

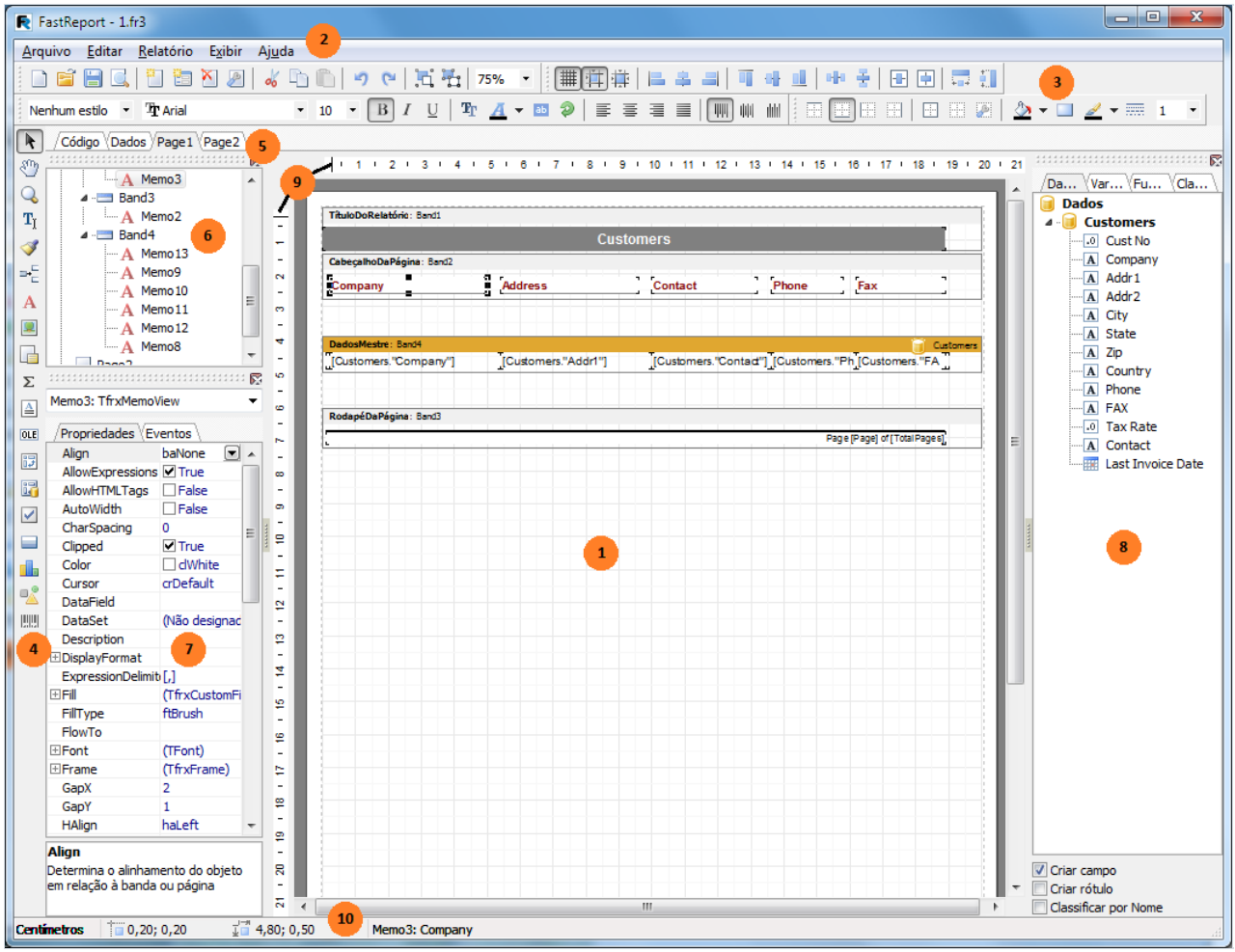
Manual do Usuário FastReport FMX

Version 2026.2

© 1998-2026 Fast Reports Inc.

Designer

O designer fornece todas as ferramentas necessárias ao usuário para criar e visualizar relatórios. A interface do designer cumpre com os requisitos atuais. Ela contém diversas barras de ferramentas que podem ser encaixadas onde for desejado. As posições das barras de ferramentas são restauradas cada vez que o designer for aberto. As posições das barras de ferramentas, além de outras configurações do designer, são armazenadas no registro ou em um arquivo .ini, se algum for atribuído.



Legenda dos recursos do designer de relatórios:

1 – área de trabalho para criar o design do relatório

2 – barra do menu

3 – barras de ferramentas

4 – barra de ferramentas de objetos

5 – guias de páginas do relatório

6 – painel "Árvore do Relatório"

7 – painel "Inspetor de Objetos"

8 – painel "Árvore de Dados": os elementos podem ser arrastados deste painel para a página do relatório

9 – réguas: uma régua pode ser arrastada para a página do relatório para criar uma linha guia azul na página (os objetos se ajustam a linhas guia próximas)

10 – barra de status

Teclas de controle

Keys	Description
Ctrl+O	Comando de menu "Arquivo > Abrir..."
Ctrl+S	Comando de menu "Arquivo > Salvar"
Ctrl+P	Comando de menu "Arquivo > Visualizar"
Ctrl+Z	Comando de menu "Editar > Desfazer"
Ctrl+C	Comando de menu "Editar > Copiar"
Ctrl+V	Comando de menu "Editar > Colar"
Ctrl+X	Comando de menu "Editar > Cortar"
Ctrl+A	Comando de menu "Editar > Selecionar Tudo"
Arrow, Tab	mover entre objetos
Del	excluir objetos selecionados
Enter	abrir editor para o objeto selecionado
Shift+arrows	alterar o tamanho de objetos selecionados
Ctrl+arrows	mover objetos selecionados
Alt+arrows	anexar o objeto selecionado a um objeto adjacente na direção especificada

Controle do mouse

Operação	Descrição
Botão esquerdo	selecionar objeto; colar objeto novo; mover ou redimensionar objetospara objetos selecionados, amplie ou reduza o zoom ao arrastar o quadrado vermelho no canto inferior esquerdo do grupo de objetos selecionados
Botão direito	menu de contexto do objeto selecionado
Dois cliques	abrir editor do objetoclicar duas vezes em espaço em branco abre a caixa de diálogo "Opções da Página"
Roda do mouse	rolar página do relatório
Shift + botão esquerdo	alternar a seleção de objetos
Ctrl + botão esquerdo	criar um quadro ao mover o mousesolte o botão para selecionar todos os objetos capturados no quadrotambém é possível clicar em um espaço em branco e mover o mouse conforme necessário
Alt + botão esquerdo	editar o conteúdo no local se um objeto "Texto" for selecionado

Barras de ferramentas

Barra Modo do designer

A barra de ferramentas Modo é integrada à barra de ferramentas Objeto ao lado esquerdo da janela do designer, e possui os seguintes botões:

Ícone	Nome	Descrição
	Selecionar Objetos	modo padrão de operação: o cursor do mouse seleciona os objetos, altera seu tamanho, etc.
	Mão	permite arrastar toda a página do relatório
	Zoom	permite o clique com o botão esquerdo para ampliar o zoom (adiciona 25% se já estiver no intervalo 25-75%, de outro modo adiciona 100%) ou o clique com o botão direito para reduzir o zoom (reduz 25% se já estiver no intervalo 25-75%, de outro modo reduz 100%)manter o botão esquerdo do mouse apertado ao arrastar amplia o zoom na área selecionada
	Editar Texto	permite a edição no local de objetos "Texto"manter o botão esquerdo do mouse apertado e mover o cursor cria e dimensiona um novo objeto "Texto" e abre seu editor
	Copiar Formatação	permite a cópia da formatação de um objeto "Texto" para outros:selecione um objeto "Texto",clique no botão Copiar Formatação,selecione os objetos-alvosaia do modo Copiar Formatação ao clicar em qualquer outro botão de modo

Barra de ferramentas padrão



Ícone	Nome	Descrição
	Novo Relatório	cria um relatório em branco
	Abrir Relatório	abre o arquivo de um relatório existentecombinação de atalho – "Ctrl+O"
	Salvar Relatório	salva o relatório em um arquivocombinação de atalho – "Ctrl+S"
	Visualizar	visualiza o relatório combinação de atalho – "Ctrl+P"
	Nova Página de Relatório	adiciona uma página nova ao relatório
	Nova Página de Diálogo	adiciona um formulário de diálogo novo ao relatório
	Excluir Página	exclui a página atual
	Opções de Página	abre a caixa de diálogo de propriedades da página

Ícone	Nome	Descrição
	Recortar	Recorta os objetos selecionados para a área de transferênciacombinação de atalho – "Ctrl+X"
	Copiar	copia os objetos selecionados para a área de transferênciacombinação de atalho – "Ctrl+C"
	Colar	cola objetos da área de transferênciacombinação de atalho – "Ctrl+V"
	Desfazer	desfaz a última operaçãocombinação de atalho – "Ctrl+Z"
	Refazer	refaz a última operação cancelada (desfeita) combinação de atalho – "Ctrl+Y"
	Agrupar	agrupa objetos selecionados
	Desagrupar	desagrupa objetos selecionados
75%	Zoom	configura o fator de zoom

Barra de ferramentas Texto



Ícone	Nome	Descrição
Sem estilo	Estilo	exibe o estilo do objeto "Texto" selecionadoselecione o estilo em uma lista suspensa para alterar o estilo do objeto selecionadodefina estilos em "Relatório > Estilos"
Arial	Nome da Fonte	exibe a fonte do objeto "Texto" selecionadoselecione a fonte em uma lista suspensa para alterar a fonte do objeto selecionadoexibe as últimas cinco fontes usadas no início da lista
10	Tamanho da Fonte	exibe o tamanho da fonte do objeto "Texto" selecionadoselecione o tamanho em uma lista suspensa para alterar o tamanho da fonte do objeto selecionadotambém é possível digitar números diretamente na caixa de edição
	Negrito (alternar)	ativa/desativa o estilo Negrito da fonte do conteúdo do objeto "Texto"
	Itálico (alternar)	ativa/desativa o estilo Itálico da fonte do conteúdo do objeto "Texto"
	Sublinhado (alternar)	ativa/desativa o estilo Sublinhado da fonte do conteúdo do objeto "Texto"
	Configurações da Fonte	abre a caixa de diálogo Configurações da Fonte
	Cor da Fonte	suspende a grade de seleção de cor da fonte
	Realce	abre a caixa de diálogo de realceé possível configurar uma condição para o realce

Ícone	Nome	Descrição
	Rotação do Texto	modifica a rotação do texto a partir de uma lista suspensa
	Alinhar à Esquerda	alinha o texto à esquerda dentro do quadro do objeto "Texto"
	Centralizar	alinha o texto ao centro dentro do quadro do objeto "Texto"
	Alinhar à Direita	alinha o texto à direita dentro do quadro do objeto "Texto"
	Justificar	justifica o texto dentro do quadro do objeto "Texto"
	Alinhar Acima	alinha o texto (vertical) à parte superior dentro do quadro do objeto "Texto"
	Centralizar Verticalmente	alinha o texto (vertical) ao meio dentro do quadro do objeto "Texto"
	Alinhar Abaixo	alinha o texto (vertical) à parte inferior dentro do quadro do objeto "Texto"

Barra de ferramentas Borda



Ícone	Nome	Descrição
	Linha Acima (alternar)	ativa/desativa a linha superior da borda do objeto "Texto"
	Linha Abaixo (alternar)	ativa/desativa a linha inferior da borda do objeto "Texto"
	Linha à Esquerda (alternar)	ativa/desativa a linha esquerda da borda do objeto "Texto"
	Linha à Direita (alternar)	ativa/desativa a linha direita da borda do objeto "Texto"
	Todas as Linhas da Borda	ativa todas as quatro linhas da borda do objeto "Texto"
	Sem Borda	desativa todas as quatro linhas da borda do objeto "Texto"
	Editor de Borda	invoca a caixa de diálogo do editor de borda
	Cor do Plano de Fundo	define a cor do plano de fundo a partir de uma lista suspensa

Ícone	Nome	Descrição
	Editor de Preenchimento	invoca a caixa de diálogo do editor de preenchimento
	Cor da Borda	suspende a grade de seleção de cor da linha da borda
	Estilo da Borda	suspende a lista de seleção do estilo da linha da borda
1	Largura da Borda	define a largura da linha da borda a partir de uma lista suspensatambém é possível digitar números (0.1 até 10) diretamente na caixa de edição

Barra de ferramentas Paleta de Alinhamento

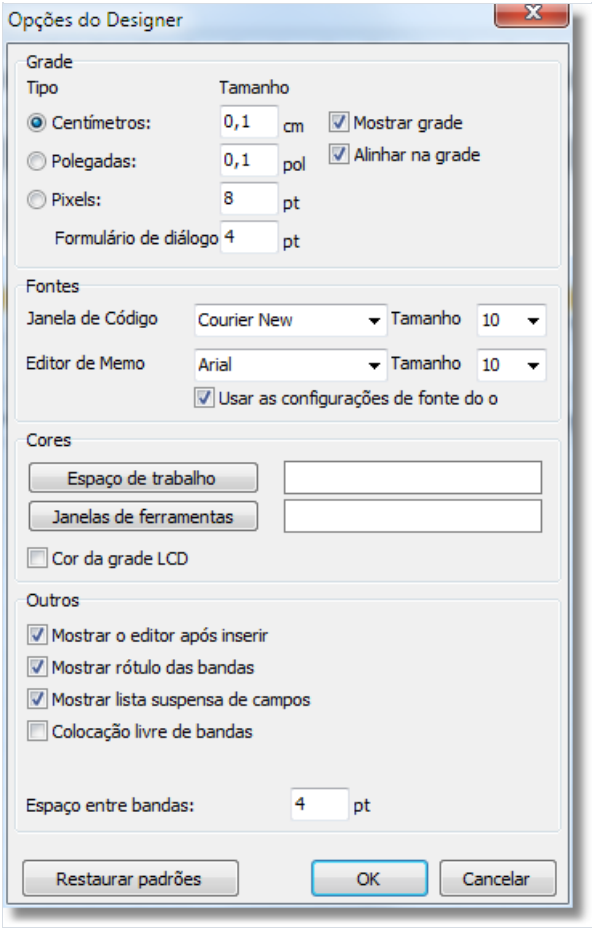


Observe que alguns botões ficarão ativos quando mais de um objeto for selecionado.

