Text" Objekt: `frtaLeft`, `frtaRight`, `frtaCenter`, `frtaVertical`, `frtaMiddle`, `frtaDown`;
- Bereichsausrichtung: `baNone`, `baLeft`, `baRight`, `baCenter`, `baWidth`, `baBottom`.

Neben diesen gibt es Konstanten für add-in Objekte, zum Beispiel, `csCheck` für "Häkchenbox" Objekte. Alles was Sie in der drop-down Eigenschaftsliste in dem Objekt Inspektor sehen können kann als Konstante in Scripten verwendet werden.

Objekte referenzieren

Sie können die Reportobjekteigenschaften und Methoden in Ihren Scripten referenzieren. Reportobjekte sind visuelle Objekte, Kontrollobjekte, Bereiche, Report Seiten und Reporte selber. Um ein Objekt zu referenzieren wird Punktnotation verwendet, z.B.: *Memo1.Text*. Um spezifische Eigenschaften und Methoden zu referenzieren ist die Punktnotation nicht notwendig.

Eigenschaften die referenziert werden können, werden im Objekt Inspektor angezeigt. Einige kombinierte Eigenschaften wie die der Schriftarten können auf diese Weise referenziert werden *Font.Name*, *Font.Size* etc.:

```
begin
  Memo1.Font.Name := 'Courier New';
  Memo1.Font.Size := 10;
  Memo1.Font.Color := clRed;
  Memo1.Font.Style := fsBold + fsItalic;
end;
```

Eigenschaften wie TStrings (Memo, SQL, Items usw.) können über Ihren Index referenziert werden:

```
if Memo1.Lines[1] = 'a' then  
    Memo1.Lines[1] := 'b'
```

Solche Eigenschaften können auch durch die Verwendung von Add, Delete, Clear und Count referenziert werden:

```
if Memo1.Lines.Count > 10 then  
    Memo1.Lines.Delete(10) // Eintrag Nr. 11 löschen  
else  
begin  
    Memo1.Lines.Clear;  
    Memo1.Lines.Add('a');  
end;
```

Eine vollständige Liste der Objekteigenschaften und Methoden finden Sie im „Objekt Eigenschaften und Methoden“ Abschnitt. **Notieren Sie dass eine nichtexistierende Methode oder Eigenschaft keine Fehlermeldung verursacht, also seien Sie vorsichtig beim Schreiben von Code.**

Prozeduren und Funktionen verwenden

Scripte können Prozedur- oder Funktionsaufrufe beinhalten. Eine besondere Eigenschaft des Interpreters ist, oder um etwas genauer zu sein, der Parser der für die Prozedur/Funktionsverarbeitung verantwortlich ist, dass die Prozeduren und Scripte nicht mehr als 3 Argumente besitzen dürfen. Scripte können beide, eingebaute und externe Prozeduren, verwenden. Die Liste der eingebauten Prozeduren und Funktionen befindet sich im “Eingebaute Prozeduren und Funktionen” Paragraph.

Hinweis. Wenn Sie eine Prozedur oder Funktion referenzieren darf kein Leerschritt zwischen dem Prozedurnamen und der öffnenden Klammer stehen.

Objekt Abänderung

In Ihren Scripten können Sie jede Abänderung der Objekte vornehmen, wie die Größe ändern, Farbe, Inhalt, usw. *Sie müssen daran denken dass in einem Einzeldurchlaufreport keine Objekte mehr geändert werden können die schon verarbeitet wurden.* Dies kommt vor wenn Sie versuchen die Inhalte eines Objekts im Reporttitel von einem Objekt dass auf einem Reportzusammenfassungsbereich liegt, zu ändern, wird sich nichts ändern. Dennoch werden Sie in der Lage sein solch eine Änderung vorzunehmen wenn Sie einen Zweidurchlaufreport machen.

In mehrseitigen Reporten funktioniert es auf die selbe Weise. Sie können jedes Objekt über den Namen referenzieren (Namen innerhalb eines Reports sind einzigartig). Nur noch nicht verarbeitete Objekte können verändert werden. Wenn Sie doch ein Projekt, welches schon verarbeitet wurde, ändern wollen müssen Sie ein Zweidurchlaufreport erstellen.

Eingebaute Funktionen

Angesammelte Funktionen

Angesammelte Funktionen können in der Report Zusammenfassung, in den Seitenfuß, Hauptfuß, Detailfuß, Unterdetailfuß, Gruppenfuß und Kreuzfuß Bereichen verwendet werden.

- **Sum(<expression> [, band] [,1]).** Errechnet die Summe der Werte, die in < Ausdruck > für die gegebene Bandreihe geführt werden. Wenn der Bereichsparameter nicht gesetzt wird, setzt Sum alle Werte auf Standard (auf den Bereichen Hauptdaten, Detaildaten and Unterdetaildaten); andernfalls bezieht sich eine Summe auf die Daten des genannten Bereichs. Wenn der "1" Parameter verwendet wird, wird die Summe auch nichtsichtbare Objekte einschließen. Beispiel:
Sum([Part total], Band1);
Sum([Part total] + [Part price]);
Sum([Part total], Band1, 1).
- **Avg, Min, Max.** Die Syntax entspricht der Summenfunktion. Die Funktion Avg berechnet einen mathematischen Durchschnitt, die Funktion Min gibt das Minimum zurück und die Funktion Max gibt den maximalen Wert einer Reihe zurück.
- **Count(<band>).** Gibt eine Auszählung der Datenzeilen zurück. Beispiel:
Count(Band1).

