REPORT GENERATOR
FreeReport v2.32

for LAZARUS

Developer's manual

1999

INTRODUCTION

Necessity of writing own report generator appears in 1997, when developing salary calculation system. Specific of this system is lot of report forms, ability of creating new forms and tuning their face. Reporting tools like QuickReport don't have necessary characteristics, that's why I create own reporting tool. Its conception was taken from "1S-Bookkeeping" 6.0 for Windows: basic report element is framed rectangle with multiline text inside it. Text can contain variables (strings rounded by square brackets). First version of report generator had only one band, but it allows creating multi-level reports.

Later, in 1998, report generator becomes component (first version was only set of units). At this moment, it becomes name "FreeReport" and increases functionality. At today FreeReport is fully visual report generator (many of the reports you can build only with mouse). Its abilities listed here:

- Band-oriented report generator.
- Build-in powerful designer, also available in run-time.
- Preview like in MS Word.
- Fast like QuickReport1.
- Compact code – w/o designer smaller than QR1.
- Unlimited number of pages in prepared report.
- Multi-page reports; composite reports; subreports; groups; multi-column reports; master-detail-detail reports; cross-tab reports; two-pass reports.
- Full control over printing process; support all paper sizes.
- Set of most useful components: Text, Line, Picture, Shape, OLE object, RichText, RX Rich 2.0, Chart, Barcode.
- TXT, RTF, CSV, HTML export.
- Text search in prepared report.
- Add-in component TfrDataStorage intended for creating tables and queries in run-time like QRDesigner does. Especially for end-user reports.
- Editing pages of prepared report.
- Build-in Pascal-like interpreter for handling of building process.
- Report form can store in DFM resources, external file, BLOb field of DB table, or in stream.
- Ability of expanding functionality by own report components, wizards, function libraries.
- IBObjects support in reporter core (with no BDE required).
- Interbase Express (IBX), ActiveX Data Objects (ADO) support in reporter core and in Data manager (with no BDE required).

LICENSE

FreeReport it's under GNU LIBRARY GENERAL PUBLIC LICENSE Version 2, June 1991.

See http://www.gnu.org/copyleft/lgpl.html for complete text or license.txt file include in your package.

1. BEFORE CREATING REPORT

Picture 1. The FreeReport non-visual components in the Delphi Component Palette

These components must be placed on the underlying Delphi form

- *TfrReport* – basic report - generator component. At design-time double Cllick on component to open report designer window. **Required component*.
- *TfrCompositeReport* - Composite report component. Designed for combining several different reports into one. For this type of report you program a fill list refrencing the required TfrReport objects. **Optional only required if building composite reports.*
- *TfrDBDataset* - source infomation for report, connects to data from table or query. This component has property DataSource, which refers to TTable - TDataSource (or TQuery - TDataSource). **Optional required to connect to database data sources - 1 for each databand in report*.
- *TfrUserDataset* - source information, for report connects to non-database data, for example, Array or StringGrid, or text files contents. This component has events OnFirst, OnNext, OnCheckEOF, which allows control of navigation in non-database data. **Optional required to connect to non-database sources of data. 1 required for each data band in report.*
- *TfrOLEObject*, *TfrRichObject*, *TfrCheckBoxObject*, *TfrShapeObject*, *TfrBarcodeObject, TfrChartObject* - link up objects. Settings for each object is discussed below. **Optional required if report uses these items.*
- *TfrDesigner* - report designer. **Optional required if report designer necessary at run-time.*
- *TfrDataStorage* - component for enduser-report. Allow enduser to create tables and queries, they have capabilities similar to those objects of the Delphi environment. **Optional required if report uses this option.*
- *TfrTextExport*, *TfrRTFExport*, *TfrCSVExport*, *TfrHTMExport* - export filter. Exports reports to files of the selected format. **Optional required if report uses these options.*

1.1. VISUAL OBJECTS

«Text» object

This object represents framed rectangle with multiline text inside it. You can set frame type, color and width; all font attributes, text align and rotation (horizontal or vertical).

The contents of the rectangle is a memo object which may contain text, variables or data fields, or a combination of those objects, however font formatting will apply to all objects contained in the rectangle.

Example of rectangle objects memo contents:
Length, см: [Length] - text and [variable]
Length, см: [Table1."Length_cm"] - text and [datafield]
*Length, см: [[Length inch] * 2.54]* - text and [[variable] * value] note addition of second set of square brackets
*Length, см: [Table1."Length_in" * 2.54]* - text and [datafield * value]

The memo of the “Text” object may be accessed in several ways. The fastest is to select the desired rectangle and dblclick, this brings up the memo editor dialog from which you may select what you wish to insert. You may type in text or insert variables or data fields or expressions. Clicking on the datafield or variable buttons in the memo editor dialog will expose the currently available fields or variables, for forming the report. You can use the following quick keys:

Insert = “Variable” button - brings up variable dialog.
Ctrl+Enter = “OK” button - accepts and closes dialog.
Ctrl+F = “Format” button - brings up format editor dialog.
Esc = “Cancel” button - closes and cancels editing.

Format editor allows you to set formatting for variables, used in the memo of «Text» object. You can bring up this editor in memo editor by «Format» button, or in context menu of «Text» object. Variable can be showed as text string (with no formatting), as numeric value, as date, time or boolean value. For each formatting category you can choose one of several formatting strings (for example, you can set number of digits in fraction part of numeric value, choose long or short date format etc.). You also can set custom format string for each category (for example, #,##0.000 for numeric value). Formatting strings in this case is equal to Delphi formatting strings (is described in Delphi online help system, «Formatting strings» topic). Boolean values can be formatted by False_string;True_string string.

This format is implemented to each variable in the memo of «Text» object. If variable cannot be formatted, it showed as text.

If you using several variables in one object, but you want different format for each variable, you can use in-place formatting by «#» tag. Just put this tag and format string inside variable brackets:

[Variable #format], where format is one of following values:

- x.x or Nx.x or Nyyyyy - numeric formatting. x.x - length of number/number of digits in fraction part; yyyyy - string like #,##0.00 (described in Delphi online help system, «Formatting strings» topic). If x.x or yyyyy string contains «.», «,», «-» characters, last this character will be used as decimal separator.
- Dxxxxx, Txxxxx - date and time. xxxxx - string like dd.mm.yy.
- Bxxxxx;yyyyy - boolean formatting. If value is False, xxxxx string will be shown, otherwise yyyyy string will be shown.

There is some examples of using «#» tag:

[Table1.«N1» #9.2] [Table1.«N2» #N9-2] [Table1.«N3» #N#,##0.00] - numeric formatting

[Table1.«Date1» #Ddd.mm.yyyy] [Table1.«Time» #Thh:mm:ss] - date/time formatting

[Table1.«Bool1» #BFalse;True] [Table1.«Bool2» #BNo;Yes] - boolean formatting

You can not use format tag in expressions created in variables editor (see below).

In context menu of «Text» object you can set the following options:

- «Stretched» - object has variable height depending of actual number of text strings in it. You should also turn on this option in object's parent band. When this band printed, it calculates maximal height of objects with Stretched option and stretch itself.
- «Word wrap» - long strings is wrapped onto several lines of text.
- «Autosize» - before drawing, object calculate its actual height and width and stretch itself.