Ícone	Descrição
	mostrar grade (ativar/desativar)
	alinhar à grade (ativar/desativar)
	ajustar à grade
	alinhar bordas esquerdas (ao primeiro objeto selecionado)
	alinhar centros horizontais (ao primeiro objeto selecionado)
	alinhar bordas direitas (ao primeiro objeto selecionado)
	alinhar bordas superiores (ao primeiro objeto selecionado)
	alinhar centros verticais (ao primeiro objeto selecionado)
	alinhar bordas inferiores (ao primeiro objeto selecionado)
	espaçar igualmente na direção horizontal
	espaçar igualmente na direção vertical
	centrar cada objeto individualmente de maneira horizontal na banda
	centrar cada objeto individualmente de maneira vertical na banda
	definir larguras iguais (igual ao primeiro objeto selecionado)
	definir alturas iguais (igual ao primeiro objeto selecionado)

Opções do designer

Configure as opções do designer através do comando de menu "Exibir > Opções...".



Aqui é possível definir as unidades preferenciais (centímetros, polegadas, pixels) e o espaçamento da grade. Dica: também é possível alterar as unidades no designer ao clicar duas vezes na parte esquerda da barra de status, onde são exibidas as unidades atuais.

É possível configurar a visibilidade e o alinhamento à grade. Isso também pode ser efetuado através de botões na barra de ferramentas "Padrão", dentro do próprio designer.

É possível configurar uma fonte para a janela do editor de código e o editor do objeto "Texto". Se a opção "Usar as configurações de fonte do objeto" for habilitada, a fonte na janela do editor de texto corresponderá com a fonte do objeto que está sendo editado.

O plano de fundo branco padrão do espaço de trabalho do designer e das janelas de ferramentas pode ser modificado com os botões "Espaço de trabalho" e "Janelas de ferramentas".

A opção "Cor da grade LCD" aumenta um pouco o contraste das linhas de grade e melhora sua visibilidade em monitores LCD.

A opção "Mostrar o editor após inserir" controla o que acontece quando objetos novos são inseridos. Se a opção for habilitada, o editor será exibido cada vez que um objeto for inserido. É recomendado desabilitar a opção ao criar uma grande quantidade de objetos em branco.

Desabilitar a opção "Mostrar rótulos das bandas" oculta os rótulos das bandas, ocupando assim menos espaço na página de design. Se forem desabilitados, os rótulos das bandas são exibidos dentro do espaço de trabalho das bandas.

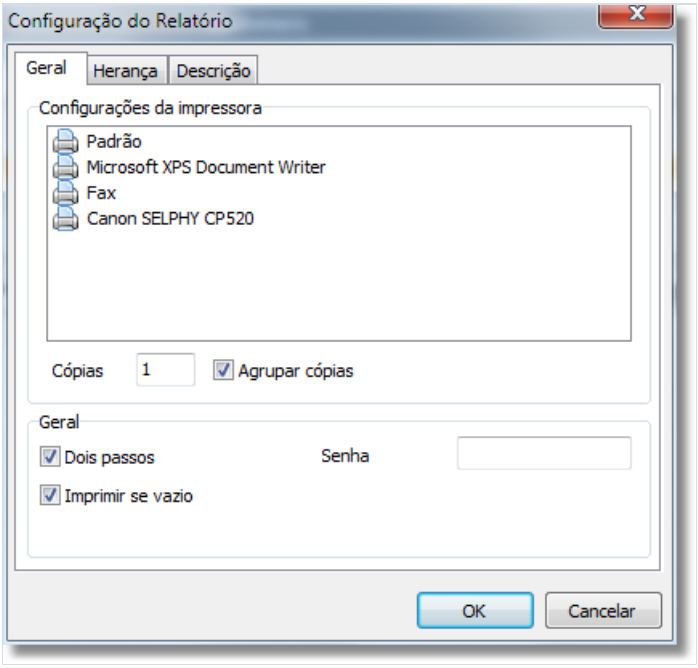
A opção "Exibir lista suspensa de campos" controla a acessibilidade da lista suspensa ao colocar o mouse sobre objetos "Texto" que estão conectados a campos de dados. Isso pode ser necessário se houver muitos objetos "Texto" estreitos em uma banda.

A opção "Colocação livre de bandas" desabilita o ajuste de bandas à página. Esta opção permanece desabilitada por padrão, e as bandas são agrupadas automaticamente na página de acordo com sua função. O espaço entre as bandas é definido no campo "Espaço entre bandas".

Configuração do Relatório

A caixa de diálogo Configuração do Relatório pode ser acessada no menu "Relatório > Opções...". A caixa de diálogo possui três guias.

Na primeira guia é possível ver as configurações gerais do relatório:



É possível vincular um relatório a uma das impressoras instaladas no sistema. Isso significa que a impressora selecionada se tornará a impressora padrão ao imprimir esse relatório. Isso pode ser útil quando houver diversas impressoras no sistema; por exemplo, documentos de texto podem ser vinculados a impressoras monocromáticas, e documentos com gráficos podem ser vinculados a impressoras coloridas. "Padrão" é listado nas configurações da impressora. Quando esta opção estiver selecionada, o relatório não será vinculado a nenhuma impressora específica, mas será impresso na impressora padrão do sistema.

Também é possível configurar o número de cópias do relatório a serem impressas, e se elas devem ser agrupadas. O valor definido nesta caixa de diálogo será mostrado na caixa de diálogo "Imprimir" ao imprimir o relatório.

Se o sinalizador "Dois passos" estiver marcado, o relatório será gerado em duas etapas. No primeiro passo é criado um rascunho do relatório que é dividido em páginas, mas não estará disponível para visualização. No segundo passo o rascunho do relatório é convertido em um relatório padrão, que é salvo no fluxo de visualização.

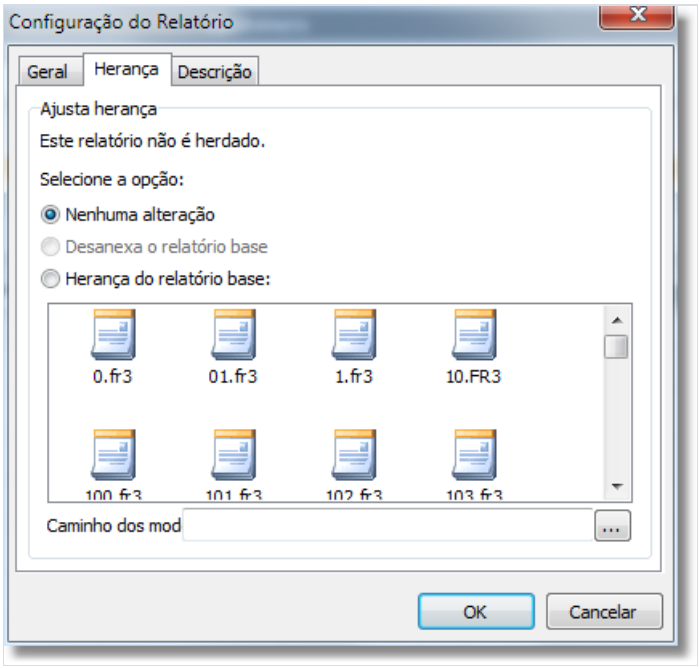
Por que são necessários dois passos? Na maioria das vezes esta opção é usada quando um relatório precisa ter acesso ao número total de páginas no relatório, por exemplo, para exibir "Página 1 de 15". O número total de páginas é calculado durante o primeiro passo e é disponibilizado através da variável de sistema "TOTALPAGES". O erro mais frequente é tentar utilizar esta variável em um relatório de passo único, quando a variável retornará 0.

Outro uso para os dois passos é ao executar cálculos no primeiro passo e exibir os resultados no segundo passo. Por exemplo, quando uma soma deve ser exibida no cabeçalho de um grupo em vez de no habitual rodapé do grupo. Cálculos deste tipo são executados ao escrever o código do script do relatório no evento "OnBeforePrint" de um objeto.

O sinalizador "Imprimir se vazio" permite a criação de um relatório que não contém nenhuma linha de dados. Se esta opção for desabilitada, relatórios vazios não serão criados.

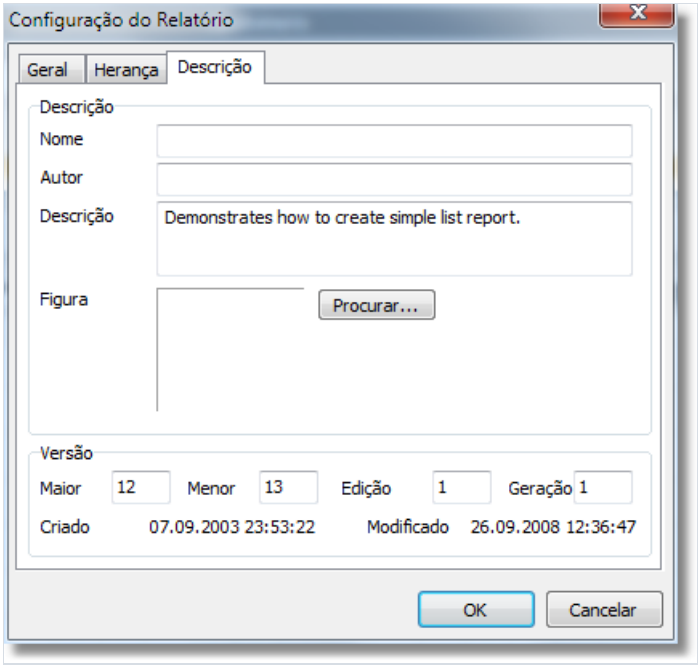
O campo "Senha" habilita a definição de uma senha, que deve ser inserida antes de abrir um relatório.

Na segunda guia é possível configurar as opções de herança:



Mais tarde, no capítulo "Herança do relatório", você obterá mais informações sobre herança. Nesta caixa de diálogo é possível ver o nome do relatório base (se o relatório for herdado), desanexar o relatório base (neste caso seu relatório será autônomo e não herdado) ou herdar o relatório a partir de um dos relatórios base disponíveis.

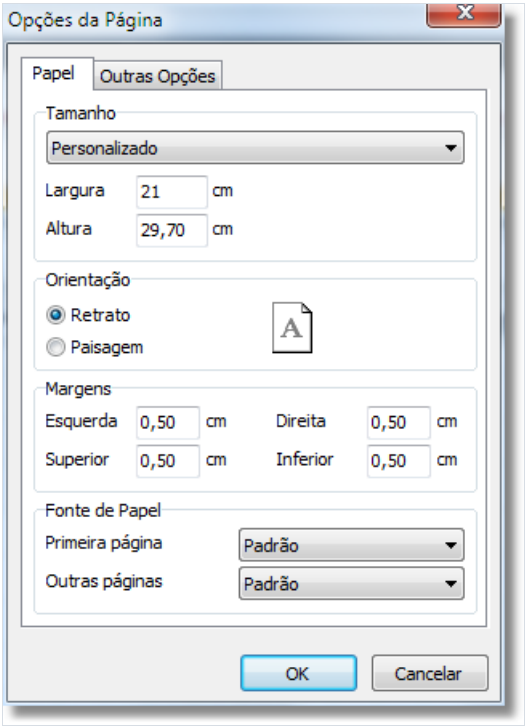
A terceira guia da caixa de diálogo permite configurar propriedades descritivas do relatório:



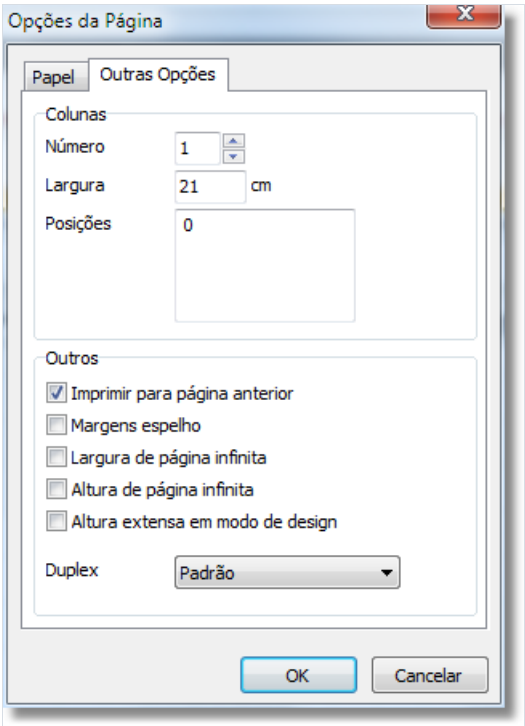
Todos os campos nesta guia servem apenas para fins informativos e não podem ser impressos.