String Funktionen

- **Str(<value>).** Konvertiert eine Zahl in ein String.
- **Copy(<string>, <from>, <count>).** Gibt einen Substring zurück <string> mit Länge <count> Schriftzeichen, fängt bei <from> an, (dasselbe wie die Delphi Funktion).
- **If(<expression>, <string1>, <string2>).** Gibt ein String zurück <string1>, wenn der Ausdruck True ist; andernfalls gibt er <string2> zurück.
- **FormatFloat(<formatstr>, <value>).** Konvertiert einen numerisch bedeutenden Wert in einen String durch Gebrauch der Maske in formatstr. Die möglichen Werte von <formatstr> sind in der Delphidokumentation "Formatieren von Strings" beschrieben.
- **FormatDateTime(<formatstr>, <value>).** Konvertiert ein Datum/Zeit Wert in ein String durch Gebrauch der Maske in <formatstr>. Die möglichen Werte von <formatstr> sind in der Delphi Dokumentation "Formatieren von Strings" beschrieben.
- **StrToDate(<value>).** Konvertiert den String <value> in ein Datum.
- **StrToTime(<value>).** Konvertiert den String <value> in eine Zeit.
- **UpperCase(<value>).** Konvertiert den String <value> in Großbuchstaben.
- **LowerCase(<value>).** Konvertiert den String <value> in Kleinbuchstaben.
- **NameCase(<value>).** Konvertiert den ersten Buchstaben des String <value> in ein Großbuchstaben und die restlichen Buchstaben in Kleinbuchstaben.
- **Length(<string>).** Gibt die Länge des <string> zurück.
- **Trim(<string>).** Trimmt (löscht) alle Leerstellen am Anfang und am Ende des <string> und gibt das Ergebnis zurück.
- **Pos(<substring>, <string>).** Gibt die Position des <substring> von dem gegebenen

<string> zurück.

Arithmetische Funktionen

- **Int(<value>)**. Gibt den ganzen Teil der Zahl zurück <value>.
- **Frac(<value>)**. Gibt den gebrochenen Teil der Zahl zurück <value>.
- **Round(<value>)**. Gibt eine gerundete Zahl zurück.
- **value1 Mod value2**. Gibt den Rest resultierend aus der Division von <value1> durch <value2> zurück.
- **MinNum(<value1>, <value2>)**. Gibt den kleineren der 2 Werte zurück.
- **MaxNum(<value1>, <value2>)**. Gibt den größeren der 2 Werte zurück.

Andere Funktionen

- **Input(<caption> [, <default>])**. Zeigt ein Dialog Fenster mit dem Kopf <caption> und einem Eingabefeld. Wenn der "default" Parameter gesetzt ist, setzen Sie diesen String in das Eingabefeld. Nachdem der Benutzer auf OK geklickt hat wird der Eingabe String zurückgegeben.
- **Date**. Gibt das aktuelle Systemdatum zurück.
- **Time**. Gibt die aktuelle Systemzeit zurück.
- **Line#**. Gibt die aktuelle Zeilennummer zurück; die Zählung beginnt am Anfang jeder neuen Gruppe. Zum Beispiel:
 - Hauptdaten
 - 1. Detaildaten
 - 2. Detaildaten
 - 3. Detaildaten
 - Hauptdaten
 - 1. Detaildaten
 - 2. Detaildaten
- **LineThrough#**. Gibt die aktuelle Zeilennummer zurück; die Zählung beginnt am Anfang jedes neuen Reports. Zum Beispiel:
 - Hauptdaten
 - 1. Detaildaten
 - 2. Detaildaten
 - 3. Detaildaten
 - Hauptdaten
 - 4. Detaildaten
 - 5. Detaildaten
- **Column#**. Gibt die aktuelle Spaltennummer in einem Tabellenvergleichs Report zurück..
- **Page#**. Gibt die aktuelle Seitennummer zurück.
- **TotalPages**. Gibt die gesamte Anzahl der Seiten in einem fertigen Report zurück. Um diese Funktion zu nutzen muss der Report ein Zweidurchlaufreport sein.
- **DayOf(<date>)**. Gibt den Tag (1..31) eines gegebenen Datums zurück.
- **MonthOf(<date>)**. Gibt den Monat eines gegebenen Datums zurück.
- **YearOf(<date>)**. Gibt das Jahr eines gegebenen Datums zurück.
- **MessageBox(<text>, <caption>, <buttons_and_icons>)**. Zeigt ein Nachrichtendialog Fenster mit Text, Caption und Druckknöpfen an. Gibt ein Wert

zurück welcher der Auswahl des Benutzers entspricht (mrOk, mrCancel, mrYes, mrNo). Verwenden Sie folgende Werte für die <buttons_and_icons> Parameter:

Druckknöpfe	Icon
mb_Ok	mb_IconError
mb_OkCancel	mb_IconQuestion
mb_YesNo	mb_IconInformation
mb_YesNoCancel	mb_IconWarning

Prozeduren und Funktionen die während der Erstellung eines Reports benutzt werden können

- **CurY**. Gibt die aktuelle Y Position zurück wo das nächste Band gedruckt wird. Sie können CurY auch ein Wert zuweisen – dementsprechend wird die aktuelle Position verändert. *Um Pixel in Millimeter und zurück zu konvertieren verwenden Sie folgendes Verhältnis: 18 Pixel = 5mm.*
- **FreeSpace**. Gibt den verbleibenden Platz auf der Seite in Pixel zurück.
- **FinalPass**. Gibt True zurück wenn der Report Zweidurchlauf ist und jetzt den letzten Durchlauf macht.
- **PageHeight**. Gibt die Höhe in Pixel minus die Höhe des Seitenfuß Bereichs zurück.
- **PageWidth**. Gibt Seitenbreite in Pixel zurück.
- **StopReport**. Beendet Report Erstellung.
- **NewPage**. Fängt eine neue Seite an.
- **NewColumn**. Fängt eine Neue Spalte in einem mehrfach-Spalten Report.
- **ShowBand(<band>)**. Zeigt Bereich <band>.