«Band» object

FreeReport, is a banded report generator, bands control where dynamic data of the report appear on it's output pages, the table below Lists the types of bands and where they appear in the finished report. The bands do not have to appear in the correct order in the Report designer window's Page the band type governs where they are located on the finished report. However placing them in the correct order makes it much easier to modify reports later.

Name	Where and when
Report title	Prints once at beginning report
Report summary	Prints once at end report
Page header	Prints once at the top of each page
Page footer	Prints once at bottom of each page
Master header	Prints once at beginning of first data level
Master data	Data of first data level – repeats once for each master data record
Master footer	Prints once at end first data level
Detail header	Prints once at beginning of 2nd data level
Detail data	Data of 2nd data level – repeats once for each detail record
Detail footer	Prints once at end of 2nd data level
Subdetail header	Prints once at beginning of 3rd data level
Subdetail data	Data of 3rd data level – repeats once for each subdetail record
Subdetail footer	Prints once at end of 3rd data level
Overlay	Prints once on each page lower layer
Column header	Prints once at beginning each column
Column footer	Prints once at end each column
Group header	Group title prints once at beginning of group
Group footer	Prints once after group
Cross header Cross data Cross footer	This group of bands is designed for creating crosstab reports which have variable amount of columns on the page.

There is several band editors (for data-band, group header, and cross-data bands).

Data-band editor allows you to choose available dataset for the band or assign «virtual» dataset to it. When assigning virtual dataset, you should set number of records in it. When building report, band with virtual dataset will be printed as many times as many records in this dataset.

Group header editor allows you to set grouping condition based on the fields of DB table, or on any expression.

In context menu of «Band» object you can set the following options (depending of band type):

- *Stretch* - height of band determined by maximum height of objects in the band, “Text” Objects stretch and word wrap options must also be set to on. Primary use would be in limiting horizontal size of a datafield and extending the band (row) height as required.
- *Breaked* – Will try to fill any unused space on Page before creating a new Page.
- *Force new Page* – will force the band’s contents to be printed on a new Page each time it is required.
- *On first page* – band prints on first page (page header & page footer only).
- *On last page* – band prints on last page (page footer only).
- *Repeat on all pages* – this option is only available for band types Master header, Detail header, Subdetail header, Group header, Cross header. If the amount of data under on of these headers requires a new Page these headers will be included on that Page.

 **«Picture» object**

Designed for insertion of graphics in the report file. Formats supported include BMP/WMF/ICO (and JPG, dependant on Delphi version). Object’s editor allows you to choose file with picture and clear picture contents. You can also use pictures from BLOb field of DB table. To do this, put reference to this field into memo of object, for example: [Table1."GraphicField"]. Memo editor may be invoked by pressing Ctrl+Enter key on «Picture» object or by Object inspector.

The context menu of the picture object allows you to set the following options:

- *Stretch* – drawing will stretch to fill the rectangle.
- *Maintain Aspect ratio* – if stretch is selected drawing proportions are preserved.
- *Center* – centers the drawing in the rectangle.

 **«SubReport» object**

Subreport object is a placeholder for another reports, which are situated on a separate page attached to main report. These pages display in the subreport object instead of being shown as a separate page of the report. When you place a subreport object the reference to the correct page is automatically created for you and a new page tab is added to the page window. Typical use of the subreport is where sub details are placed under sub details or in a side by side situation.

Subreport objects can be placed side-by-side. If you want place subreport objects one-under-other, spread it on separate data-bands. There is some limitation of using subreports:

- Don’t use columns in subreports.
- Don’t use ReportTitle, ReportSummary, PageHeader, PageFooter, ColumnXXX bands.
- Don’t use breaked bands.
- Don’t use groups.

 **«Line» object**

The “Line” object is used to draw horizontal or vertical lines. The thickness and color are changeable. To draw a Line click on the drawline object. When you move the mouse over the currently active page the cursor will change to a pencil. Click and hold down mouse button where you wish to start the line, then move mouse where you want it to end, and release mouse button. When finished drawing lines click the select tool. To change the color or thickness of the Line use the tools in the frame formatting toolbar.

«CheckBox» object

Appears as a rectangle with an “X”. In object’s memo editor place a reference to a variable or a field of a table, which are Boolean values. When the value is true the “X” will display if value is false rectangle will be empty.

For using this object in run-time, you should use TfrCheckBoxObject component from FR components palette. If opened report contains non-plugged components, you'll get an error message.

«RichText» object

Designed for insertion of RTF (Rich Text format) documents created externally in other applications such as MS-Word into the report. It will retain all the formatting of the RTF file. The RTF editor is based on the delphi\demos\richedit component, it allows all basic operations on text - change font, smoothing, create labelled list. It allows you to enter variables, enclosed in square brackets, as in “Text” object.

You can also use RTF’s from BLOb field of DB table. To do this, put reference to this field into memo of object, for example: [Table1."RichField"]. Memo editor may be invoked by pressing Ctrl+Enter key on «RichText» object or by Object inspector.

If text out of object’s bounds, extras strings will not printed.

For using this object in run-time, you should use TfrRichObject component from FR components palette. If opened report contains non-plugged components, you'll get an error message.

«OLE» object

Designed for insertion of an OLE object in document. The “OLE” object’s editor allows insertion of a new OLE object. Click on insert in the editor to invoke the standard OLE insert dialog window where you will be presented with a list of available ole objects.

You can also use OLE objects from BLOb field of DB table. To do this, put reference to this field into memo of object, for example: [Table1."OLEField"]. Memo editor can be showed by pressing Ctrl+Enter key on «RichText» object or by Object inspector.

«Stretched» option in context menu of object allows you to show data from Excel grid correctly in some cases.

For using this object in run-time, you should use TfrOLEObject component from FR components palette. If opened report contains non-plugged components, you'll get an error message.

«Chart» object

Designed for insertion of graphic charts or diagrams in a document. The Type tab of the chart editor allows you to pick from six types of charts and their display options.

For link chart to the data fields, you should set names of two "Text" objects, which contents will be used as chart's value and legend string for each value. When building report, contents of these objects will be accumulated in the memo of chart object. The Data tab allows you to supply the names of these objects. I.e. for legend might be “memo1”, for values might be “memo6”.

«Chart» object allows you to create «Top-10» charts. This chart represents several biggest values and summary of other values not included in chart. To do this, set number of top values on «Data» tab of chart editor and set legend name for non-included values («Others» word is usual used).

The Marks tab of the chart editor allows you to pick what type of marker will be shown on the chart value is the default. If any other selection is made the show marks option of the display tab must be set.

If «Text» object which used as chart’s value contains formatted value (for example, «10 000.00» or «$100.00»), FR attempts to expand numeric value from this string. It skips all non-digit

symbols at begin and end of string, then skips all symbols - digit separators (usually «spaces»). If you set more advanced formatting (for example, «10000km2»), you can't using these objects as chart values. You should create non-visible object with same contents, but with no formatting, and use this object as chart value. You can hide object by Objects inspector (set Visible property of object to 0).