Opções da Página

As "Opções da Página" estão disponíveis através do menu "Arquivo > Configurações da Página..." ou ao clicar duas vezes em um espaço em branco na página. A caixa de diálogo possui duas guias:



Na primeira guia da caixa de diálogo é possível configurar o tamanho e o alinhamento da folha, assim como as margens. Em "Fonte de Papel", a lista suspensa permite a seleção da bandeja da impressora para a primeira página e o resto das páginas do relatório.



Na segunda guia da caixa de diálogo é possível definir o número de colunas para relatórios com várias colunas. As configurações atuais também são exibidas no designer.

O sinalizador "Imprimir para página anterior" permite imprimir páginas sem iniciar uma página nova. Em vez disso o começo será em qualquer espaço em branco na página anterior. Esta opção pode ser útil quando o modelo de um relatório contém diversas páginas ou ao imprimir relatórios em lote (compostos).

A opção "Margens espelho" troca as margens esquerda e direita de páginas pares durante a visualização ou impressão de um relatório.

As opções "Largura e altura de página infinita" aumentam o tamanho da página dependendo do número de registros de dados na página (quando o relatório for executado). Se forem habilitadas, o relatório irá ser exibido em uma página grande na janela de visualização, em vez de diversas páginas de tamanho padrão.

A opção "Altura extensa em modo de design" aumenta a altura da página no designer. Este recurso pode ser útil se houver muitas bandas em uma página. Ele deve ser usado ao trabalhar com bandas sobrepostas. Ele afeta somente a altura da página no modo de design.

Criar relatórios

Objetos do relatório

FastReport Designer representa relatórios como uma coleção de páginas esquemáticas. Objetos são colocados em qualquer lugar nas páginas do relatório e são usados para definir a aparência do relatório e exibir diversas informações, como texto e gráficos. Estes são os objetos FastReport que são incluídos no pacote padrão:

 - Objeto "Banda": uma área na página do designer que se comporta de acordo com seu tipo (por exemplo, Banda de cabeçalho, Banda de dados)

 - Objeto "Texto": exibe uma ou mais linhas de texto dentro de uma área retangular

 - Objeto "Figura": exibe um arquivo gráfico no formato "BMP", "JPEG", "ICO", "WMF" ou "EMF"

 - Objeto "Linha": exibe uma linha horizontal ou vertical

 - Objeto "Texto do sistema": exibe informações do sistema (data, hora, número da página, etc.) ou valores agregados

 - Objeto "Sub-relatório": permite inserir uma página de design de outro relatório na página host

 - Objetos da categoria "Desenhar": exibem diversas formas geométricas (linha diagonal, retângulo, retângulo arredondado, elipse, triângulo e diamante)

 - Objeto "Gráfico": exibe dados em diversos formatos gráficos (gráfico de pizza, histograma, etc.)

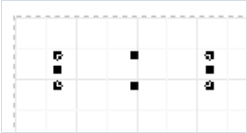
 - Objeto "Caixa de Seleção": exibe uma caixa de seleção com uma marca de seleção ou uma cruz

 - Objeto "Código de Barras": exibe os dados como um entre diversos tipos de códigos de barra

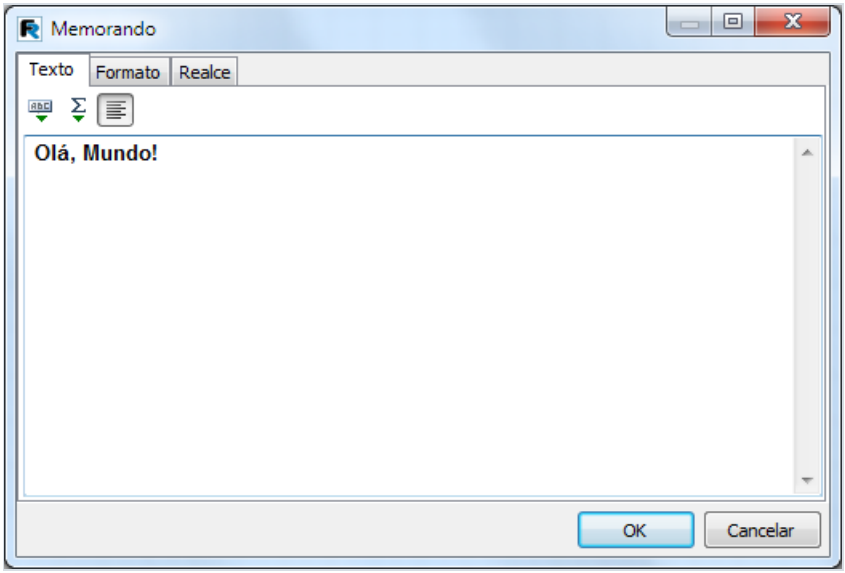
Os objetos básicos usados com maior frequência são os objetos "Banda" e "Texto". Você saberá mais sobre suas funcionalidades mais adiante neste capítulo.

Relatório de exemplo "Olá, Mundo!"

O relatório de exemplo irá conter apenas um item de informação: "Olá, Mundo!". Abra o designer de relatórios e clique no botão "Texto" na barra de ferramentas de Objetos. Mova o cursor do mouse sobre a página e clique novamente. O objeto é inserido na posição do mouse.



A janela do editor de texto é aberta imediatamente. Se ela não aparecer automaticamente, clique duas vezes no objeto. A abertura do editor de texto pode ser configurada nas configurações do designer – isso será visto mais adiante. Digite "Olá, Mundo!" e clique no botão OK.

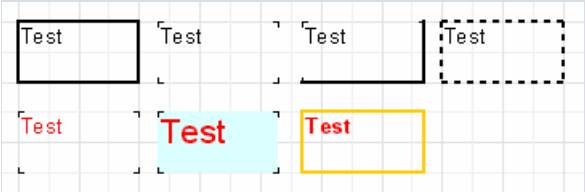


O design do relatório está concluído. Para visualizar o relatório, selecione o item de menu "Arquivo > Visualizar" ou clique no botão correspondente na barra de ferramentas. A janela de visualização que contém a página do relatório com o texto "Olá, Mundo!" é exibida. Este relatório pode ser impresso, salvo em um arquivo (*.fp3) ou exportado para um dos formatos de exportação suportados.

O objeto "Texto"

O objeto "Texto" possui diversos recursos. Ele pode exibir texto em um quadro e ser preechido com uma cor. O texto pode ser exibido usando qualquer fonte de qualquer tamanho e estilo. Todas as propriedades podem ser definidas visualmente com a ajuda das barras de ferramentas.

Seguem alguns exemplos do design do texto:

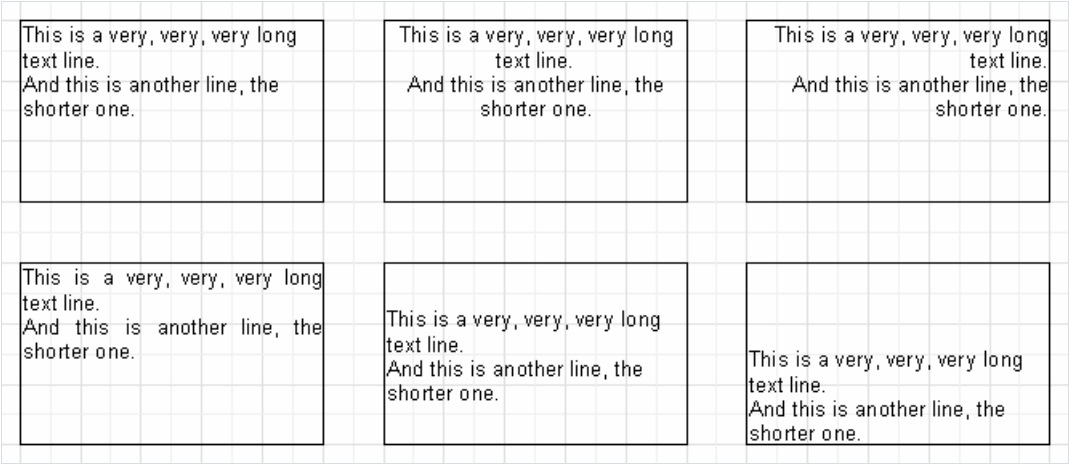


Agora vamos dar uma olhada em outros recursos deste objeto básico. Como exemplo, criaremos um novo objeto "Texto" que exibirá duas linhas de texto:

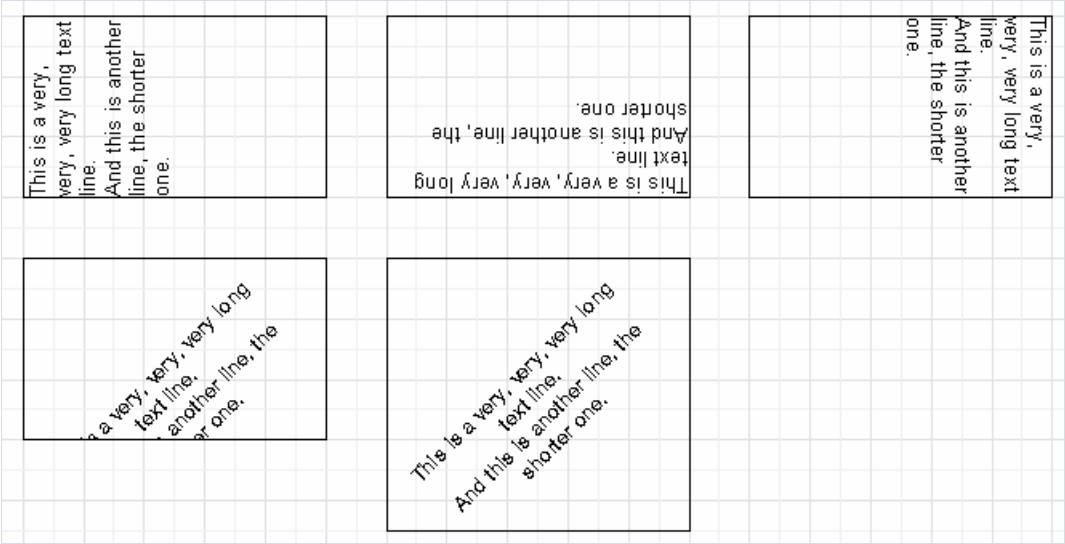
```
Esta é uma linha de texto muito, muito, muito longa.
E esta é outra linha, a mais curta.
```

Habilite a borda do objeto na barra de ferramentas e redimensione o objeto para 9x3 cm usando o mouse. Vemos que o objeto pode não apenas exibir uma única linha, mas diversas linhas de texto. Agora reduza a largura do objeto para 5 cm. É óbvio que linhas longas não couberam no objeto e foram quebradas. Isso é controlado através da propriedade `WordWrap` do objeto. Se ela for desabilitada (no inspetor de objetos ou através do menu de contexto do objeto), linhas longas serão apenas cortadas.

Agora veremos como funciona o alinhamento de texto dentro do objeto. Os botões de alinhamento estão localizados na barra de ferramentas "Texto" e eles controlam o alinhamento de texto horizontal ou vertical. Observe o botão "Justificar", que permite alinhar parágrafos a ambas as bordas do objeto. Para fazer isso, `WordWrap` deve estar habilitada.



Todo o texto no objeto pode ser girado em qualquer ângulo no intervalo 0...360°. O botão na barra de ferramentas "Texto" permite girar o texto rapidamente para as predefinições 0, 45, 90, 180 ou 270°. Para qualquer outro valor, configure o ângulo desejado na propriedade `Rotation` no inspetor de objetos. Ao girar texto em ângulos diferentes de 90, 180 ou 270°, o texto pode ser cortado pelo quadro do objeto, conforme é mostrado abaixo. Para resolver isso, aumente levemente a altura do objeto para que todo o texto caiba dentro do objeto.