Eigenschaften und Methoden von Objekten

Alle visuellen Objekte des Reports sind Nachkommen der Klasse TfrView. Die folgenden Eigenschaften und Methoden können von einem Script adressiert werden:

Eigenschaft	Typ	Beschreibung
BandAlign - BandAusrichtung	Integer	Gibt die Ausrichtung von Objekten innerhalb eines Bereiches an. Mögliche Werte sind: baNone, baLeft, baRight, baCenter, baWidth, baBottom.
Enabled - Aktiviert	Boolean	Ermittelt ob ein Objekt auf ein Event antworten kann oder nicht. Mögliche Werte: True oder False.
FillColor - FüllFarbe	Integer	Die Hintergrundfarbe eines Objekts. Die Farbe kann über eine Konstante clXXX angegeben werden.
FrameColor - RahmenFarbe	Integer	Farbe eines Objektrahmens.
FrameStyle- RahmenStil	Integer	Gibt den Stil des Rahmens an. Mögliche Werte sind: psSolid, psDash, psDot, psDashDot, psDashDotDot, psDouble.
FrameTyp - RahmenTyp	Integer	Art des Objektrahmens - eine Reihe von Konstanten frftTop, frftBottom, frftLeft, frftRight.
FrameWidth - RahmenBreite	Double	Breite des Rahmens.
Height - Höhe	Integer	Höhe des ausgewählten Objekts.
Left - Links	Integer	Legt die horizontale Koordinate der linken Ecke eines Objekts verhältnismäßig zum Formular in Pixel fest.
Memo - Memofeld	String	Der Text im Memofeld des gewählten Objekts. Die Eigenschaft kann über ein Verweis auf sein Index angesteuert werden, z. B.: Memo[1].
Memo.Count	Integer	Gibt die Anzahl der Zeilen in einem Memofeld zurück.
Name	String	Name des gewählten Objekts.
Stretched - Gestreckt	Boolean	Entweder wird das Objekt gedehnt um die Größe und Form des Bedienungselement anzunehmen oder seine natürliche Größe wird beibehalten.
Top	Integer	Legt die y Koordinate der oberen linken Ecke eines Objekts verhältnismäßig zu seinem Ausgangsteil fest.
Visible - Sichtbar	Boolean	Legt fest ob das Objekt auf dem Bildschirm erscheint oder nicht. Mögliche Werte: True oder False.
Width - Breite	Integer	Legt die Breite (horizontale Größe) eines Objekts fest.

Methoden:

Methoden	Parameter	Beschreibung
Hide	-	Macht ein Objekt unsichtbar indem Sie die „visible“ Eigenschaft auf False stellen.
Memo.Add	String	Fügt eine neue Zeile einer String Liste in ein Memofeld ein.
Memo.Clear	-	Löscht den ganzen Text eines Objekts.
Memo.Delete	Integer	Löscht eine Zeile.
Show	-	Macht ein Objekt sichtbar indem Sie die „visible“ Eigenschaft auf True setzen.

Standardobjekte

Objekt "Rechteckobjekt" (TfrMemoView)

Als Zusatz zu den generellen oben beschriebenen Eigenschaften und Methoden, hat jedes Objekt seine eigenen Eigenschaften:

Eigenschaft	Typ	Beschreibung
Alignment - BandAusrichtung	Integer	Gibt die Ausrichtung des Textes innerhalb eines Objekts an. Mögliche Werte sind: frtaLeft, frtaRight, frtaCenter, frtaVertical, frtaMiddle, frtaDown.
AutoWidth - AutoBreite	Boolean	Legt fest ob das Objekt automatisch an die Breite des Textes in der Größe verändert wird.
CharSpacing - ZeichenAbstand	Integer	Legt den Abstand zwischen den Buchstaben fest.
Font.Name	String	Die Bezeichnung der Schriftart.
Font.Size	Integer	Die Größe einer Schriftart in Pixel..
Font.Style	Integer	Legt fest ob die Schriftart einen dieser Attribute besitzt: fett, kursiv, unterstrichen. Mögliche Werte: fsBold, fsItalic, fsUnderline.
Font.Color	Integer	Legt die Farbe der Schriftart fest.
GapX	Integer	Horizontale Entfernung zwischen dem Objektrahmen und dem Text innerhalb des Objekts.
GapY	Integer	Vertikale Entfernung zwischen dem Objektrahmen und dem Text innerhalb des Objekts.
HideZeros - NullenUnterdr.	Boolean	Wenn dies auf True gesetzt ist werden Nullwerte von Variablen ignoriert. Mögliche Werte: True; False.
LineSpacing - Zeilenabstand	Integer	Abstand zwischen den Zeilen des Textes.
Suppress - Unterdrücken	Boolean	Gibt an ob wiederholte Werte unterdrückt werden oder nicht. Mögliche Werte: True; False.
TextOnly - NurText	Boolean	Gibt an ob Variablen verarbeitet werden oder nicht. Wenn Textonly auf True ist werden Variablen nicht verarbeitet. Mögliche Werte: True; False.
WordBreak - WortTrennung	Boolean	Wenn ein Wort am rechten Rand umbricht, gibt Wordbreak an ob am Ende einer Silbe umgebrochen wird. Mögliche Werte: True; False (Russian words only).
WordWrap - WortUmbruch	Boolean	Legt fest ob der Text am rechten Rand umbricht so dass er in das Objekt passt. Mögliche Werte: True; False.

Objekt "Bereich" (TfrBandView)

Eigenschaft	Typ	Beschreibung
Breaked -Gebrochen	Boolean	Band bricht ab. Mögliche Werte: True; False.
ChildBand - TochterBand	String	Dieses ChildBand wird nach dem Bereich gedruckt.
ColumnGap - Spaltenabstand	Integer	Horizontaler Abstand zwischen den Spalten innerhalb eines Objekts.
Columns - Spalten	Integer	Anzahl der Spalten im Bereich.
ColumnWidth - Spaltenbreite	Integer	Breite einer Spalte.
Condition - Bedingung	String	Gibt die Bedingungen für Gruppierungen an. Betrifft den Bereichsgruppenkopf.
DataSource - DatenQuelle	String	Legt fest woher das Objekt seine darzustellenden Daten erhält.
EOF	Boolean	Legt fest ob das Ende der Datenmenge erreicht ist oder nicht. Mögliche Werte: True; False.
FormNewPage	Boolean	Erzwingt Drucken auf einer neuen Seite nachdem der Bereich und seine Detailbereiche gedruckt wurden. Mögliche Werte: True; False.
Master	String	Gibt den Bereich, welcher zur Gruppierung von Daten verwendet wird, an.
OnFirstPage	Boolean	Druckt auf die erste Seite. Mögliche Werte: True; False.
OnLastPage	Boolean	Druckt auf die letzte Seite. Mögliche Werte: True; False.
PrintChildIfInvisible	Boolean	Druckt entweder Tochterbänder oder nicht wenn das Ausgangsband unsichtbar ist. Mögliche Werte: True; False.
PrintIfSubsetEmpty	Boolean	Druckt entweder das Band oder nicht wenn das Tochterband leer ist. Mögliche Werte: True; False.
RepeatHeader – Kopf wiederholen	Boolean	Bestimmt ob dieser Bereich auf jeder Seite wiederholt wird oder nicht. Mögliche Werte: True; False.