You can use two kinds of charts: single and multiple. Multiple charts show several charts on one coordinate grid. When used single charts, memo of chart object contains the following strings:

Header1;Header2;Header3
Value1;Value2;Value3

When used multiple charts, memo of chart object contains the following strings:

Header1;Header2;Header3
Value1.1;Value2.1 *(first chart)*
Value2.1;Value2.2;Value2.3 *(second chart)*
Value3.1;Value3.2;Value3.3;Value3.4;Value3.5 *(third chart)*

In this case, you'll get three charts: first with two points, second with three points and third with five points.

You can also build chart by editing its memo. Just put appropriate values into memo of object.

This component is available in Delphi 3 and above. If you want use it in Delphi 2, you should install TeeChart component first, then correct FR.INC file and recompile FR library.

For using this object in run-time, you should use TfrChartObject component from FR components palette. If opened report contains non-plugged components, you'll get an error message.

«Shape» object

Designed for insertion of geometrical shapes in the report. i.e. (rectangle, rounded rectangle, ellipse, triangle). When you place a shape object the default dialog editor will prompt you for the desired shape, you may set the thickness of the shape border and the fill color by using the color and size tools in the frame formatting toolbar. Note the triangle shape does not allow full background color fill instead it produces a band of color across the middle of the triangle.

For using this object in run-time, you should use TfrShapeObject component from FR component palette. If opened report contains non-plugged components, you'll get an error message.

«Barcode» object

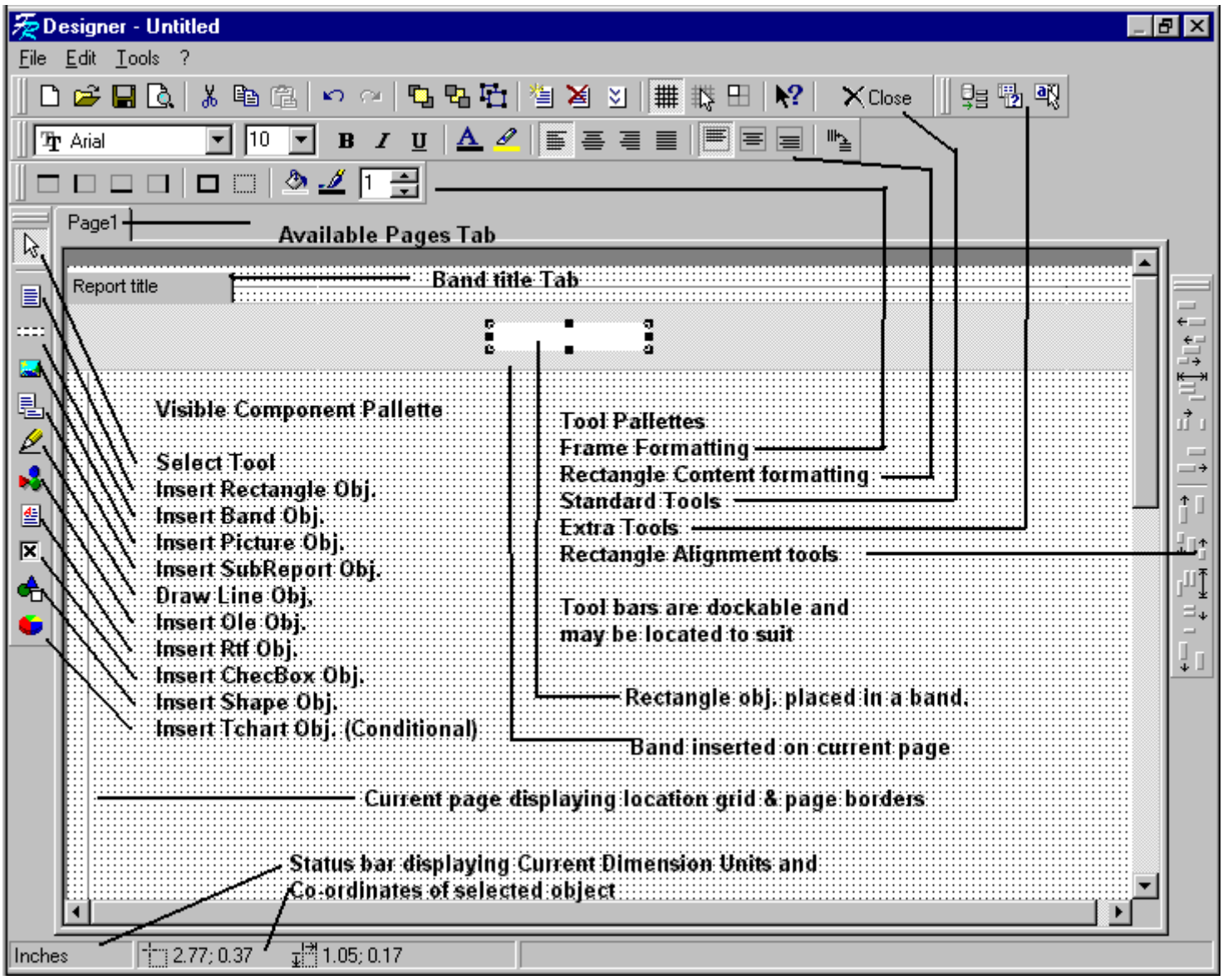
Designed for insertion of bar codes in the report. You can use 2 of 5 interleaved, Code39, Code39 Extended, Code128A,B, Code93, Code93 Extended, MSI, PostNet, Codebar, EAN8, EAN13 bar codes.

For using this object in run-time, you should use TfrBarcodeObject component from FR component palette. If opened report contains non-plugged components, you'll get an error message.

2. DESIGNER BASICS

FreeReport is supplied with a built in report designer**,** which is accessed at design time by double clicking the mouse on the TfrReport component on the project form. The designer allows quick and easy access to report designing and previewing reports within the Delphi IDE. The designer was created with dockable panels (toolbars), which may be changed to suite the user taste. Information about the panels are kept in memory while using the designer, when the designer is closed the tool bar properties are written to the windows registry so the will be used when you re-enter the design window..

To use the designer for enduser runtime requirements it is necessary to place a TfrDesigner component on the underlying form in the project, or place a reference to FR_Desgn in the **uses** list of the project form's module. This gives the end user the ability to design and edit reports.



2.1. USING THE KEYBOARD

- Arrow keys – move to next object.
- Ctrl + Arrowkeys - moves selected object(s) in direction of arrow.
- Shift + Arrowkeys - increase or decrease dimension of selected object(s) in arrow direction.
- Enter - brings up editor of selected object.
- Del - delete selected object.
- Ctrl + Enter - brings up memo editor of selected object.
- Ctrl + 1..9 - sets frame thickness of selected object.
- Ctrl + Z - undo last action.
- Ctrl + Y - redo cancelled action.
- Ctrl + G - toggles grid on/off.
- Ctrl + B - toggles align to grid option on/off.
- Ctrl + F - turn object's framing off.
- Ctrl + D - turn object's framing on.
- Ctrl + X - cut to clipboard.
- Ctrl + V - paste from clipboard.
- Ctrl + C - copy to clipboard.
- Ctrl + A - select all objects on Page.
- Ctrl + N - create new empty report.
- Ctrl + O - open report file.
- Ctrl + S - save report file.
- Ctrl + P - preview report.