Vamos dar uma olhada rápida em outras propriedades do objeto "Texto" que influenciam sua aparência. A maior parte destas propriedades estão disponíveis apenas no inspetor de objetos:

- `BrushStyle` : o tipo de preenchimento do objeto
- `CharSpacing` : o espaçamento entre caracteres, em pixels
- `GapX` , `GapY` : os recuos do texto na borda esquerda e superior, em pixels
- `LineSpacing` : o espaçamento entre linhas, em pixels
- `ParagraphGap` : o recuo da primeira linha do parágrafo, em pixels

Tags HTML no objeto "Texto"

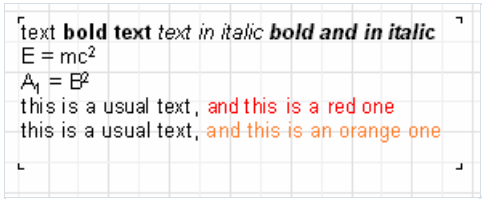
O objeto "Texto" entende algumas tags simples de HTML. As tags podem estar localizadas dentro do texto do objeto. Tags são desabilitadas por padrão; para habilitá-las selecione "Permitir Tags HTML" no menu de contexto do objeto ou habilite a propriedade `AllowHTMLTags` no inspetor de objetos. Aqui está a lista de tags suportadas.

Tag	Meaning
`<b>`	texto em negrito
`<i>`	texto em itálico
`<u>`	texto sublinhado
`<sub>`	subscrito
`<sup>`	sobrescrito
`<font color>`	cor da fonte
`<nowrap>`	texto que não é quebrado quando `WordWrap` for habilitado, todo o texto é movido para a próxima linha

Observe que somente algumas tags são suportadas, mas isso deve ser suficiente para a maior parte das aplicações. Não é possível alterar o tamanho ou nome da fonte através de tags HTML.

Os exemplos a seguir mostram como as tags podem ser usadas.

```
text <b>bold text</b> <i>text in italic</i> <b><i>bold and in italic</b></i>
E = mc<sup>2</sup>
A<sub>1</sub> = B<sup>2</sup>
this is a usual text, <font color=red>and this is a red one</font>
this is a usual text, <font color="#FF8030">and this is an orange one</font>
```



Exibir expressões com a ajuda do objeto "Texto"

Um dos recursos mais importantes deste objeto básico é sua habilidade de exibir não apenas texto estático, mas também expressões. Expressões podem ser colocadas dentro do objetos, misturadas a texto normal. Vamos ver um exemplo simples de como isso funciona. Digite o seguinte no objeto:

```
Olá, Mundo! Hoje é [DATE].
```

Quando o relatório for executado, obteremos algo parecido a isto:

```
Olá, Mundo! Hoje é 01.01.2004.
```

Como isso acontece? Quando FastReport cria o relatório e encontra uma expressão entre colchetes, o mecanismo de relatórios calcula o valor da expressão e insere este valor no texto no lugar da expressão. Objetos "Texto" podem conter inúmeras expressões misturadas ao texto normal. Expressões complexas podem conter parênteses (por exemplo [1+2*(3+4)]). Constantes, variáveis, funções e campos de BD podem ser usados em expressões. Iremos aprender mais sobre eles mais tarde neste capítulo.

O FastReport reconhece expressões entre colchetes automaticamente no texto. Mas o que acontece se nosso texto normal conter colchetes que não queremos que sejam considerados como expressões? Por exemplo, se precisamos exibir o seguinte:

```
a[1] := 10
```

FastReport consideraria [1] como uma expressão e exibiria o texto como:

```
a1 := 10
```

o que não desejamos que aconteça, é claro. Uma maneira de evitar isso é desabilitar o reconhecimento de expressões. Desabilite a propriedade `AllowExpressions` (ou "Permitir Expressões" no menu de contexto) e todas as expressões no texto serão ignoradas. Em nosso exemplo, o FastReport iria exibir exatamente o que precisamos:

```
a[1] := 10
```

Porém, às vezes, é necessário que um texto tenha tanto uma expressão quanto texto normal entre colchetes, por exemplo:

```
a[1] := [myVar]
```

Desabilitar `AllowExpressions` nos permite exibir colchetes nos lugares necessários, mas também desabilita a manipulação de expressões. Nesta situação, o FastReport permite a utilização de um conjunto de símbolos alternativo para designar expressões. Por padrão, a propriedade `ExpressionDelimiters` é responsável por isso. Em nosso exemplo podemos usar chevrons para as expressões, em vez de colchetes:

```
a[1] := <myVar>
```

O valor "<,>" deve ser definido na propriedade `ExpressionDelimiters`, onde a vírgula é necessária para separar os símbolos de abertura e fechamento. Outro requisito é que os símbolos de abertura e fechamento não podem ser idênticos, então "%,%" não funcionará. Podem ser usados símbolos complexos, por exemplo "<%,%>". Então nosso exemplo poderia ter o seguinte aspecto:

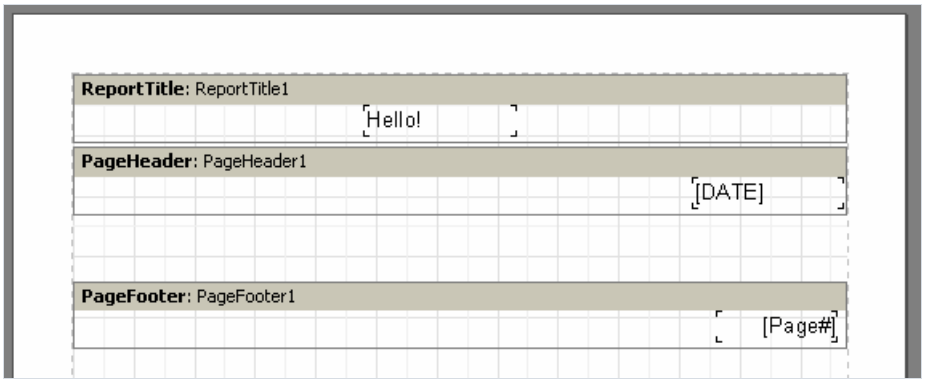
```
a[1] := <%myVar%>
```

Bandas no FastReport

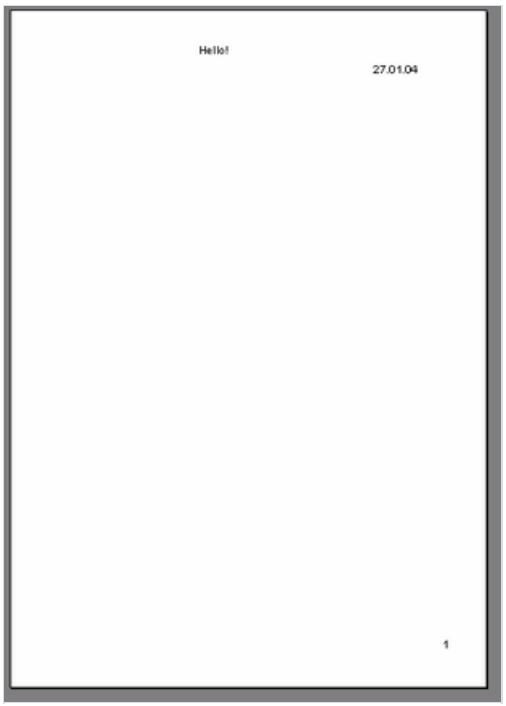
Bandas são usadas para colocar os objetos que elas contêm em locais específicos na página de saída. Ao colocar um objeto na banda "Cabeçalho da Página" nós dizemos ao mecanismo de relatórios que o objeto deve ser exibido no topo de cada página no relatório final. De maneira similar, objetos na banda "Rodapé da Página" são exibidos na parte inferior de cada página. Vamos demonstrar isso com um exemplo. Criaremos um relatório contendo "Oi!" no topo da página, a data atual à direita e o número da página no rodapé da página ao lado direito.

Abra o designer do FastReport e clique no botão "Novo Relatório" na barra de ferramentas. É exibido um modelo de relatório que já contém três bandas: "TítuloDoRelatório", "DadosMestre" e "RodapéDaPágina". Vamos remover a banda "DadosMestre" brevemente (clique em qualquer espaço livre dentro da banda ou em seu cabeçalho e remova-a com a tecla "Delete" ou o item "Remover" no menu de contexto). Agora vamos adicionar uma nova banda ("Cabeçalho da Página"). Clique no botão "Inserir Banda" na barra de ferramentas de objetos e selecione "Cabeçalho da Página" na lista suspensa. Vemos que uma banda nova é adicionada à página. Ao mesmo tempo, as bandas existentes são movidas para baixo. O designer do FastReport posiciona as bandas na página automaticamente – bandas de cabeçalho no topo, bandas de dados no meio e bandas de rodapé embaixo.

Agora vamos adicionar alguns objetos. Adicione um objeto "Texto do Sistema" à banda "CabeçalhoDaPágina" e em seu editor selecione "[DATE]" na lista suspensa de variáveis do sistema (lembre-se que a data atual também pode ser exibida em um objeto "Texto" ao digitar "[DATE]" em seu editor). Depois adicione um objeto "Texto" contendo "Oi!" à banda "TítuloDoRelatório". Um objeto "Texto" que exibe o número da página já foi adicionado automaticamente à banda "RodapéDaPágina".



Ao executar o relatório, você verá que os objetos no relatório pronto são impressos nas posições corretas na página.



Então bandas são responsáveis por posicionar objetos na página. Dependendo do tipo de banda, é possível imprimir objetos no topo ou na parte inferior de uma página, e na primeira ou última página. As bandas básicas necessárias na maioria dos relatórios se comportam deste modo:

- Banda "Cabeçalho da Página": exibida no topo de cada página
- Banda "Rodapé da Página": exibida na parte inferior de cada página
- Banda "Título do Relatório": exibida no topo da primeira página. Ela pode estar antes ou depois da banda "Cabeçalho da Página", dependendo da propriedade "TitleBeforeHeader" da página (encontrada no inspetor de objetos após clicar em qualquer espaço livre na página)
- Banda "Sumário do Relatório": exibida no espaço livre no final do relatório

Bandas de dados

Agora iremos aprender como imprimir dados de tabelas de BD ou de consultas. O que é considerado uma 'tabela' ou uma 'consulta'? Elas são compostas de dados organizados em linhas (registros ou fileiras) que contém uma ou mais colunas (campos). Para imprimir estes tipos de dados, o FastReport utiliza um tipo de banda especial, as diversas bandas que são chamadas "Dados...". Para imprimir uma tabela inteira ou apenas algumas fileiras e campos, é necessário adicionar uma ou mais dessas bandas ao relatório, vinculá-las à tabela e colocar objetos de campo dentro delas. Quando FastReport criar o relatório, as bandas serão impressas na página de saída, uma vez para cada registro na tabela. Se não sobrar espaço livre na página de sáida para uma banda, o mecanismo de relatórios criará uma nova página de saída antes de continuar.

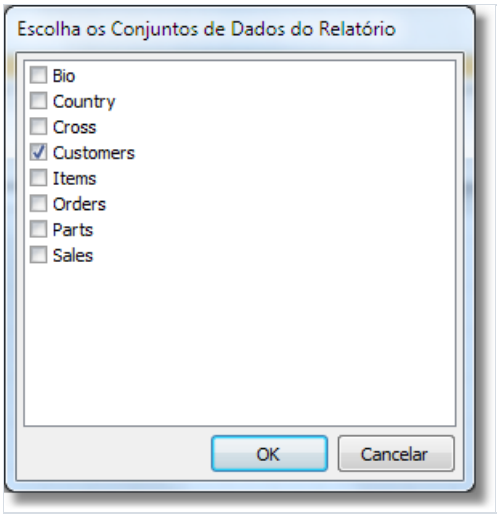
O componente TfrxDBDataSet

O componente conector `TfrxDBDataSet`  da paleta de componentes Delphi do FastReport é usado para conectar uma tabela (ou qualquer outra fonte de dados) ao FastReport. Este componente atua como o mensageiro entre a fonte de dados e o núcleo do FastReport. O componente é responsável pela navegação de registros e pela referência da tabela. Isso torna o núcleo do FastReport independente de bibliotecas de acesso a dados. O FastReport pode utilizar as bibliotecas "BDE", "IB_Objects" (que possuem uma implementação não padrão que é incompatível com TDataSet) e outras, assim como dados de fontes não conectadas a um banco de dados, como matrizes e arquivos.

- O componente `TfrxDBDataSet` foi projetado para ser usado com fontes de dados que sejam compatíveis com `TDataSet` (como BDE, ADO, IBX e a maioria das outras bibliotecas).
- O componente `TfrxDBDataSet` é destinado ao trabalho com IB_Objects.
- O componente `TfrxUserDataSet`  funciona com outras fontes de dados (matrizes, arquivos, etc.).

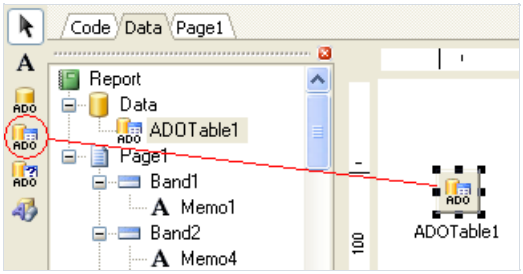
É muito fácil usar o componente `TfrxDBDataSet`. Conecte-o à fonte de dados ao configurar a propriedade `DataSet` (para a conexão direta a uma tabela ou consulta) ou a propriedade `DataSource` (para a conexão através do componente `TDataSource`). Os dois métodos são equivalentes e mutuamente exclusivos, apesar do primeiro permitir o gerenciamento dos dados sem precisar de um componente `TDataSource`.