Methoden:

Methode	Parameter	Beschreibung
First	-	Baut die Datenquelle für ein Bereich basierend auf der ersten Aufzeichnung auf.
Next	-	Baut die Datenquelle für ein Bereich basierend auf der nächsten Aufzeichnung auf.
Prior	-	Baut die Datenquelle für ein Bereich basierend auf der vorherigen Aufzeichnung auf.

Object "Bild" (TfrPictureBox)

Eigenschaft	Typ	Beschreibung
BlobType	Integer	Bestimmt den Bildtyp der innerhalb eines Blob Feldes enthalten ist. Mögliche Werte: btBMP, btJPG, btICO, btWMF.
Center - Zentriert	Boolean	Bestimmt ob ein Bild innerhalb eines Objekts zentriert wird. Mögliche Werte: True; False.
DataField - DatenFeld	String	Legt das Feld der Tabelle fest welches ein Bild enthält.
KeepAspect - SeitenVerhältnis	Boolean	Gibt an ob die verhältnismäßigen Proportionen eines Bildes beibehalten werden wenn es in der Größe geändert wird. Mögliche Werte: True; False.

Add-in Objekte

Objekt "Bar code" (TfrBarcodeView)

Eigenschaft	Typ	Beschreibung
DataField	String	Der Name des Feldes der die Daten enthält.

Object "Häkchenbox" (TfrCheckBoxView)

Eigenschaft	Typ	Beschreibung
CheckColor	Integer	Bestimmt die Farbe des Kreuzes wenn die Häkchenbox angekreuzt ist.
CheckStyle	Integer	Bestimmt den Stil der Häkchenbox. Mögliche Werte: csCross, csCheck.
DataField	String	Der Name des Feldes der die Daten enthält.

Objects "RichText", (TfrRichView)

Eigenschaft	Typ	Beschreibung
GapX	Integer	Horizontaler Abstand zwischen dem Objektrahmen und dem Text innerhalb des Objekts.
GapY	Integer	Vertikaler Abstand zwischen dem Objektrahmen und dem Text innerhalb des Objekts.
TextOnly - NurText	Boolean	Gibt an ob Variablen verarbeitet werden oder nicht. Wenn Textonly auf True gesetzt ist werden Variablen nicht verarbeitet. Mögliche Werte: True; False.
DataField	String	Der Name des Blobfeldes der die Daten enthält.

Objekt "Figur" (TfrShapeView)

Eigenschaft	Typ	Beschreibung
Shape	Integer	Legt die Form des Objekts fest. Mögliche Werte: skRectangle, skRoundRectangle, skEllipse, skTriangle, skDiagonal1, skDiagonal2.

Dialog Bedienungselemente

Alle Dialog Bedienungselemente sind Abkömmlinge der Klasse TfrStdControl und besitzen die folgende Reihe von Eigenschaften und Methoden:

Eigenschaft	Typ	Beschreibung
Color	Integer	Die Hintergrundfarbe eines Objekts. Die Farbe kann über eine Konstante clXXX angegeben werden.
Enabled - Aktiviert	Boolean	Legt fest ob ein Objekt auf ein Event antwortet. Mögliche Werte: True oder False.
Font.Name	String	Die Bezeichnung der Schriftart.
Font.Size	Integer	Die Größe der Schriftart in Pixel.
Font.Style	Integer	Legt fest ob die Schriftart eine dieser Attribute besitzt: fett, kursiv, unterstrichen. Mögliche Werte: fsBold, fsItalic, fsUnderline.
Font.Color	Integer	Legt die Farbe der Schriftart fest.
Height - Höhe	Integer	Höhe des Objekts.
Left - Left	Integer	Legt die horizontale Koordinate der linken Ecke eines Objekts verhältnismäßig zum Formular in Pixel fest.
Name	String	Name des Objekts.
Top - Oben	Integer	Legt die y Koordinate der oberen linken Ecke eines Objekts verhältnismäßig zu seinem Ausgangsteil fest.
Visible - Sichtbar	Boolean	Legt fest ob das Objekt auf dem Bildschirm erscheint oder nicht. Mögliche Werte: True oder False.
Width - Breite	Integer	Legt die Breite (horizontale Größe) eines Objekts fest.

Methoden:

Methode	Parameter	Beschreibung
Hide	-	Macht ein Objekt unsichtbar durch Einstellen der visible Eigenschaft auf False.
SetFocus	-	Gives the input focus to the control.
Show	-	Macht ein Objekt sichtbar durch Einstellen der visible Eigenschaft auf True.

Objekt "Beschriftung"

Eigenschaft	Typ	Beschreibung
Alignment - Ausrichtung	Integer	Bestimmt die Ausrichtung einer Zeile Text innerhalb eines Objekts. Mögliche Werte: taLeftJustify, taRightJustify, taCenter.
AutoSize - AutoGrösse	Boolean	Legt fest ob das Objekt sich automatisch zur Breite des Beschriftungstext in der Größe verändert. Mögliche Werte: True; False.
Caption - Überschrift	String	Die Überschrift ist der Buchstabenstring der auf der Beschriftung dargestellt wird.
WordWrap - WortUmbruch	Boolean	Bestimmt ob der Text auf eine andere Zeile umbricht so dass er in die Beschriftung passt. Wenn es auf True gesetzt ist sollte AutoSize auf False sein. Mögliche Werte: True; False.