2.2. USING THE MOUSE

- Left click - in Page window selects object; in visual component palette selects object to insert – after select LeftClick over Page window inserts new object.
- Right click – brings up context menu for selected object.
- Double click – brings up default editor of selected object. Double click over free space on Page brings up the Page option dialog, where you may set the Page options i.e. margins, size etc.
- Shift + left click – toggle object's selection.
- Ctrl + left click – draws selection rectangle. All object that fits in this rectangle will be selected after you release mouse key.
- To scale several selected objects, drag red square on the right-bottom corner of object's group.

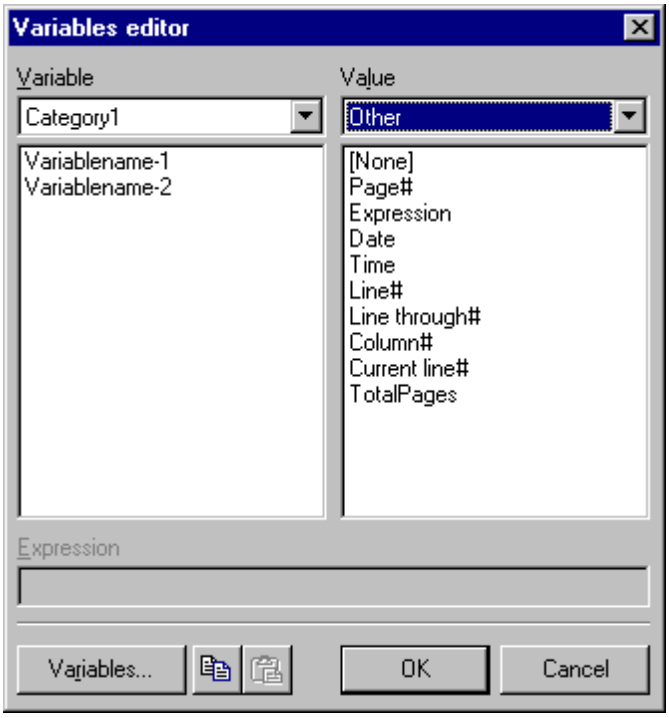
2.3. REPORT VARIABLES

There are many reasons for the use of named Variables in reports, an example would be to create names for cryptic field names like n1 n2 n3 in an underlying data base. When looking at such names in a report design they don't make much sense it would be much better to have something like LastName, MiddleName, FirstName to insert as an object than the cryptic names.

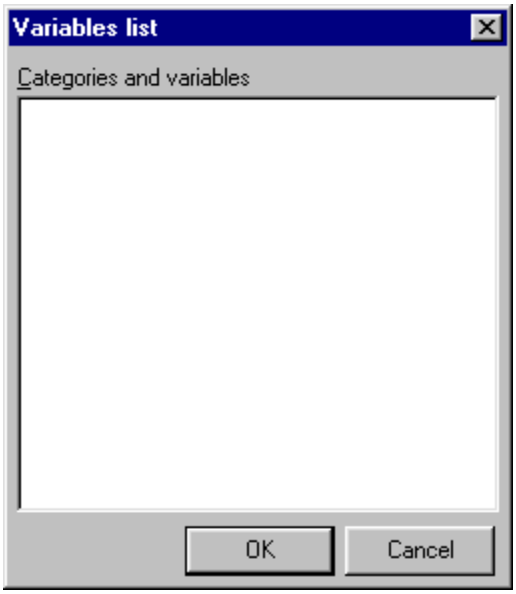
These variables can be more than just replacements for field names; they could be mathematical expressions like sums or rounded values, dates, time, string concatenation etc.

To work with a list of variables it is first necessary to create it. To show the available variable list, choose "File|Variable list" in report designer's menu.

If no variables created, left drop-down list of Variable categories will be empty and the variable list below it will be empty. The drop-down list on the right displays the available datasets and the window below shows available data fields of the selected dataset.



To create a variable list click on the “Variables” button at the bottom left of the variable List editor and the Variables list window appears as below.



In Variables list editor you should type categories and its variables in the following format:

CATEGORY1
  Variable1
  Variable2
^ space is required!
CATEGORY 2
  Variable 1
  Variable 2
etc.

You should define at least one category. When finished press Ok and you will be returned to the Variables editor.

The next step is to assign values to all created variables. To attach any variable to any data field select the variable name in the left hand list then select the desired datafield in the right hand list.

The drop-down list on the right displays the available datasets and the window below shows available data fields of the selected dataset. The last available category in the dataset drop-down list is the "Other". When selecting it, you'll get access to the following data:

- Page# (built-in function returns current Page number).
- Expression allows creation of expression in the "Expression" textbox.
- Date (built in function returns current date).
- Time (built in function returns current time).
- Line# (built in function returns current row number of report dataset).
- Line through# (built-in function returns row through).
- Column# (built-in function returns Column number).
- TotalPages (built in function returns number of pages in report). To use this function, you should set "Two-pass report" option in Report options dialog.

2.4. REPORT OPTIONS

To set Report Options choose "File|Report option" from designer's menu.

The dropdown list at the top of the dialog box lists all available printers on the system. If there is no printer available installed in your system you can select "Default printer" which allows you to use any paper dimension but it will be impossible to print. It will only allow you to work with report in the designer and make preview.

If "Select when report loaded" option is checked then printer information is stored with the report and when this report is loaded that printer will automatically be selected. If that printer is not found on the system then the windows system default printer will be selected.

The "Two pass report" option must be checked if you need to use "total pages" function in the report, i.e. print "Page xx from yy".

When you select a printer the Page window in the report designer shows the available printable area for the Page size and printer selected.

2.5. PAGE OPTIONS

To set the page options for the current page of report, choose "File|Page options" in designer's menu, or Dblclick on an empty spot on the page.

You can select paper format from the drop-down list of paper formats supported by the current printer. If the current printer supports custom paper sizes and you select custom from the list you will be able to enter the width and height of the custom size. Any other choice uses preset sizes. You may also choose the orientation for the Page. Note All printers may not support custom paper sizes, (for example, printer driver "HP LaserJet 6L" does not support page dimensions less than 76 * 127 mm; printer driver "HP LaserJet 4L" does not support custom size at at all).

On "Margins" tab of dialog you may select whether or not to use margins and the size of the margins. If the "Don't use" option is selected, no printable area border will be shown in the designer's Page window, and all page contents will be printed correctly on any printer. But object's sizes will be not the same on different printers.

If you leave the option unchecked and have all the margin settings at zero the pageborder will reflect the maximum printable area for the selected printer. You may find some chopping when you switch between reports designed for one printer to another, particularly between ink-jet and dot matrix. Ink-jet printers usually have a smaller printable area than a dot matrix due to the way they feed the paper.

If margins is set to non-zero values, hash marked border will show in the report designer's page window. If you use dot matrix printers, pay close attention to the printable area: some dot matrix printers will not print if the object to be printed overruns the printable area, others will produce pages

with the over-run printed on them. This obviously produces weird looking reports. In this case, you should set margins manually.

On the “Options” tab of the dialog you may set more options for the page. You may set the number of columns across the page width and the space between them. If “Print on previous page” flag set to true this will allow a new page to start printing on the unused space of the previous page.