Para deixar o componente Delphi (e os dados conectados a ele) disponível para o relatório, qualquer conjunto de dados usado no relatório deve ser habilitado. Faça isso no item de menu "Relatório > Dados..." no designer do FastReport e selecione os conjuntos de dados necessários na caixa de diálogo que é aberta.



Relatório "Lista de Clientes"

Nosso segundo relatório será muito mais complexo que o primeiro (ele irá conter uma tabela de BD de clientes da empresa). Vamos usar o banco de dados de demonstração demo.mdb que é incluída no kit de distribuição do FastReport. Crie um novo relatório. Vá para a guia "Dados", coloque o componente "Tabela ADO" na página:



e configure uma propriedade:

```
TableName = 'Customer'
```

Agora a tabela e todos os seus campos se tornarão visíveis no painel "Árvore de Dados" no IDE.

Agora vamos fazer o design do relatório. Primeiro adicione um objeto "Texto" que contém "Lista de clientes" à banda "TítuloDoRelatório".

Depois conecte a banda "DadosMestre" a nosso conjunto de dados. Isso pode ser feito de três maneiras:

- clique duas vezes na banda
- selecione "Editar..." no menu de contexto da banda
- ou clique na propriedade "DataSet" da banda no inspetor de objetos

Agora coloque quatro objetos "Texto" (que irão exibir os campos número do cliente, nome do cliente, telefone e fax do conjunto de dados) na banda. Faremos isso de maneiras diferentes para demonstrar alguns recursos do designer do FastReport.

A primeira maneira é colocar um objeto "Texto" na banda e digitar "[ADOTable1."CustNo"]" nele. Esta é a maneira menos conveniente, já que o link do campo deve ser inserido manualmente, havendo a possibilidade de digitar o texto de maneira incorreta. É um pouco mais fácil usar o designer de expressões – clique duas vezes no objeto "Texto" e clique no botão mais à esquerda da guia "Texto" da janela do editor que é aberto. Para inserir nosso campo, clique duas vezes no nome do campo no designer de expressões. Clique no botão OK para fechar a caixa de diálogo e ver o link do campo inserido no objeto "Texto".

A segunda maneira de adicionar o campo de BD ao objeto "Texto" é ao configurar duas propriedades no inspetor de objetos. Coloque um segundo objeto "Texto" na banda, sem escrever nada na janela do editor. Configura as propriedades do objeto usando o inspetor de objetos:

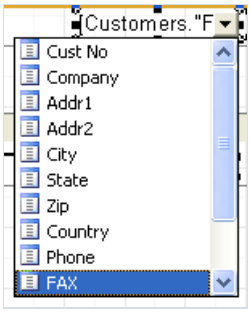
```
DataSet = ADOTable1
DataField = 'Company'
```

Ambas as propriedades são apresentadas como uma lista, então apenas selecione o valor necessário da lista suspensa usando o mouse.

A terceira maneira é "arrastar e soltar" o campo necessário do painel "Árvore de Dados" para dentro do relatório.

Esta é a maneira mais simples e mais fácil, mas antes de fazer isso, desmarque a caixa de seleção "Criar rótulo" na parte inferior do painel de "Árvore de Dados", ou um segundo objeto "Texto", que contém o nome do campo, é criado junto ao objeto link do campo. Desta maneira, selecione o campo "Phone" e arraste-o para a banda.

Para a quarta maneira é necessário ativar o sinalizador da opção do designer "Mostrar lista suspensa de campos" (através do menu principal: "Exibir > Opções"), e também é necessário que a banda já esteja conectada ao conjunto de dados. Coloque um objeto "Texto" em branco na banda e coloque o cursos do mouse sobre o objeto – um botão da lista suspensa aparecerá na extremidade direita. Clique neste botão para abrir a lista de campos de BD e selecione o campo "FAX".



O design de nosso relatório está concluído.



Clique no botão "Visualizar" para ver o resultado.



Customer list

1221	Kauai Dive Shoppe	808-555-0269	808-555-0278
1231	Unisco	809-555-3915	809-555-4958
1351	Sight Diver	357-6-876708	357-6-870943
1354	Cayman Divers World Unlimited	011-5-697044	011-5-697064
1356	Tom Sawyer Diving Centre	504-798-3022	504-798-7772
1380	Blue Jack Aqua Center	401-609-7623	401-609-9403
1384	VIP Divers Club	809-453-5976	809-453-5932
1510	Ocean Paradise	808-555-8231	808-555-8450
1513	Fantastique Aquatica	057-1-773434	057-1-773421
1551	Marmot Divers Club	416-698-0399	426-698-0399
1560	The Depth Charge	800-555-3798	800-555-0353
1563	Blue Sports	610-772-6704	610-772-6898
1624	Makai SCUBA Club	317-649-9098	317-649-6787
1645	Action Club	813-870-0239	813-870-0282
1651	Jamaica SCUBA Centre	011-3-697043	011-3-697043
1680	Island Finders	713-423-5675	713-423-5676

Exibir campos de BD com a ajuda do objeto "Texto"

Como foi visto, objetos "Texto" podem exibir dados de um BD, assim como texto estático e expressões. Também vimos que isso pode ser feito de duas maneiras: ao colocar um link ao campo do BD no texto do objeto, ou ao conectar o objeto ao campo necessário através de suas propriedades `DataSet` e `DataField`. A primeira maneira é apropriada para exibir tanto o conteúdo do campo e o texto de qualificação no mesmo objeto. Por exemplo:

```
Pessoa de contato: [frxDBDataSet1."Contact_Person"]
```

É necessário usar uma sintaxe especial para links ao campo do BD: [nomeconjuntodedados."nomecampo"]. O nome do campo (assim como o nome do conjunto de dados) pode conter espaços, mas não deve haver nenhum espaço entre os símbolos "ponto" e "aspas".

Além disso, também podemos aplicar operações de computação a um campo na expressão, como é mostrado aqui:

```
Comprimento (cm): [<frxDBDataSet1."Length_in"> * 2.54]
```

Observe como devem ser usados colchetes e chevrons. Lembre-se de que colchetes são usados por padrão para delimitar expressões no texto do objeto. Quando necessário, os colchetes podem ser substituídos por qualquer outro par de caracteres de abertura/fechamento (consulte a seção "Exibir expressões com a ajuda do objeto 'Texto'"). Chevrons são usados dentro de expressões para delinear as variáveis do FastReport ou campos de BD. De maneira lógica, deveríamos escrever:

```
Pessoa de contato: [<frxDBDataSet1."Contact_Person">]
```

em vez de

```
Pessoa de contato: [frxDBDataSet1."Contact_Person"]
```

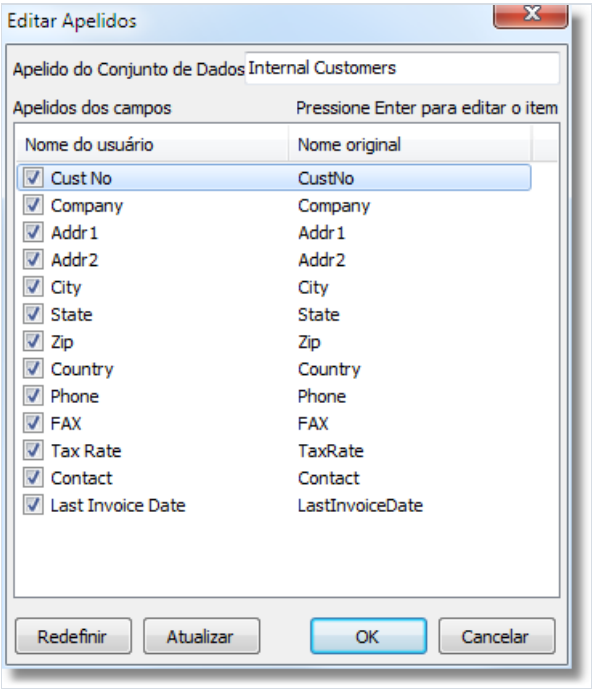
No entanto, ambas as notações estão corretas, já que o FastReport não requer colchetes quando a expressão contém somente uma variável ou somente um campo de BD. Em todos os outros casos, os chevrons são necessários, como em:

```
Comprimento (cm): [<frxDBDataSet1."Length_in"> * 2.54]
```

Aliases

No relatório anterior, a fonte de dados se chamava ADOTable1, os campos eram chamados "CustNo", "Company", "Phone" e "FAX", e nós nos referimos a eles no relatório usando algo parecido com "[ADOTable1."CustNo""]". Isso é fácil de entender? Não exatamente. Seria mais claro se a fonte de dados e o campo fossem chamados "Nossos clientes" e "Número", respectivamente. No entanto, há uma maneira de evitar isso. Podemos usar pseudônimos ou aliases para esses itens. No FastReport, tanto o conjunto de dados como seus campos possuem uma segunda propriedade de nome conhecida como `UserName`, ou seja, um alias, que pode ser alterada com facilidade. Sempre que for dado um alias a um componente (ou seja, `UserName` foi alterado do seu nome padrão), este alias deve ser usado no FastReport – a propriedade Name do componente não será reconhecida.

É muito fácil configurar aliases para uma fonte de dados e seus campos no FastReport. Para abrir o editor de alias, clique duas vezes no componente ADOTable1 ou use seu menu de contexto. É possível modificar o conjunto de dados e os aliases dos campos no editor, e também especificar quais campos são necessários no relatório. Vamos renomear o conjunto de dados e os campos conforme é visto abaixo:



O alias do conjunto de dados também pode ser modificado sem usar o editor de alias, ao alterar a propriedade `UserName` do componente ADOTable1.

Tendo feito isso, agora precisamos modificar o relatório, já que os nomes do relatório e os campos foram alterados. Para modificar os nomes dos campos em objetos do relatório, é mais fácil usar o quarto método descrito na seção "Relatório 'Lista de Clientes'". Mova o cursor sobre o objeto "Texto", de modo que o botão apareça no lado direito do objeto, clique no botão e selecione o campo da lista. Após fazer isso a origem dos dados no conjunto de dados e seus campos se torna mais aparente.

Observação: É melhor atribuir aliases no início do design do relatório, a fim de evitar renomear todos os campos em um relatório mais tarde.

Variáveis

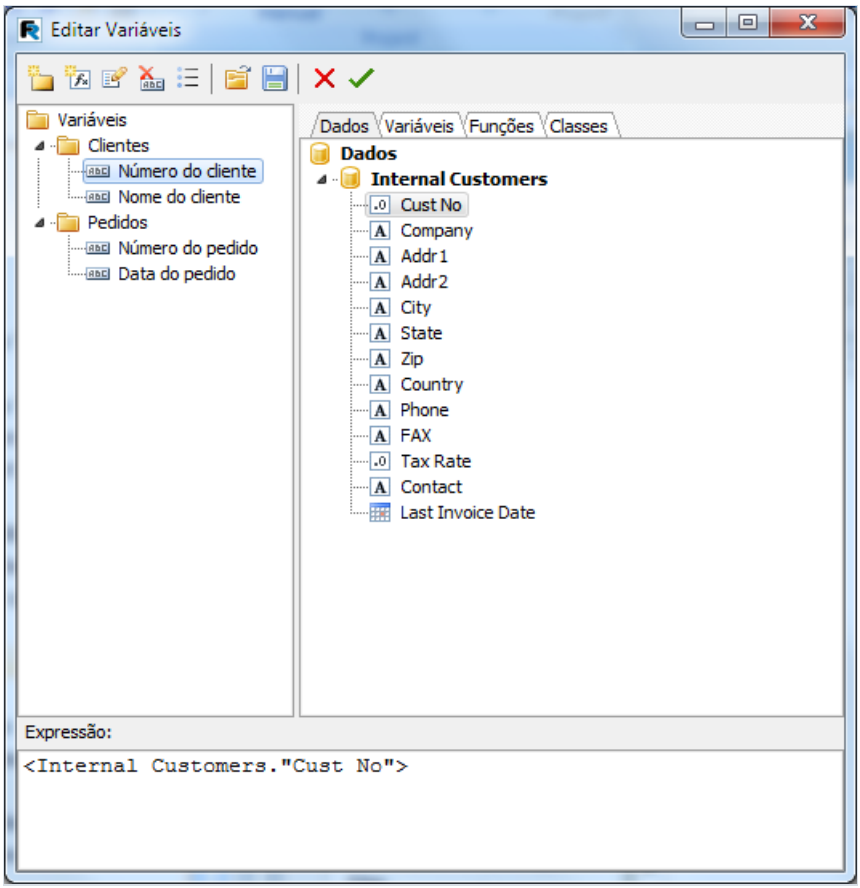
Assim como aliases, há outra maneira de ajudar o designer do relatório a definir nomes mais inteligíveis para campos de BD e outros elementos de informação. Um nome de campo de BD ou qualquer expressão podem ser associados a uma variável. Para criar e trabalhar com variáveis no FastReport, selecione o item de menu "Relatório > Variáveis...".