Objekt "Eingabefeld"

Eigenschaft	Typ	Beschreibung
ReadOnly - Schreibgeschützt	Boolean	Gibt an ob der Benutzer die Inhalte des Eingabefelds ändern kann. Wenn Schreibgeschützt auf True gesetzt ist kann der Wert nicht geändert werden. Mögliche Werte: True; False.
Text	String	Bestimmt den Text der in dem Eingabefeld erscheint.

Objekt "Memofeld"

Eigenschaft	Typ	Beschreibung
Lines - Zeilen	String	Bestimmt die Textzeilen in einer Memo. Auf einzelne Zeilen können Sie über einen Zeilenindexverweis zugreifen z.B. Memo1.Lines [0].
Lines.Count	Integer	Die Anzahl der Zeilen in einem Memofeld.
ReadOnly - Schreibgeschützt	Boolean	Legt fest ob der Benutzer den Inhalt des Memofelds ändern kann oder nicht. Wenn Schreibgeschützt auf True gesetzt ist kann der Text nicht geändert werden. Mögliche Werte: True; False.
Text	String	Gibt den Text der in dem Memofeld erscheint an. Es besteht aus allen Zeilen verknüpft zu einer Zeile (mit den Symbolen CR+LF zur Zeilenumschaltung und Zeilenvorschub).

Methoden:

Methoden	Parameter	Beschreibung
Lines.Add	String	Fügt einer Stringliste in einem Memofeld eine neue Zeile ein.
Lines.Clear	-	Löscht den ganzen Text eines Objekts.
Lines.Delete	Integer	Löscht eine Zeile.

Objekt "Druckknopf"

Eigenschaft	Typ	Beschreibung
Caption - Überschrift	String	Die Überschrift Eigenschaft ist der Text welcher auf dem Druckknopf erscheint
ModalResult	Integer	Wenn der Benutzer wählt, eine Dialogbox durch Drücken eines Druckknopfs zu schließen setzt der Druckknopf das ModalResult um die Box zu schließen. Mögliche Werte: mrNone, mrOk, mrCancel.

Objekt "Häkchenbox"

Eigenschaft	Type	Beschreibung
Alignment - Ausrichtung	Integer	Gibt die Ausrichtung des Textes verhältnismäßig zum Schalter an. Mögliche Werte: taLeftJustify, taRightJustify.
Caption - Überschrift	String	Die Überschrift ist der Buchstabenstring welcher auf der Häkchenbox dargestellt wird.
Checked - Markiert	Boolean	Gibt an ob eine Häkchenbox markiert ist (True) oder nicht (False). Mögliche Werte: True; False.

Objekt "Radioknopf"

Eigenschaft	Typ	Beschreibung
Alignment - Ausrichtung	Integer	Gibt die Ausrichtung des Textes verhältnismäßig zum Schalter an. Mögliche Werte: taLeftJustify, taRightJustify.
Caption - Überschrift	String	Die Caption ist der Buchstabenstring welcher auf der Häkchenbox dargestellt wird.
Checked - Markiert	Boolean	Gibt an ob ein Radioknopf markiert ist (True) oder nicht (False). Mögliche Werte: True; False.

Objekt "Liste"

Eigenschaft	Typ	Beschreibung
Items - Elemente	String	Die Elemente der Liste. Auf einzelne Zeilen können Sie über einen Verweis auf den Index des Elements zugreifen z.B. ListBox1. Items [0].
ItemIndex	Integer	Index der gewählten Zeile.
Items.Count	Integer	Die Anzahl der von der Liste dargestellten Zeilen im Array.

Methoden:

Methode	Parameter	Beschreibung
Items.Add	String	Fügt der Stringliste eine neue in der Liste dargestellte Zeile hinzu.
Items.Clear	-	Löscht alle in der Liste dargestellten Zeilen von der String Liste.
Items.Delete	Integer	Löscht eine in der Liste dargestellte Zeile aus der String Liste. Die Zeile die gelöscht werden soll wird durch den Indexparameter festgelegt.

Objekt "Kombinationsfeld"

Eigenschaft	Typ	Beschreibung
Items - Elemente	String	In Items stehen die Einträge des Kombinationsfeldes. Auf einzelne Zeilen können Sie über einen Verweis auf den Index des Elements zugreifen z.B. ComboBox1. Items [0].
ItemIndex	Integer	Index der gewählten Zeile.
Items.Count	Integer	Die Anzahl der von dem Kombinationsfeld dargestellten Zeilen.
Style	Integer	Die Stileigenschaft legt fest wie ein Kombinationsfeld seine Items darstellt. Dies kann wie eine drop-down Liste mit einem Eingabefeld in der Text eingegeben wird sein (csDropDown), wie eine drop-down Liste mit keinem Eingabefeld so dass das Item nicht bearbeitet werden kann (csDropDownlist) oder als Liste aus einer anderen Quelle (csLookup). Mögliche Werte: csDropDown, csDropDownList, csLookup.
Text	String	Bestimmt den Text der ausgewählt im Kombinationsfeld erscheint.

Methoden:

Methoden	Parameter	Beschreibung
Items.Add	String	Fügt eine neue Zeile hinzu.
Items.Clear	-	Löscht alle im Kombinationsfeld dargestellten Zeilen von der String Liste.
Items.Delete	Integer	Löscht eine Zeile aus der String Liste..

Datenzugriffskomponenten

Objekt "DBLookupComboBox"

Dieses Objekt ist der Nachfolger von TfrStdControl und besitzt die selben Basiseigenschaften und Methoden:

Eigenschaft	Typ	Beschreibung
KeyField - Schlüsselfeld	String	Feldbezeichner des gewählten Werts.
ListField - Listenfeld	String	Felder dessen Werte in der Liste angezeigt werden.
ListSource - Listenquelle	String	Quelle der Daten.
Text	String	Der gewählte Wert.