2.6. FR OBJECT INSPECTOR

The Object Inspector allows you to manipulate some object properties: set object name, it’s position, dimensions and visibility. Inspector works like Delphi’s Object Inspector. Like the other toolbars, it may be showed or hided. To show Object inspector, choose “Tools|ToolBars|Object Inspector” in designer’s menu. To shrink Object Inspector, dblClick on its title bar; dblClick again and it will expand.

2.7. HIGHLIGHTNING

For those reports , which you would like to have objects change their font color, background color etc. based on a specific condition or expression, click the  button in the Text formatting toolbar. This will bring up the “Highlight attributes” dialog where you may enter the conditions to be satisfied in the text box and select the various options available.

For example, assume it is necessary to pick out orders exceding $1,000.00, Demo report 3-level is an example of this. To achieve this, select rectangle containing sum order, click on the highlight button in the formatting toolbar. In the textbox of the dialog type the condition “Value > 1000” without quotes. Select appropriate font & background color then click Ok. Your report will now show any orders over $1000.00 highlighted with the chosen color. Try various combinations of font styles with background colors to achieve the results you need.

2.8. DESIGNER OPTIONS

To set the default options for the report designer choose “Tools|Options” menu command. Here you can set grid size, report measurement units: pixels, millimeters and inches. Grid size 18 pixels corresponds to 5mm. You can also control how the selected object appears on screen, either frame or inverse color. If Coloured Buttons is turned off, then all Buttons become black-white; Editing after insertion governs the default action when inserting objects as to whether the default editor for the object appears after insertion or not. If inserting a large amount of empty rectangles turn this option off. Snap to grid snaps objects to the nearest grid point when moving them. Note: even though screen measurements may be in units other than mm, when setting margin sizes in Page option they are still in mm.

2.9. EXTRA TOOLS

The report designer may have additional tools that expand the designers environment. They may be called from the tools menu or you may display them in their own toolbar. To display the toolbar, choose “Tools|ToolBars|Tools”.

By default designer have one tool which allows you to place DB fields on the report. Select the fields you wish to insert by holding down the Ctrl key and leftclick on the desired fields. Select the orientation and whether or not to insert headers then click Ok. All the objects will be inserted as a group in the report window with each rectangle containing the correct text or field reference.

3. PREVIEW DESCRIPTION

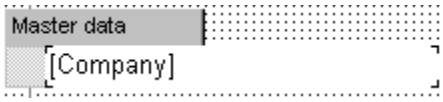
- Open file report: Ctrl+L, or click open button in toolbar.
- Preserve report to file: Ctrl+S, toolbar button. If export filter components have been placed in the underlying Delphi Form then the list box will display available formats.
- Edit report: Ctrl+E, context menu. This command accessible, if underlying Delphi form has the TfrDesigner component.
- Add page: Insert, context menu. This command accessible, if underlying Delphi form has the TfrDesigner component.
- Delete page: Delete, context menu. This command accessible, if underlying Delphi form has the TfrDesigner component.
- Search text: Ctrl+F, toolbar button. Repeated search - F3.
- Print report: Ctrl+P, toolbar button.
- Change scale: click on scale button, context menu, spacebar.
- Rightclick shows context menu.
- Report navigation keys:
  Scroll bars, Arrow keys - window scrolling;
  Page Up, Page Down - up/down (large piece);
  Ctrl+PageUp, PageDown, Click right lower corner - page up/down;
  Home/End - depends on which scrollbar is active;
  Ctrl + Home, End - go to beginning/end of report.

4. CREATING REPORTS

Before creating any report, place TfrDBDataset or TfrUserDataset component on Delphi form. You must have as many TfrxxxDataset components as much data bands in your report, and link it to Delphi's TDataSource->TDataset components (if you dealing with DB data).

4.1. SIMPLE REPORT (LIST)

This is a simplest kind of report. For creating it, you should place «Master data» band on the page, then place objects with appropriate data on it:



You also should connect «Master data» band to appropriate dataset (TfrDBDataset or TfrUserDataset components). Example of this and other reports you can see in demo.

4.2. MASTER-DETAIL REPORT

For creating this report, place «Master data» and «Detail data» bands on the page, then place objects with appropriate data on these bands. Does not matter, in which order you place bands on the page - «Master data» band will be printed first. If appropriate detail list is empty, master record will be skipped. If you don't want this, turn on the option «Print if detail empty» of «Master data» band.

You can print new master data on new page - just turn on «Start new page» option of the band.

4.3. MASTER-DETAIL-SUBDETAIL REPORT

For creating this report, place «Master data», «Detail data» and «Subdetail data» bands on the page, then place objects with appropriate data on these bands. All principles are like to the master-detail report.

4.4. CROSS-TAB REPORT

This report is intended for printing table with variable number of columns. During report building, all off-bounds columns will be printed on new page (like in MS Excel). For creating this report, place «Master data» and «Cross data» bands on the page. Place object in the cross of these bands. This object will be printed as the cell of cross-table. That's all! If you want, place «Cross header» and «Cross footer» bands in your report. «Cross footer» band is usable if you want to print sum of data in the row. «Cross header» band can prints on each page, if it has «Print on all pages» option.

4.5. REPORTS WITH VARIABLE ROW HEIGHT

Height of data row may be variable in some reports. Example of this - bill, with long goods names.

FR can print these reports. If band has variable height, you should turn on the «Stretched» option of the band. You also should turn on this option for all stretched objects in the band. If calculated height of band smaller than its initial height, nothing happens. Otherwise, objects with this option will be stretched - its bottoms will be stretched to band bottom.

4.6. REPORTS WITH MULTIPLE DATA STRINGS

FR don't allow you to place two identical bands on the page. Except of this - you can place many data-bands (master data, detail data, subdetail data) and their headers & footers on the page. During report building, identical bands prints together. This allows you to create reports with data row which consist of several identical data-bands (each of these bands can be of variable height). If entire data row exceeds the bottom of page, it prints at the next page.

All written is right, if top data band has data source, and other not have. Otherwise, we get case of Master-detail-detail report.

4.7. MULTICOLUMN REPORTS

Just set number of columns in «Page options» dialog in designer.

4.8. MULTIPAGE REPORTS

FR can print reports consisting of several pages, e.g. title page and others. For adding and removing page, use toolbar buttons or menu of designer. If you want to print several multipage reports (each report for each data record in some dataset), set TfrReport.ReportType property to rtMultiple, and assign appropriate dataset (TfrxxxDataset) to TfrReport.Dataset property.

4.9. NESTED REPORTS

Nested report is report with «Subreport» object placed in it. «Subreport» object is reference to the other report, placed in other page. During report building, on the place of «Subreport» object appropriate report will be printed.

Subreport objects can be placed side-by-side. If you want place subreport objects one-under-other, spread it on separate data-bands. There is some limitation of using subreports:

- Don’t use columns in subreports.
- Don’t use ReportTitle, ReportSummary, PageHeader, PageFooter, ColumnXXX bands.
- Don’t use breaked bands.
- Don’t use groups.

4.10. MASTER-DETAIL-DETAIL REPORTS

For creating this report, place «Master data» and two «Detail data» bands on the page. Link all bands to the appropriate datasets. You can also create «Master-Detail-Detail-Detail», «Master-Master», «Master-Detail-SubDetail-SubDetail» reports etc.