A lista de variáveis no FastReport possui uma estrutura de dois níveis. O primeiro nível contém a categoria e o segundo nível contém a variável em si. Esta categorização de variáveis é conveniente quando a lista de variáveis for muito longa. Uma lista de variáveis deve conter pelo menos uma categoria, o que significa que variáveis não podem estar localizadas no nível superior. Além disso, nomes de categorias não podem ser incluídos em relatórios, então cada nome na lista deve ser único.

Vamos ilustrar o uso de variáveis com o seguinte exemplo. Assumiremos que temos duas fontes de dados: a primeira é "frxDBDataSet1", com os campos "CustNo" e "Name", e a segunda é "frxDBDataSet2", com os campos "OrderNo" e "Date". Podemos associar a seguinte lista de variáveis aos campos:

```
Clientes
  Número do cliente
  Nome do cliente
Pedidos
  Número do pedido
  Data do pedido
```

onde "Clientes" e "Pedidos" são duas categorias. Abra o editor de variáveis e crie esta estrutura usando os botões "Nova Categoria", "Nova Variável" e "Editar". Para vincular as variáveis aos campos de BD, selecione uma variável e clique duas vezes no campo de BD necessário no painel direito. O link ao campo do BD será exibido no painel inferior. A variável agora está associada a esta expressão, então o valor da variável se torna o valor da expressão. Se for necessário, a expressão pode ser editada ou modificada manualmente, e qualquer função do FR ou outras variáveis podem ser usadas nela. Lembre-se que categorias não devem ser associadas a expressões.



Após criar a lista de variáveis, feche o editor de variáveis. Agora podemos inserir as variáveis no relatório. Ao contrário de campos de BD, há um número menor de métodos de inserção. Podemos inserir uma variável no texto do objeto manualmente ao digitar "[Número do cliente]", ou podemos arrastar uma variável do painel "Árvore de Dados" para a página do relatório. Neste caso é necessário abrir a guia "Variáveis" neste painel.

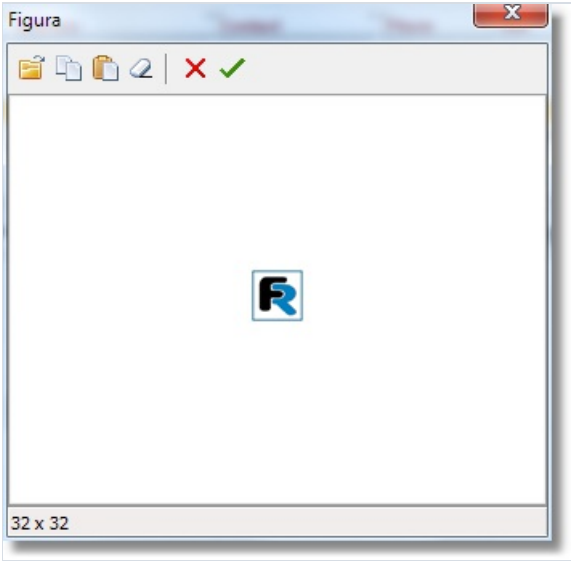
Vamos revisar o que aprendemos até agora:

- O design de um relatório é composto de páginas de design
- As páginas podem conter objetos FR, que podem ser colocados na página ou dentro de uma banda
- Bandas são espaços reservados na página de design e controlam onde os objetos que elás contêm são exibidos na página de saída, de acordo com o tipo de banda
- Objetos "Texto" contêm o texto que desejamos exibir em uma certa posição. Eles podem ter várias linhas e podem conter texto estático, campos de dados, variáveis, expressões ou uma combinação destes
- Bandas de tipos de dados (Mestre, Detalhe, Sub-detalhe, etc.), quando conectadas a um `TfrxDbDataset`, controlam o número de vezes que estas bandas são exibidas (fileiras) e, em conjunto com o mecanismo de relatórios, determinam o número de páginas concluídas geradas pelo relatório. Observação: apesar destas bandas de dados terem tipos como Mestre, Detalhe, etc., isto somente designa a relação de locais na hierarquia de posicionamento das bandas nas páginas de saída. As relações de dados dependem das relações de tabela/consulta nos frxdbdatasets conectados. Cada nível de banda de dados requer um `TfrxDbDataset` separado ou uma associação equivalente.

Objeto "Figura"

O próximo objeto a ser abordado é o objeto "Figura". Ele também é usado nos relatórios com frequência. Ao usar este objeto é possível inserir logos, fotos (de um funcionário, por exemplo) ou qualquer outras informações pictóricas. O objeto pode exibir gráficos nos formatos `BMP`, `ICO`, `WMF` ou `EMF`.

Vamos dar uma olhada nas funcionalidades deste objeto. Crie um relatório em branco e coloque um objeto "Figura" na página do relatório. Se o editor de figura não abrir automaticamente, clique duas vezes no objeto "Figura". Você pode carregar qualquer figura de um arquivo gráfico ou apagar a figura existente e fechar o editor ao clicar na marca de seleção verde.



Algumas propriedades do objeto estão disponíveis no menu de contexto, assim como no inspetor de objetos:

- Tamanho Automático (`AutoSize`)
- Esticar: ativado por padrão (`Stretch`)
- Centro (`Center`)
- KeepAspectRatio: ativado por padrão (`KeepAspectRatio`)

Quando "Tamanho Automático" estiver ativado, o objeto é redimensionado para o tamanho da figura que contém. Às vezes isso pode ser útil se são exibidas figuras de bancos de dados de diversos tamanhos. "Tamanho Automático" é desativado por padrão.

"Esticar" é ativado por padrão. Esta propriedade estende a figura dentro do objeto. Altere o tamanho do objeto usando o mouse e você verá que toda a figura é sempre exibida dentro do quadro do objeto. Se "Esticar" for desativado, a figura (ou somente parte dela) é exibida em seu tamanho original. Este comportamento difere de "Tamanho Automático" porque você tem controle sobre o tamanho do quadro do objeto, que pode ser maior o menor que a figura que contém. Com "Tamanho Automático" ativado, você não possui nenhum controle sobre o tamanho do quadro do objeto.

A propriedade "Centro" alinha a figura dentro do objeto.

"KeepAspectRatio" é ativado por padrão. Ela evita que a figura seja distorcida quando o tamanho do objeto for modificado. Esta propriedade somente é efetiva quando "Esticar" também estiver ativado. Quando "KeepAspectRatio" estiver ativado, um círculo desenhado permanece sendo um círculo sem se tornar um oval, não importando o tamanho do objeto "Figura". A figura estendida ocupa somente a parte do espaço interno do objeto

que é necessária para exibí-la com a taxa de proporção correta. Quando é desativado, a figura é estendida para preencher o quadro do objeto e será distorcida se a taxa de proporção do quadro não for a mesma da figura.

Outra propriedade útil que está disponível somente no inspetor de objetos é `FileLink`. Ao inserir o nome de um arquivo, como c:\figura.bmp, ou uma variável que contém o nome de um arquivo, como [figura_arquivo], irá carregar a figura do arquivo nomeado ao executar o relatório.

Relatório com figuras

O objeto "Figura", como muitos objetos no FastReport, pode exibir dados de um BD. Este objeto pode ser conectado a um campo de BD ao configurar as propriedades `DataSet` e `DataField` no inspetor de objetos. Ao contrário do objeto "Texto", esta é a única maneira de conectar este objeto a seus dados.

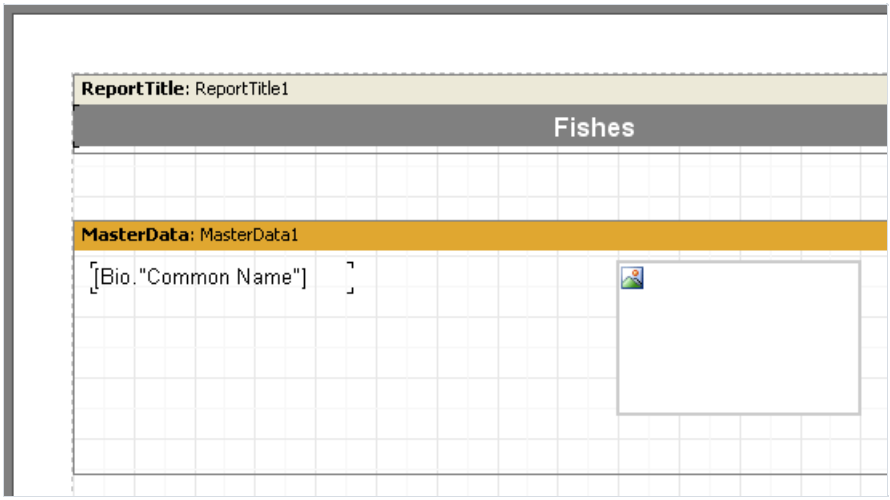
Vamos demonstrar isso com um relatório que exibe os nomes e as imagens de alguns peixes. Crie um novo relatório, coloque o componente "ADO Table" no relatório e configure estas propriedades:

```
TableName = 'Biolife'
UserName = 'Bio'
```

Agora iremos adicionar objetos à página do relatório. Coloque um objeto "Texto" que contém "Peixe" na banda "TítuloDoRelatório". Conecte a banda "DadosMestre" à fonte de dados (clique duas vezes na banda e selecione "Bio" da lista). Aumente a altura da banda para 5 cm ao arrastar a parte inferior da banda para baixo ou ao usar o inspetor de objetos. Coloque um objeto "Texto" na banda e conecte-o ao campo "CommonName", usando qualquer um dos métodos já descritos. Depois disso, coloque um objeto "Figura" ao lado e conecte-o ao campo "Gráfico" ao configurar suas propriedades no inspetor de objetos.

```
DataSet = Bio
DataField = 'Graphic'
```

Observe que ambas as propriedades são do tipo "Lista" e podem ser configuradas com os valores necessários usando o mouse. Para criar espaço para a figura, estenda o objeto para 4 x 2,5 cm.



O design do relatório agora está concluído, e ele produz o relatório mostrado abaixo:

Fishes

Clown Triggerfish



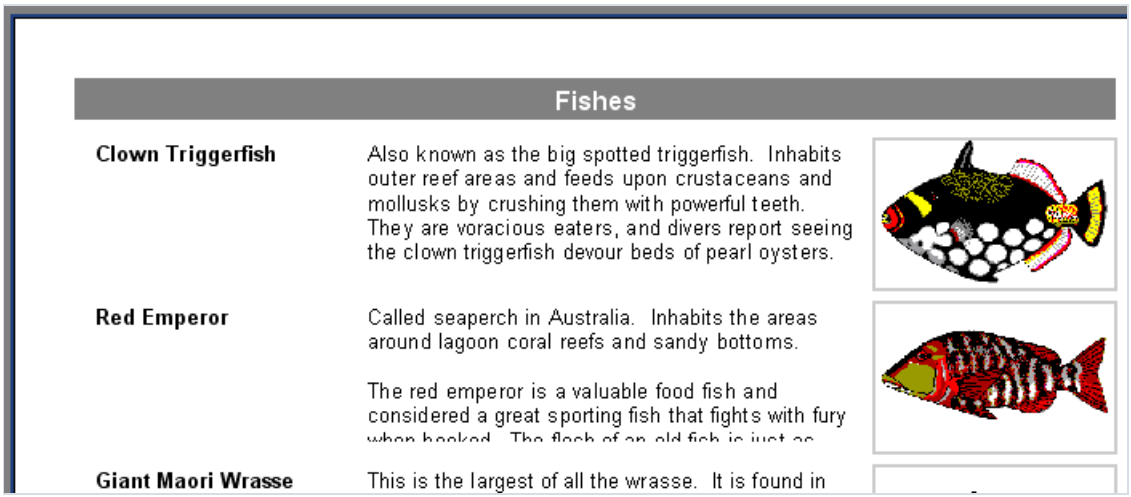
Red Emperor



Exibir texto com várias linhas

Vamos melhorar o exemplo anterior. A tabela "Biolife" possui um campo "Notes" que contém uma descrição detalhada de cada peixe. Atualize nosso relatório ao adicionar este campo a ele.

À primeira vista, isso parece fácil – adicionar um objeto "Texto" à banda de dados entre os objetos existentes, conectá-lo ao campo "Notes" e configurar o tamanho do objeto como 8 x 2,5 cm. No entanto, a visualização do relatório não mostra exatamente o que desejamos:



FastReport executou exatamente o que foi instruído a fazer. O campo "Notas" contém um texto com várias linhas de comprimentos variados, porém o objeto "Texto" que exibe as informações deste campo tem um tamanho fixo. É por isso que algumas linhas parecem cortadas. O que pode ser feito a respeito?

É claro que o tamanho do objeto pode ser aumentado ou o tamanho da fonte pode ser reduzido. No entanto, isso pode levar à perda de espaço na página de saída, já que alguns peixes possuem descrições longas, e outros possuem descrições curtas. O FastReport tem algumas propriedades que nos permitem solucionar este problema.