Objekt "IBXTable"

Eigenschaft	Type	Beschreibung
Active	Boolean	Bestimmt ob die Tabelle aktiv ist.
DatabaseName	String	Name des DB Alias.
Fields - Felder	Variant	Liste der ansprechbaren Felder. Die Eigenschaft kann über Index angesprochen werden – Name eines Feldes: a: = Table1. Fields ['Customer'].
FieldCount	Integer	Anzahl der Felder in einer Datenquelle.
Filter	String	Ausdruck zur Eintragsfilterung.
IndexName	String	Name des zweiten Index.
MasterFields	String	Die Felder die verwendet werden um mit einer Hauptdatenquelle zu joinen.
MasterSource	String	Hauptdatenquelle.
TableName	String	Name der Tabelle.
EOF	Boolean	True wenn das Ende eines Satzes von Aufzeichnungen erreicht ist.
RecordCount	Integer	Anzahl der Aufzeichnungen in der Tabelle

Methoden:

Methoden	Parameter	Beschreibung
Open	-	Öffnet die Tabelle. Es ist das selbe wie Active: = True.
Close	-	Schließt die Tabelle. Es ist das selbe wie Active: = False.
First	-	Baut den Index auf dem ersten Eintrag in der Tabelle auf.
Last	-	Baut den Index auf dem letzten Eintrag in der Tabelle auf.
Next	-	Baut den Index auf dem folgenden Eintrag in der Tabelle auf.
Prior	-	Baut den Index auf dem vorherigen Eintrag in der Tabelle auf.

Objekt "IBXQuery"

Eigenschaft	Typ	Beschreibung
Active	Boolean	Bestimmt ob die Query aktiv ist.
DatabaseName	String	Name des Interbase Datenbank.
DataSource	String	Hauptdatenquelle.
Fields	Variant	Liste der ansprechbaren Felder. Die Eigenschaft kann über Index angesprochen werden - Name eines Feldes: a: = Query1. Fields ['Customer'].
FieldCount	Integer	Anzahl der Felder in einer Datenquelle.
Filter	String	Ausdruck zur Eintragsfilterung.
SQL	String	Der Text der Query. Die Eigenschaft kann über Index angesprochen werden: Query1. SQL [0].
SQL.Count	Integer	Anzahl der Zeilen des Query Textes..
EOF	Boolean	True wenn das Ende eines Satzes von Einträgen erreicht ist..
RecordCount	Integer	Anzahl der Aufzeichnungen in der Tabelle.

Methoden: Die Selben wie in der Komponente IBXTable und diese:

Methoden	Parameter	Beschreibung
SQL.Add	String	Fügt eine Teile hinzu.
SQL.Clear	-	Löscht eine Zeile.
SQL.Delete	Integer	Löscht eine Zeile mit dem gegebenen Index.

Objekt "IBXDataBase"

Eigenschaft	Typ	Beschreibung
Connected	Boolean	Wenn auf True ist die Verbindung hergestellt.
DatabaseName	String	Geben Sie hier den Pfad und Name des Interbase Datenbank an.

LoginPrompt	Boolean	Bestimmt ob es nötig ist den Benutzer aufzufordern das DB Passwort einzugeben .
Params	String	Parameter der Verbindung. Die Eigenschaft kann über Index angesprochen werden: DataBase1. Params [0]. Übliche Parameter für Interbase sind: user_name=SYSDBA password=masterkey lc_ctype=WIN1252
Params.Count	Integer	Anzahl der Parameterzeilen.

Methoden:

Methode	Parameter	Beschreibung
Params.Add	String	Fügt ein Parameter hinzu.
Params.Clear	-	Macht Parameterliste leer.
Params.Delete	Integer	Löscht eine Zeile mit dem gegebenen Index.

Verwendung des Interpreters

Hier sind einige Beispiele zur Verwendung des Interpreters

1. Um die Summe einer Bestellung hervorzuheben: weiß, wenn die Summe weniger wie 2000 beträgt; grün, wenn die Summe zwischen 2000 und 10000 liegt; rot, wenn die Summe größer als 10000 ist, geben Sie das folgende Script in das Objekt mit der Summe ein:

```
if [Summa] < 2000 then
  FillColor := clTransparent
else if [Summa] < 10000 then
  FillColor := clGreen
else
  FillColor := clRed
```

[Summa] ist Ihre Variable oder Feld mit der aktuellen Bestellsomme. Sie können numerische Konstanten zu Farbwahl verwenden:

```
FillColor := 128 + 128*256 + 128*65536 // (gray color)
```

2. Um nur Datenreihen mit Bestellsommen größer als 2000 anzuzeigen, verwenden Sie den folgenden Code im Script des dazugehörigen Datenbereichs (Sie können den Memofeldeditor durch Drücken von Strg+Enter anzeigen):

```
if [Summa] > 2000 then
  Visible := 1 else
  Visible := 0
```

3. Benutzen Sie die FreeSpace Funktion um festzustellen wie viel Platz noch auf der momentanen Seite ist. Wenn nicht genug Platz vorhanden ist rufen Sie die NewPage Prozedur auf um "hard page break" einzufügen und eine neue Seite zu beginnen.

```
if FreeSpace * 5/18 < 30 then NewPage
```

4. Verwenden Sie die CurY Eigenschaft um die aktuelle Position zu verändern. Z. B., benutzen Sie diesen Code in dem Reportzusammenfassungsscript um den Reportzusammenfassungsbereich auf das Ende der Seite zu setzen:

```
CurY := PageHeight - Height
```