4.11. COMPOSITE REPORTS

Composite report is report included several other reports. For creating composite report, you should use TfrCompositeReport object. Place it on the form and fill its «Reports» property in run-time by references to the other reports. Reports will be printed sequentially. If page of report have «Print to previous page» option, it will be printed at remained space at the last page of previous report.

4.12. REPORTS WITH BREAKED BANDS

Let's create simple report. It will contain two fields: first, named «Program», and second, named «Description». If we take a look on the created report, we see that page surface used not fully - there is a lot of empty space at the bottom of page. In our case we want to carry long memos onto the next page. This will be done, if «Breaked» option of the band is set.

4.13. REPORTS WITH MEMO FIELDS AND PICTURES

If you want to show BLOb data in the object, insert reference to appropriate field to object's memo. If you want to show BLOb data from non-DB source, you must do this in OnEnterRect event handler. Example of this may be found in demo, «Text file» report.

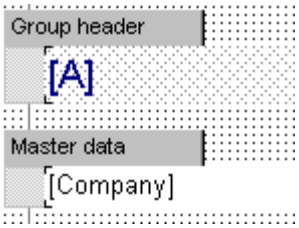
4.14. REPORTS WITHOUT BANDS

If you want print a free-form report, containing data from only one record, you can don't use bands. All the objects placed directly on the page, will be printed at their places.

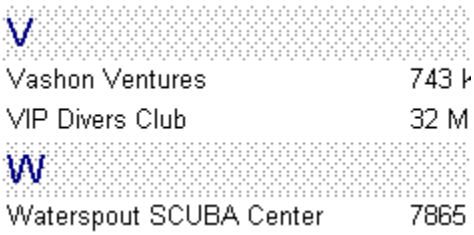
4.15. REPORTS WITH GROUPS

Groups used for grouping data by some criteria. You can use any FR expression as group condition (usually used expression based on DB fields). When this expression changes, FR forms new group.

For creating this report, place «Group header» and «Master data» bands on the page. Assign appropriate dataset to «Master data» band. In editor of «Group header» band type grouping expression. For example, to print customers list grouped on first letter of customer name, type the following expression: Copy([CustomerName], 1, 1), where [CustomerName] is reference to appropriate DB field. Note: dataset used for «Master data» band should be sorted on grouping condition. You can do this by using queries with ORDER BY statement.



will be printed as



There are some limitations of using groups:

- Don’t use groups in the subreports.
- Group can be top-level list only.

4.16. REPORTS WITH CHARTS

To build a report displaying charts it is necessary to have one data band i.e. master connected to appropriate TfrDataset. The band would have various rectangle objects placed on it. If you have used FR’s Object inspector you should have noticed that each rectangle placed on a band receives a name i.e. first rectangle placed in a report is memo1, 2nd is memo2 etc., these are the names you will supply to the chart object. A chart object is placed in it’s own band. Lets assume we have a master data band with a number of memo objects placed on it. Place a second master data band on the report and connect this band to the virtual dataset. On this band place a chart object. In the chart object editor select the type of report and options on the display tab, on the data tab enter the name of the memo to be used for the legends, then enter the name of the memo which has the numerical data you wish to chart. If there is a large number of records you may wish to set the “top ten” group value to a small number. This will cut down the number of items shown on the chart, size the chart rectangle to a reasonable size for displaying the chart preview the report and see your results. This is a brief summary of displaying a chart you may have to use variables in some cases depending on the type of values being supplied in the underlying data.

5. TFRDATASTORAGE COMPONENT

Component TfrDataStorage is intended to create databases, tables and queries in run-time mode, setup of a list of accessible fields, creation of lookup-fields, master-detail relationship. This component allows to execute the same activity, as in Delphi IDE. The information is saved together with the report form.

When component is connected, "Data Manager" and "Parameters dialog" added to the list of Designer Tools. First allows to work with the data - to determine the databases, tables and queries and edit them. The second tool is intended for editing the parameters of the query. All databases, tables and queries created with this tool are in datamodule of «ReportData», which one is accessible in run-time mode. For each table or query there is a data source for FreeReport with the same name, as well as for the table, but signed underlines ahead.

Data manager supports BDE, Interbase Express (IBX) and ActiveX Data Objects (ADO). IBX&ADO does'nt requires BDE. To activate appropriate DB engine, uncomment appropriate line in FR.INC file.

5.1. CREATING DATABASE

Creating database is necessary if you using ADO or IBX data components. Before creating any tables or queries, you should create database first.

To create new database, start the tool "Data Manager". In the appeared window click on "New database" button. After that the dialog box "Database parameters" will open. If you using ADO or IBX, set the name of database component and choose appropriate DB file. If you using BDE, set the name of database component, set alias name, which you want to add to the alias list, and choose appropriate DB driver.

You can also set DB parameters, such as user name and password for connection to SQL-based DBs. Note: different drivers uses different parameter names. For example, to connect to Interbase SQL server through BDE native driver, use the following parameters:

SERVER NAME=Path to your *.gdb file
USER NAME=SYSDBA
PASSWORD=masterkey

To connect to Interbase SQL server through IBX, use the following parameters:

user_name=SYSDBA
password=masterkey

When "Login prompt" checkbox is turned on, database login dialog will appears when you connecting to the SQL server. If you also set user name and password in database parameters, you can turn this option off. This automatically connected to DB.

5.2. CREATING TABLE

For creation of the new table start the tool "Data Manager". In the appeared window click on "New table" button. After that the dialog box "Select table" will open. You can choose necessary table from available DB's or load it directly from the file (only if using local Paradox or DBase tables with BDE). After the table is selected, the dialog box "Table Properties" will open. Here it is possible to set a table name (by default is offered "Table" + first free number). TDataSource and TfrDBDataSet created together with the table. The name of TfrDBDataSet, as well as for the table, but signed underlines ahead.

Also in this window you can execute fields editor, select index for the table (if table have secondary indexes), set filter expression. To set Master dataset for the table you can select necessary data set from a "Master" list, and in line "Field Master" select fields from a Master dataset (or, clicked in a right part of line to call dialog box for visual linkage of fields).

5.3. CREATING QUERY

For creation of the new query start the tool "Date Manager". In the appeared window click on "New query" button. After that the dialog box "Query Properties" will open. Here it is possible to set a query name (by default is offered "Query" + first free number). TDataSource and TfrDBDataSet created together with the query. The name of TfrDBDataSet, as well as for the query, but signed underlines ahead.

The "Alias" drop-down list contains path of the database tables, on which the query is built. Here there can be a value from a drop-down list or path to the catalog with the tables. It is a field it is possible to leave empty, in this case alias or path it is necessary to indicate in the text of the query. After that it necessary to type the text of the inquiry in a field "SQL Text". If in the text of the query consist the parameters (i.e. identifiers signed ":" in a start of a name), each parameter is necessary for describing in the editor of parameters.

Also in this window you can execute fields editor and set Master dataset. To set the Master dataset for the query you can select from a list "Master" the necessary data set. Thus text of the query

should be contained parameters - names of fields from a Master-source. Besides each parameter should have an option "Assign from a Master-source".