Essas propriedades permitem que uma banda ou um objeto ajustem sua altura automaticamente para criar o espaço necessário para um registro (fileira). Para obter isso, precisamos apenas ativar a propriedade "Esticar", tanto da banda como do objeto "Texto". No entanto, isso não é tudo, já que um objeto "Texto" com texto mais longo deve ser capaz de se estender automaticamente. Para isso precisamos configurar outras propriedades também.

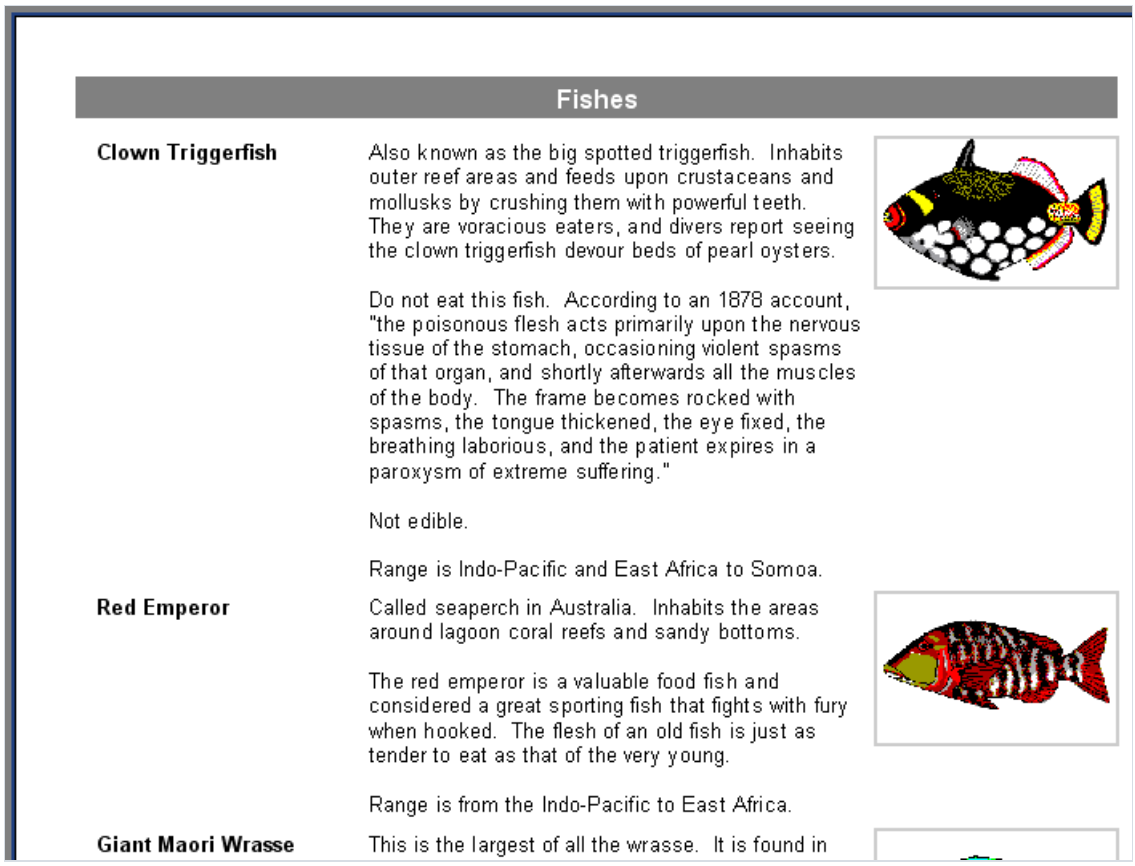
O objeto "Texto" é capaz de definir sua altura e largura automaticamente para criar espaço para seu conteúdo. "AutoWidth" e "EsticarMode" podem ser usado para isso. "AutoWidth" permite que o objeto texto varie sua largura, para que todas as linhas criem espaço sem dividir nenhuma palavra. Este modo é útil quando um objeto possui uma única linha de texto, e quando o aumento de tamanho à direita não afeta outros objetos. A propriedade "Esticar" permite que a altura de um objeto aumente para acomodar o texto, sem alterar a largura do objeto. "Esticar" tem diversos modos que podem ser selecionados no inspetor de objetos:

smDontEsticar – não estender o objeto (padrão)

smActualHeight – estender o objeto para criar espaço para todo o texto

smMaxHeight – estender o objeto para que sua parte inferior alcance a parte inferior da banda no qual estiver colocado. Abordaremos este modo mais adiante

Aqui estamos interessados na propriedade "Esticar" do objeto "Texto". Habilite-a usando o menu de contexto do objeto ou ao configurar "EsticarMode" como smActualHeight. Também habilite a propriedade "Esticar" da banda. Visualize o relatório e certifique-se de que tudo é exibido como esperado.

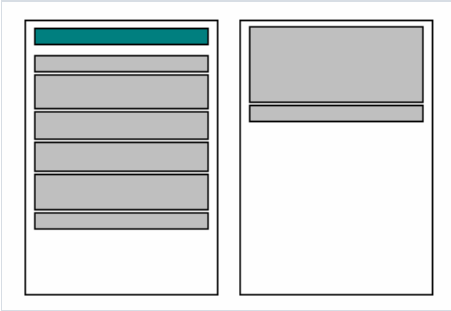


Como você pode ver, ao criar o relatório, o FastReport preenche os objetos com dados e os estende se "Esticar" for habilitado. Ele então calcula a altura da banda para que cada objeto tenha espaço suficiente. Se a propriedade "Esticar" da banda for desabilitada, este ajuste de altura não é executado e a banda é exibida na altura definida no design. Neste caso, veríamos que objetos com texto mais longo seriam estendidos, mas a banda não o seria, o que levaria à sobreposição do texto, já que a banda seguinte seria exibida imediatamente após a anterior.

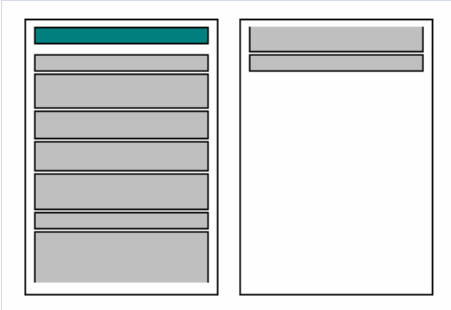
Divisão de dados

Vamos dar uma olhada em um aspecto peculiar deste relatório: há muito espaço em branco na parte inferior das páginas. Por que? Quando um relatório é criado, o mecanismo do FastReport preenche o espaço em branco da página com as bandas. Após exibir cada banda, a posição atual é movida para baixo. Quando FastReport percebe que ná há espaço livre suficiente para exibir a próxima banda (sua altura é maior que o espaço livre da página), ele cria uma nova página e continua com a exibição da banda neste ponto. Esta sequência é repetida para cada registro no conjunto de dados.

Nosso relatório contém um objeto com texto longo, e é por isso que a altura da banda é bastante grande. Além disso, se uma banda grande não possuir espaço suficiente em uma página, ela é deslocada para a próxima, deixando muito espaço em branco na parte inferior da página, como é mostrado aqui:



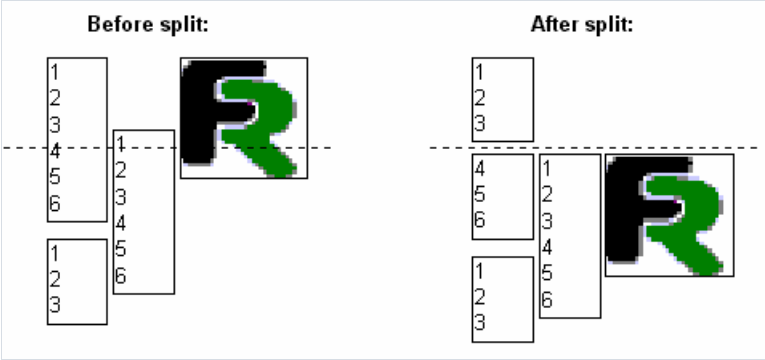
Para limitar o desperdício de papel, vamos usar um recurso do FastReport que transforma o conteúdo de uma banda em parágrafos. Tudo que precisamos fazer é habilitar a propriedade "AlllowSplit" da banda "Dados de 1o. Nível". Você verá que agora há menos espaço vazio na parte inferior das páginas do relatório:



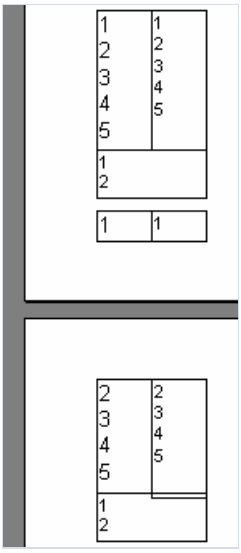
Como funciona essa divisão de bandas? Há alguns objetos no FastReport que suportam este recurso. Eles são os objetos "Texto", "Linha" e "RichEdit". Eles podem ser "divididos", e outros objetos não podem. Quando FastReport precisa dividir uma banda, ele o faz da seguinte maneira:

- exibe os objetos não divisíveis que cabem no espaço vazio
- exibe os objetos divisíveis parcialmente (objetos "Texto" são exibidos de uma forma que todas as linhas cabem no objeto)
- cria uma nova página e continua com a exibição do objeto na banda
- se um objeto não divisível não possuir espaço no espaço vazio, ele é deslocado para a próxima página; ao mesmo tempo todos os objetos localizados abaixo dele são deslocados conforme necessário
- o processo continua até que todos os objetos da banda seja exibidos por completo.

O algoritmo de divisão se tornará mais claro ao olhar para este exemplo:



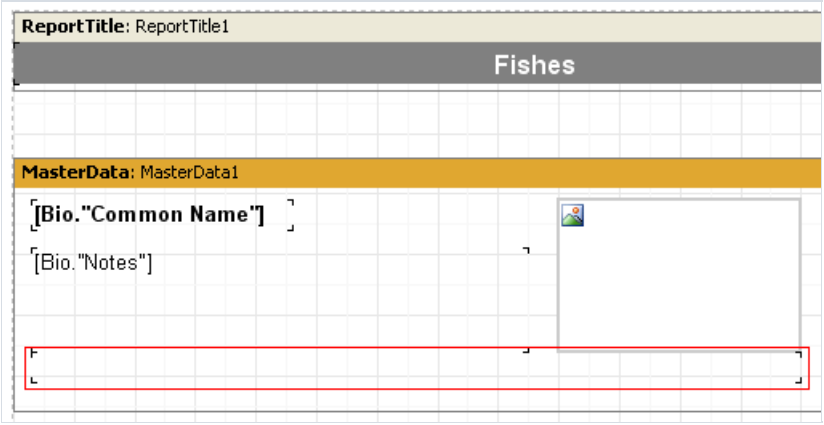
Deve ser observado que o algoritmo de divisão não é perfeito, e que a saída final pode não ser igual ao esperado. Esta opção deve ser usada com muito cuidado em casos onde objetos na banda dividida são agrupados de maneira complexa ou o tamanho das fontes difere. Segue um exemplo do que pode ser gerado.



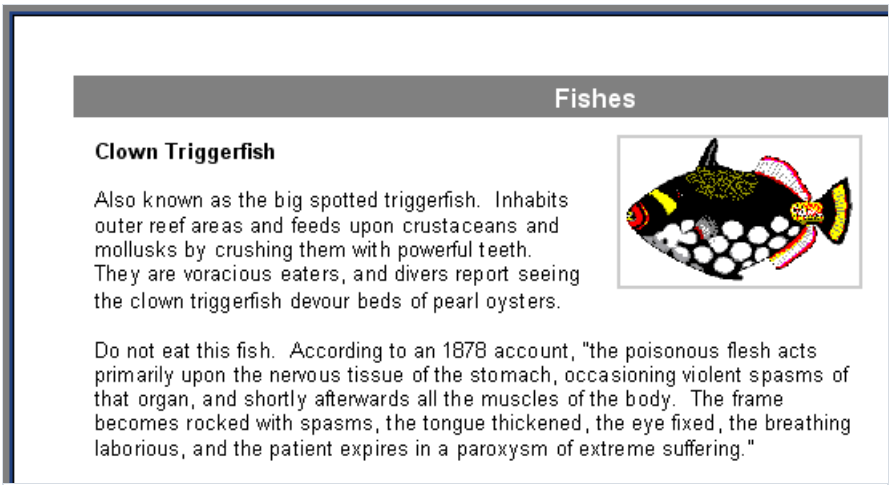
Quebra automática de linha de objetos

Em alguns designs de relatório pode ser necessário que o texto seja disposto ao redor de outros objetos (frequentemente ao usar imagens). Vamos demonstrar como o FastReport pode fazer isso em nosso exemplo atual.