Reportbeispiele

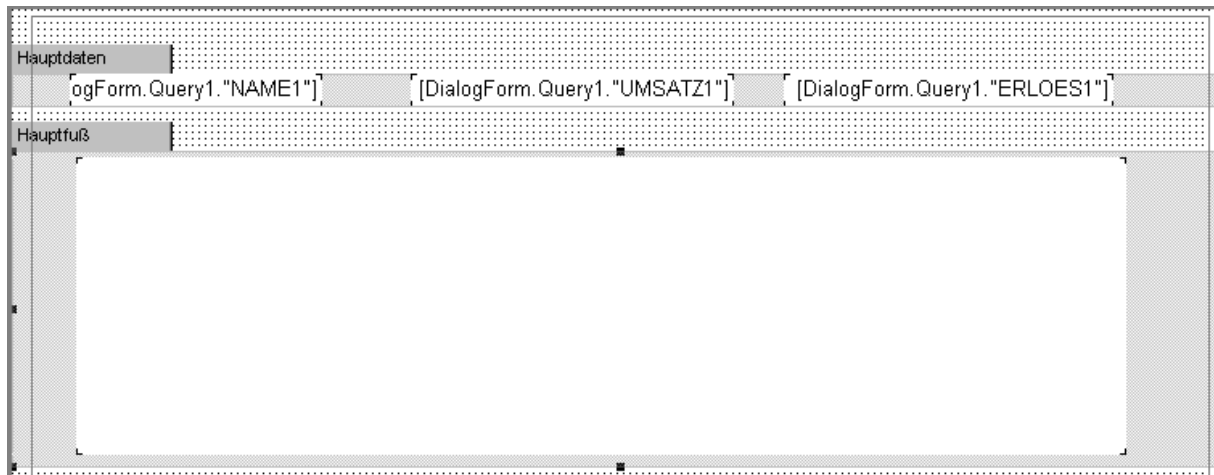
Einfügen von Grafiken und Diagrammen in ein Report

Erstellen Sie ein neues Report und fügen ein Dialog-Form ein. Platzieren Sie auf dem Dialog-Form ein TIBXQuery Objekt mit SQL: `Select k.Name1, u.Adressnr, u.Jahr, u.* From AdUmsatz u left join Kunden k on k.Kundennr = u.AdressNr where Jahr = '2003'`. Wählen Sie als Databasename Mainform.IB_Main aus.

Fügen Sie ein Hauptdaten Band und verbinden in mit Query1.

Erstellen Sie Objekte Memo1, Memo2 und Memo3 mit Felder „Name1“, „Umsatz1“ und „Erloes1“.

Fügen Sie ein Hauptdatenfuß Band und legen ein TfrChartView Objekt  drauf.



Mit einem Doppelklick auf das TfrChartView Objekt erhalten Sie ein Dialog wo die Datenquelle und die Art wie das Diagramm aussieht eingeben können.

Auf erster Karteireiterseite „Art“ können Sie das Aussehen von Diagramm steuern. Geben Sie auf der Seite „Daten“ „für Legende“ ein Objektnamen ein, in unserem Beispiel ist es Memo1 die Daten aus Query1. „Name1“ bezieht (Kundenamen). Im Feld „für Werte“ geben Sie Memo2 ein. Um berechnete Werte in der Diagramm auszudrucken können Sie MemoView verwenden die berechneten Werte anzeigen. Zum Beispiel:

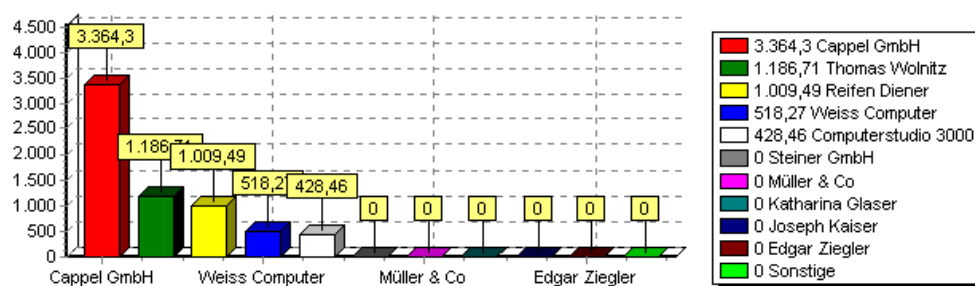
Memo_GesUmsatz.Memo

`[[Query1."UMSATZ1"] + [Query1."UMSATZ2"] + .. + [Query1."UMSATZ13"]]`

In Top 10 Gruppe ungleich 0 (NULL) kann die Diagramm auf N Einträge sich konzentrieren.



Computerstudio 3000	428,46	428,46
Thomas Wolnitz	1.186,71	891,18
Weiss Computer	518,27	276,94
Cappel GmbH	3.364,30	3.364,30
Katharina Glaser	0,00	0,00
Edgar Ziegler	0,00	0,00
Steiner GmbH	0,00	0,00
Edgar Kranz oHG	0,00	0,00
Müller & Co	0,00	0,00
Reifen Diener	1.009,49	363,22
Joseph Kaiser	0,00	0,00



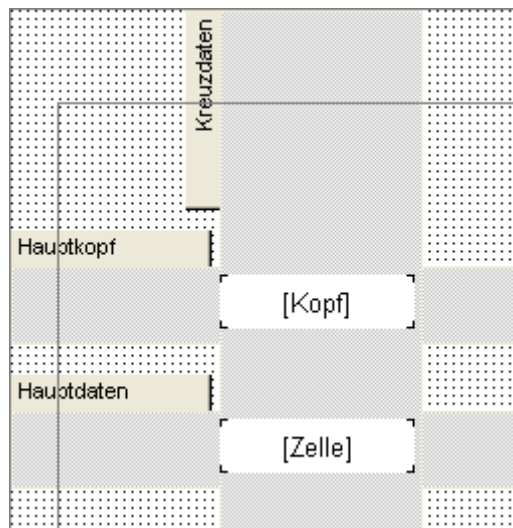
Drucken von Spaltenreports mit variabler oder unbekannter Anzahl von Spalten

In der Praxis ist es oft notwendig Spaltenreports mit einer unbekannten Anzahl an Spalten oder einer Anzahl von Spalten die während der Programmausführung geändert werden auszudrucken. Dies wird z.B. bei der Entwicklung eines Datenbankeditors mit der Möglichkeit Daten aus einer Tabelle auszudrucken benötigt. In diesem Fall könnte der Report mit Programmcode erstellt werden aber FastReport hat eine bessere Lösung für dieses Problem.