At clicking on "OK" button the regularity of the query is checked up. If in the text of the query or in the description of parameters there is an error, the applicable warning will be issued and the editing of properties of the query will proceed.

5.4. FIELDS EDITOR

In this editor is possible to fill in a list of accessible fields of the table or query. If the list is empty, all fields will be accessible.

To add fields to the list click on "Add fields" button. From the appeared window select fields and click "OK" button. For selection of group of fields click the mouse on the first field of group, then at the pushed button "Shift" - on last. To add a lookup-field, use the "Add lookup" button. See "Creating lookup field".

For deleting selected fields from a list use "Delete" button.

5.5. CREATING LOOKUP FIELDS

For what the lookup-fields necessary, we shall explains on an example.

Let's allow, there are two database tables: the table "Orders" with fields N, Date, ClientID, Amount and table "Clients" with fields ID, Name, Address. Table "Orders" is contained the information on the orders (number, date, identifier of the client who has made the order, and sum of the order). Table "Clients" is contained the information on clients (identifier of the client, full name, address). To create the elementary report on the table "Orders" like Number of the order - Date - Name of the client - Sum, it is required to link both tables on fields ClientID - > ID. This problem is decided by creation of a lookup-field, which one is added to the table "Orders" and represents the link on a field Name of the table "Clients".

Dialog box of creation of a lookup-field is accessible from the editor of fields. For creation of a lookup-field it is necessary to set it name and type, and also the size (in case is selected the string type). Further it is necessary to fill in following fields:

- Primary key field - field in an source dataset, which one as the link on a field from a lookup-dataset. In our example it will be by a field ClientID.
- Datasource - lookup-data set.
- Lookup key - field in a lookup-dataset being key. In our example it is a field ID.
- It is necessary to substitute a resultant field - field of a lookup-set, which one in an source data set. In our example it is a field Name.

After that in the table "Orders" there is a dummy field with a given name, which one contains decryption of a field ClientID. You can access to this field, as to a common field, but only for reading.

5.6. QUERY PARAMETERS EDITOR

In this dialog box is possible to assign the type to each parameter, and also to point, whence to take an parameter value: from a master-dataset, from dialog box, or at once to assign particular value. In a case, when parameter takes from a master-dataset, the dataset should contain a field with a name conterminous to a name of parameter. Besides it is a field should be in a list of accessible fields (see fields editor). Thus it is optional to indicate the type of parameter.

5.7. JOINING DATA

In this dialog box is possible visually to link fields master and detail of data sets. When the data sets are connected with each by Master-Detail, at moving on master data set the detail data set is filtering. So it consist only the records that have relation to master data set. For link fields of data sets select fields from list at left side (detail dataset), then select field from list at right side (master dataset),

and click "Add" button. Thus the link fields moved to the lower list. To clear lower list push "Clear" button. The linked fields should have the identical type and to be primary key.

5.8. PARAMETERS DIALOG

If the report contains as the data source one or several queries with parameters, which one have an option "Request value" (see editor of query parameters), before creating the report on a screen will be show dialog box of parameters input values. In dialog box all parameters will be collected, which one meet in all queries indispensable for construction of the report.

By default for input the value of parameter used command element "Edit box", which one has an description - name of parameter. If some parameters, the command elements are injected the self under the self. It is inconvenient - because parameter name is abbreviated English word. In this case it is possible to take advantage of the Dialogs designer.

The dialog box will be accessible, if the report is true in it requires. The empty report or report which is not inclusive of the links to the queries with parameters, does not show dialog box.

5.9. DESIGNER OF PARAMETERS DIALOG

The Dialogs designer is very good means for change appearance of Dialog box of parameters input values. It allows to arrange control objects of dialog box in the necessary order, to change the descriptions (by default used parameters name). Besides the designer allows to replace the type of a command element for parameters input values - by default command element is "Edit box".

The designer does not allow to add new objects or to remove existing. It is possible only to change their position and sizes, to set properties (for example, parameters font). All operations are make by the mouse. The mouse works like in designer of report form.

To set properties of object, make a double click by the mouse on a them. In the appeared dialog box it is possible to set the text, which one will be showing in object. The conversation also can be called, if some one-type objects are selected. If the selected object - command element, than in properties dialog box becomes accessible the switch "Object type".

It is possible to select three types:

- Edit box - represents line, in which one it is possible to type any text. If in a field "Text" of dialog box something is typed, this value will be imaged in edit box of input dialog box.

- The list - represents a drop-down list. The useful thing, if is necessary to submit the user selection from several values. If the line of a list contains a character ";", in a list the part of line up to a semicolon will be show, and by selection of this value actually in parameter the part of line after a semicolon will be show. In such a way conveniently, for example to select month: at usage of a list with lines such as "January; 1", "February; 2" in a list the titles of months will be show, and in parameter numbers will get.

- Lookup the list - is convenient for using for selection of value from the reference book (database tables with a primary key field and field inclusive a title). For setup lookup it is necessary to select the table, its primary key field, which one will be substituted in parameter, and the field, which one will be show in drop down list.

It is possible to change the sizes of the form - it is necessary only to allow, that after execute below of dialog box will be added button "OK".

6. PROGRAMMING

6.1. USING INFORMATION FROM NON DATABASE SOURCES. EVENTS

Frequently you may find the need to extract data from other sources, not having a relation to databases (for example, file, array or string grid). This is the purpose of the TfrUserDataset component, it generates the events OnFirst, OnNext, OnCheckEOF to handle navigation in this type of data. You should also create event handlers for the OnGetValue and OnEnterRectangle events of the TfrReport component.

An event, OnGetValue, is triggered each time, a Text object encounters a variable and it needs to be given a value. If the variable encountered has a known value, it is processed internally. Otherwise it must be handled externally, for example:

```
procedure TForm1.Doc1GetValue(const ParName: string; var ParValue: Variant);
begin
  if ParName = 'Var1' then
    ParValue := '1'
  else if ParName = 'Var2' then
    ParValue := 2
end;
```

OnEnterRect event handler called each time before object is drawn. It is most useful for filling object's memo or picture (for "Picture" object). This is an example of OnEnterRect event handler:

```
procedure TForm1.Doc1EnterRect(Memo: TStringList; View: TView);
begin
  if Memo.Count > 0 then
    if Memo[0] = '[Memo]' then
      Memo.Assign(Table1Memo)
    else if (Memo[0] = '[Picture]') and (View is TPictureView) then
     (View as TPictureView).Picture.Assign(Table1Picture);
end;
```

An OnUserFunction event handler is called whenever a function is found in expression or variable containing function call. Function may take up to three parameters, which are listed in elements p1…p3. This is an example of OnUserFunction event handler:

```
procedure TForm1.Doc1UserFunction(const name: string; p1, p2, p3: Variant;
 var val: string);
var
  d: Double;
begin
 if name <> 'CRAZYMONEY' then exit;
 d := Parser.Calc(p1);
 // To use parser, include FR_Pars in your uses clause
 if d > 10000 then
   val := '''' + FormatFloat('#,##0.00',d) + ' - wow!' + ''''
 else
   val := '''' + FormatFloat('#,##0.00',d) + '''';
 // result is formatted string
end;
```

6.2. DESCRIPTION OF BUILT-IN FUNCTIONS

FreeReport contains expression parser, which goal is to calculate values passed in the text string like «1 + 2 * (3 + 4)». It processes mathematical and logical expressions, string operations, external variables and functions.