Sogenannte Kreuztabellenreports werden für diesen Zweck verwendet. Das unverkennbare Feature eines solchen Reports ist die Daten in Spalten auszudrucken. Es ist nicht notwendig die Menge der Spalten zu wissen. Speziell für diese Art von Report werden KreuzXXX Abschnitte vertikal platziert, nicht horizontal. In diesen Abschnitten werden "Rechteckobjekte" in den Überschneidungen der horizontalen (zum Beispiel «Hauptdaten») und der vertikalen (zum Beispiel «Kreuzdaten») Abschnitte zugeteilt. In diesem Fall wird der Report durch folgendes Prinzip erstellt: für alle Einträge in der Datenquelle des horizontalen Abschnitts werden alle zugehörigen Einträge in der Datenquelle des vertikalen Abschnitts durchsucht und zugeordnet.

Während die Inhalte der Tabelle gedruckt werden, beinhaltet die Datenquelle «Hauptdaten» die Einträge der Tabelle. Die Liste der Felder jedes einzelnen Eintrags ist die Datenquelle für den «Kreuzdaten» Abschnitt.

Das einfachste Beispiel eines Kreuztabellenreports finden Sie im PRNTBL1 Unterverzeichnis..



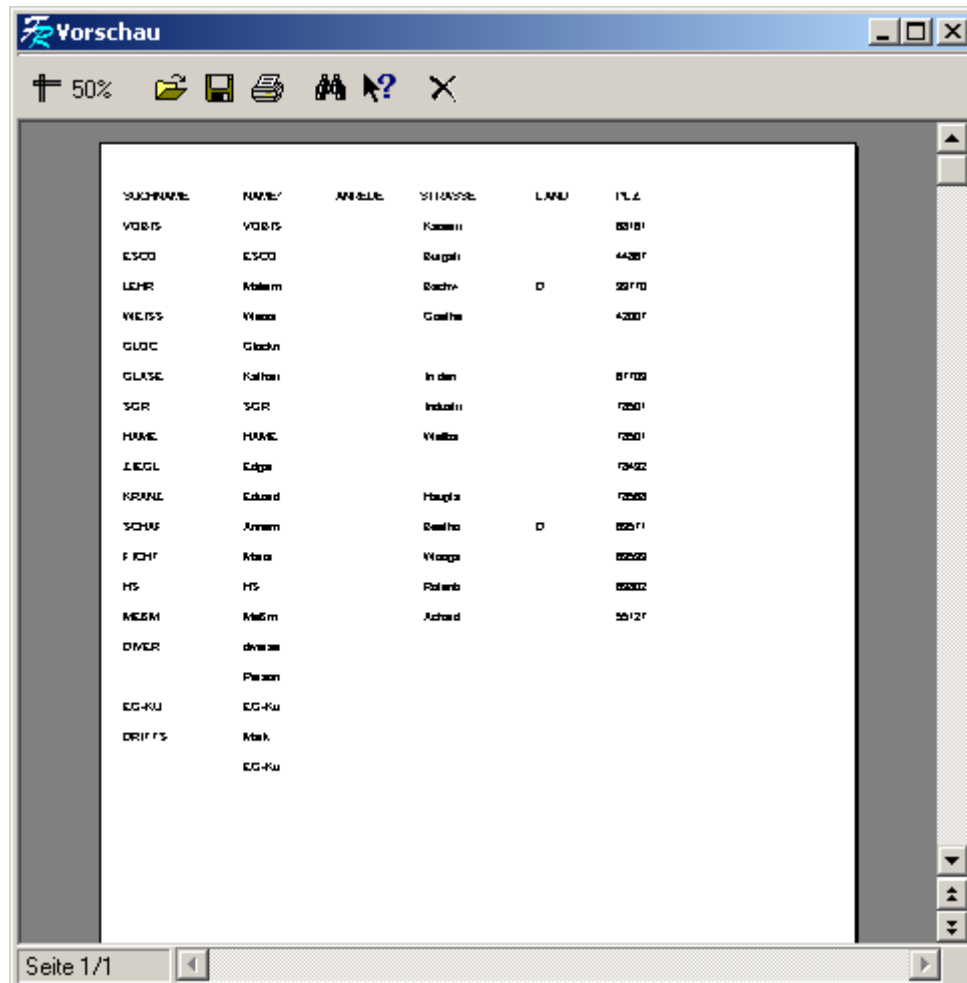
Das Bild zeigt, dass die "Rechteckobjekte" in den Überschneidungen der «Hauptkopf» und «Kreuzdaten» Abschnitte platziert sind, bzw. in den «Hauptdaten» und «Kreuzdaten» Abschnitten. Wenn der Report erstellt wird, werden die Einträge in der CUSTOMER.DB Tabelle als Datenquelle für den «Hauptdaten» Abschnitt verwendet. Die Anzahl der Spalten im Report wird durch zwei TfrUserDataset Virtuellquellen ermittelt welche die Anzahl der "Einträge" mit der Anzahl der Felder der CUSTOMER.DB Tabelle verknüpft. Die Werte der Felder werden im OnGetValue Event des TfrReport Objekts festgelegt:

```

procedure TForm1.frReport1GetValue(const ParName: String; var ParValue:
Variant);
begin
    if ParName = 'Cell' then
        ParValue := Table1.Fields[frUserDataset1.RecNo].Value;
    if ParName = 'Header' then
        ParValue := Table1.Fields[frUserDataset2.RecNo].FieldName;
end;

```

Wenn dieses Beispiel ausgeführt wird können alle CUSTOMER.DB Tabellen als Vorschau betrachtet werden.



VORNAME	NAME	ANSELLE	STRASSE	LAND	PLZ
VORIS	VORIS		Kaiser		83181
ESCO	ESCO		Burgli		44387
LEHR	Meier		Bach	D	32170
WEISS	Weiss		Graf		43007
GLUC	Gluck				
GLASC	Kalhou		In den		87103
SGR	SGR		Industri		72501
HAME	HAME		Waltz		72501
LEGL	Edge				73432
KRAME	Kramer		Haug		72503
SCHAF	Annem		Bach	D	82571
FICHT	Kara		Wagge		82522
HS	HS		Poland		82512
MEER	Meer		Achard		35127
DMER	dmmer				
	Pasen				
EG-KU	EG-Ku				
DRIFT	Mark				
	EG-Ku				

Übersetzung: S. Schneider, GDI mbH, Landau/Pfalz