Parser uses 1 and 0 instead of True and False. It has two properties for event handlers - OnGetValue and OnGetFunction, identical to TfrReport events.

Parser module can be used independently of FR, just add FR_Pars to the **uses** clause of the module you wish to use it with.

Using strings and variables

You can use string constants in expressions. As in Pascal, string is a sequence of characters, rounded by single quotes, for example: ‘Hello, World!'. If a string contains a single quote, it is necessary to duplicate it: ‘What's new' becomes ‘What''s new'.

You can also use variables or DB fields in expressions, for example: “[Total amount] / 2” is a valid expression. A valid variable can be a field from any accessible table. Reference to a field should have the following format: [FormName.TableName.”FieldName”], where FormName is form name or data module name, where the table is located; TableName is name of the table; FieldName is a field name in the table. If a field name does not contain blanks, then the quotes may be omitted. If the parameter FormName is not used, FR searches for the table on the current form (or data module). If the parameter TableName is not used, FR search for FieldName in current table (current for band, in which expression is contained).

Aggregate functions*

Aggregate functions can be used in ReportSummary, PageFooter, MasterFooter, DetailFooter, SubdetailFooter, GroupFooter, CrossFooter bands.

- *Sum(<expression> [, band] [,1]).* Calculates the sum of the values passed in expression for band row given. If the band parameter is not set, sum defaults to all of data values (on the bands MasterData, DetailData, SubdetailData); otherwise a sum refers only to data on the named band. If “1” parameter is used, calculates sum for non-visible objects too. Example: Sum([Part total], Band1), Sum([Part total] + [Part price]).
- *Sum(<field>). A Sum* of a named field. A field Name can be short, or long, containing a form name and datamodule set (in which case the names are separated by dots). A sum performs for all of the named datasets records. Example: Sum(CustomerData.Customer.CustNo), Sum(Table1.Amount).
- *Avg, Min, Max.* Syntax is analogous to the Sum function. Function Avg calculates average arithmetic, function Min returns minimum, function Max is maximum value from row.
- *Count[band].* Returns count of strings given. Example: Count(Band1).
- *Count(<dataset>).* Returns amount of records in a named dataset. A named dataset’s Name can contain the form name, on which it is located. Example: Count(CustomerData.Customer), Count(Table1).

String functions

- *Str(<value>).* Converts number given in value to a string.
- *Copy(<string>, <from>, <count>).* Returns portion of string # of char in count, starts at from, (same as Delphi function).
- *If(<expression>, <string1>, <string2>).* Returns string string1, if expression expression true; otherwise returns string string2.

- *FormatFloat(<formatstr>, <value>)*.* Converts a numerical significance value in string, making use of mask in formatstr. The possible values of formatstr described in Delphi documentation, «Formatting strings» topic.
- *FormatDateTime(<formatstr>, <value>)*.* Converts a date/time value in string, making use of mask in formatstr. The possible values of formatstr described in Delphi documentation, «Formatting strings» topic.
- *StrToDate(<value>)*.* Converts string in value to date**.**
- *StrToTime(<value>)*.* Converts string invalue to time.
- *UpperCase(<value>)*.* Converts the string symbols to upper case.
- *LowerCase(<value>)*.* Converts the string symbols to lower case.
- *NameCase(<value>)*.* Converts the string symbols to lower case, and first symbol is in upper case.

Arithmetic functions

- *Int(<value>).* Returns whole part of number value.
- *Frac(<value>).* Returns fractional part of number.
- *Round(<value>).* Returns rounded off significance.
- *value1 Mod value2.* Returns remainder from dividing by value1.

Other functions*

- *Input(<caption> [,<default>]).* Shows dialog window with headstring “caption” and edit box. If “default” parameter is set, puts this string in edit box. After user clicks OK, returns input string.
- *Date.* Returns date.
- *Time.* Returns time.
- *Line#.* Returns information about Line number; enumeration begins with group beginning. For example:

  Master data
  1. Detail data
  2. Detail data
  3. Detail data

  Master data
  1. Detail data
  2. Detail data
- *LineThrough#.* Returns information about Line number; enumeration begins with report beginning, for example:

  Master data
  1. Detail data
  2. Detail data
  3. Detail data

  Master data
  4. Detail data
  5. Detail data
- *Column#.* Returns current column number in cross-tab report.
- *Page#.* Returns current page number.
- *TotalPages.* Returns total amount of pages in formed report. For use of this function a report must be two pass report.

Note 1. Do not leave space between function name and its opening braket.
Note 2. Functions, marked by sign «», is not built into parser (module FR_Pars).*

6.3. INTERPRETER DESCRIPTION

Interpreter basics

FreeReport contains built-in interpreter for handling of report building. Interpreter based on Pascal-like language with the following features:

- operators: assignment (:=), if…then…else, while…do, repeat…until;
- begin…end blocks;
- untyped variables;
- accessing to the FR's objects and its properties.

Each object (visual object or band) can contain one or more operators (script). Script editor accessible in memo editor when "Script" check is turn on. Script runs each time before object printed. You can use the following properties of visual objects:

- Left, Top, Width, Height – position and sizes;
- Flags – object's options;
- Visible - visibility;
- FrameTyp – type of object's frame (1 – right line, 2 - top, 4 - left, 8 - bottom);
- FrameWidth, FrameColor – width and color of object's frame;
- FillColor – background color of object;
- Text - Memo contents.

Text objects has additional fields:

- FontName, FontSize, FontColor – name, size and color of the font;
- FontStyle – font style (1 - italic, 2 - bold, 3 - underline);
- Adjust – text alignment (1 - right, 2 - center, 3 - width, +4 – vertical text, +8 – vertical centering, +16 – down align).

You can also use variables in script. Variable can be of any type, with the name of max. 32 chars length. Name can consists of latin letters, digits and underscore. This variable can be showed in visual objects; you also can use report variables or DB fields in script. Script variables stored internally in the TfrVariables object. You can access to it by frVariables global variable declared in the FR_Class unit.

If script contains more than one operator, you should use begin…end brakets, for example:

```
begin
  FillColor := clGreen;
  FontColor := clWhite;
end
```

Using interpreter

Here is some examples of using interpreter:

1. To highlight sum of order: white, if sum less than 2000; green, if sum is between 2000 and 10000; red, if sum is greather than 10000, type the following script in object with sum:

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

[Summa] is your variable or field with actual sum of order. You can use numeric constants for choosing color:
`FillColor := 128 + 128*256 + 128*65536` (gray color).

2. To show only data rows with order sum greather than 2000, use the following code in the script of appropriate data-band (you can show its memo editor by Ctrl+Enter key):

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

3. To show group condition in group footer, use the following code in group header band:

```
Condition := [Your_group_condition]
```

and place the following string to the memo of “Text” object of group footer:

Total of [Condition]

4. To set value of script variable programmatically, use frVariables reference, which defined in FR_Class unit. For example:

```
frVariables['SelectedMonth'] := 'April';
frReport1.ShowReport;
```

Then, place the following line in desired “Text” object:

Selected month: [SelectedMonth]