Adicione mais um objeto "Texto" à banda de dados abaixo do objeto Bio."Notes", conforme mostrado aqui:



Iremos desabilitar o Esticar no objeto Bio."Notes" e habilitá-lo no objeto inferior. Para fazer o texto "fluir" do objeto Bio."Notes" para o objeto inferior, configure a propriedade `FlowTo` do objeto Bio."Notes" no inspetor de objetos – é uma lista suspensa. Selecione o nome do objeto inferior nesta lista. O relatório resultante terá a seguinte aparência:



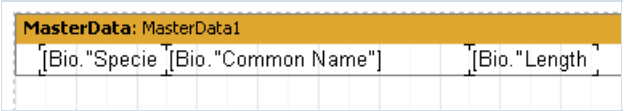
Quando o relatório for gerado, se o texto não couber no objeto superior, o excesso é deslocado para o objeto inferior. Ao arranjar os dois objetos em volta da imagem, é obtido o efeito da quebra automática de linha.

Observe: o objeto principal deve ser inserido no relatório antes de inserir o objeto vinculado, ou de outro modo o fluxo do texto pode não funcionar corretamente! Se isto ocorrer, selecione o objeto vinculado e traga-o para a frente usando o item de menu "Editar > Trazer para Frente".

Exibir dados na forma de uma tabela

Às vezes é necessário exibir dados na forma de uma tabela com borda. Um exemplo deste tipo de relatório é uma lista de preços. Para criar este tipo de relatório no FastReport, somente é necessário habilitar bordas para os objetos localizados na banda de dados. Vamos demonstrar diversas variações de bordas com um exemplo.

Crie um relatório similar a este:



Coloque os objetos "Texto" lado a lado na banda e diminua a altura da banda.

O primeiro tipo de tabela, e também o mais simples, tem uma borda completa nas células. Para obter isso, habilite todas as linhas da borda (encontradas na propriedade Frame.Typ) para cada objeto:



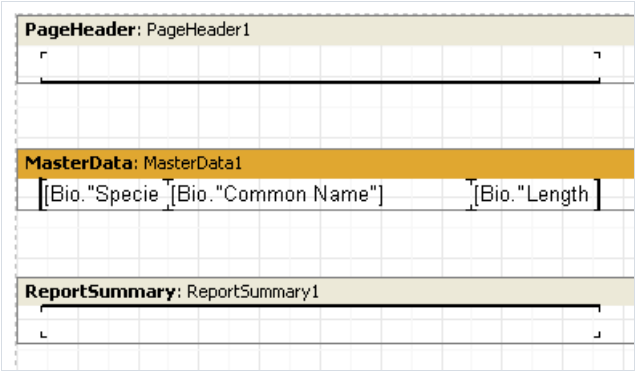
90020	Clown Triggerfish	50
90030	Red Emperor	60
90050	Giant Maori Wrasse	229
90070	Blue Angelfish	30
90080	Lunartail Rockcod	80

O próximo tipo de borda desenha somente linhas de células horizontais ou verticais, novamente usando a propriedade Frame.Typ:



90020	Clown Triggerfish	50
90030	Red Emperor	60
90050	Giant Maori Wrasse	229
90070	Blue Angelfish	30
90080	Lunartail Rockcod	80

Finalmente, para desenhar apenas uma borda externa para a tabela, o relatório precisa ser modificado:

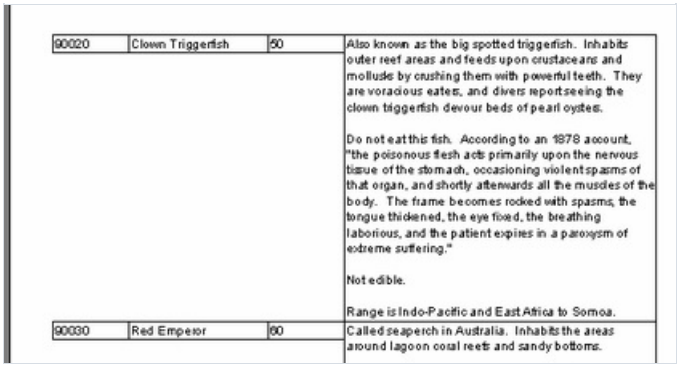


Você pode ver que adicionamos dois objetos "Texto", um na banda de cabeçalho da página e outro na banda de rodapé da página. As linhas de borda apropriadas para os objetos nas bordas da banda de dados foram habilitadas, resultando em um relatório com o seguinte aspecto:



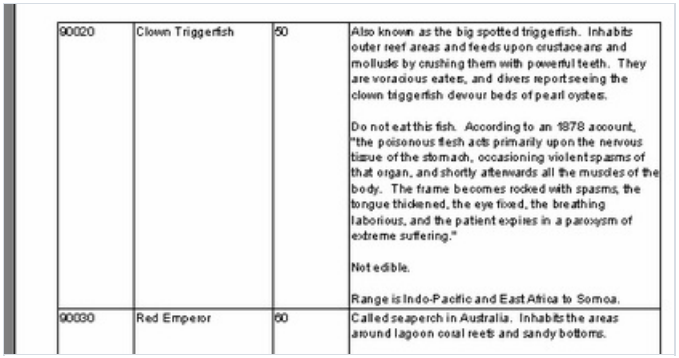
90020	Clown Triggerfish	50
90030	Red Emperor	60
90050	Giant Maori Wrasse	229
90070	Blue Angelfish	30
90080	Lunartail Rockcod	80
90090	Firefish	38

Todos esses exemplos continham bandas que tinham tamanhos fixos. Como é possível exibir uma tabela quando a banda é estendida? Vamos explicar como isso é feito através de um exemplo. Adicione um campo novo (um texto com várias linhas de Bio."Notes") a nosso relatório. Como você já aprendeu, a propriedade "Esticar" deve ser habilitada tanto para este objeto como para a banda no qual o objeto está localizado, para que a altura da banda seja alterada dependendo do tamanho do texto no objeto "Texto". O relatório gerado tem este aspecto::



90020	Clown Triggerfish	50	Also known as the big spotted triggerfish. Inhabits outer reef areas and feeds upon crustaceans and mollusks by crushing them with powerful teeth. They are voracious eaters, and divers report seeing the clown triggerfish devour beds of pearl oysters. Do not eat this fish. According to an 1878 account, "the poisonous flesh acts primarily upon the nervous tissue of the stomach, occasioning violent spasms of that organ, and shortly afterwards all the muscles of the body. The frame becomes rocked with spasms, the tongue thickened, the eye fixed, the breathing laborious, and the patient expires in a paroxysm of extreme suffering." Not edible. Range is Indo-Pacific and East Africa to Somoa.
90030	Red Emperor	60	Called seaperch in Australia. Inhabits the areas around lagoon coral reefs and sandy bottoms.

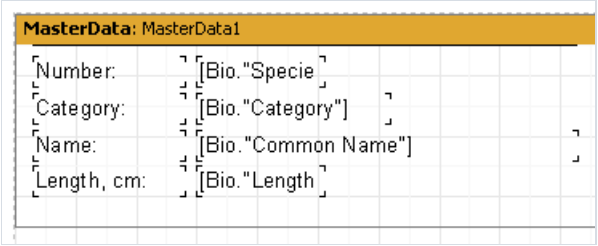
Isso difere um pouco do que precisamos – a aparência é melhor se as bordas dos objetos vizinhos também são estendidas. O FastReport pode solucionar este problema com facilidade. Configure a propriedade "StretchMode" como smMaxHeight no inspetor de objetos para todos os objetos que devem ser estendidos, e o núcleo do FastReport calculará a altura de banda máxima e "estenderá" objetos que têm Esticar habilitado até a extremidade inferior da banda. Já que bordas de objetos são estendidas juntas com o objeto, a aparência do relatório é alterada:



90020	Clown Triggerfish	50	Also known as the big spotted triggerfish. Inhabits outer reef areas and feeds upon crustaceans and mollusks by crushing them with powerful teeth. They are voracious eaters, and divers report seeing the clown triggerfish devour beds of pearl oysters. Do not eat this fish. According to an 1878 account, "the poisonous flesh acts primarily upon the nervous tissue of the stomach, occasioning violent spasms of that organ, and shortly afterwards all the muscles of the body. The frame becomes rocked with spasms, the tongue thickened, the eye fixed, the breathing laborious, and the patient expires in a paroxysm of extreme suffering." Not edible. Range is Indo-Pacific and East Africa to Somoa.
90030	Red Emperor	60	Called seaperch in Australia. Inhabits the areas around lagoon coral reefs and sandy bottoms.

Imprimir rótulos

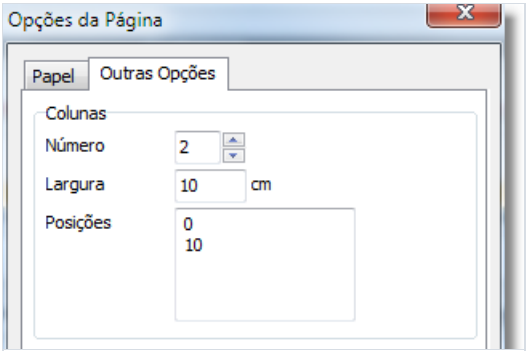
Ao contrário de relatórios do tipo tabela, outros relatórios (como relatórios de impressão de rótulos) podem ter os dados arranjados de maneira que um campo esteja abaixo do outro. Vamos dar uma olhada em um exemplo deste tipo de relatório que exibe dados sobre peixes, como no exemplo anterior. Os dados do relatório são apresentados como rótulos e possuem a seguinte estrutura:



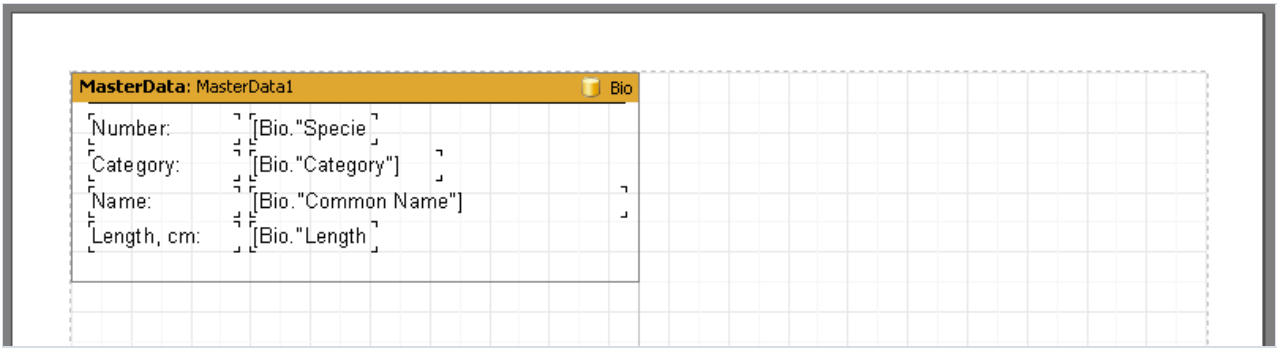
Ao visualizar, veríamos o seguinte:



Observe que há muito espaço vazio à direita da página. Para usar toda a página, é possível configurar o número de colunas nas quais são exibidos os dados nas configurações da página do relatório. Para fazer isso, clique duas vezes no espaço vazio na página de design ou use "Arquivo > Configurações da Página..." no menu.



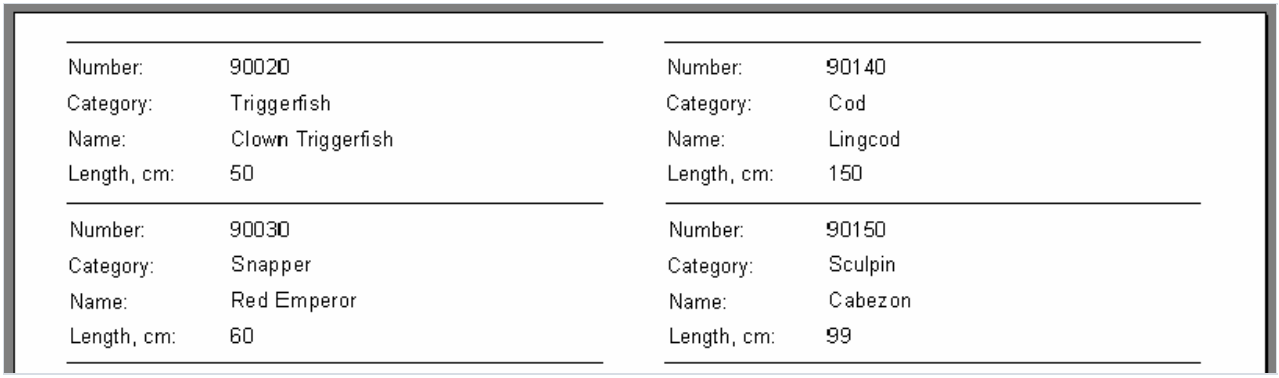
A contagem de colunas, a largura e as posições são configuradas na guia "Outras Opções" na caixa de diálogo. Em nosso exemplo, somente o Número precisa ser configurado como 2, ja que o FastReport ajusta as outras opções automaticamente. A borda da coluna é exibida no designer como uma linha vertical fina:



O relatório agora é criado da seguinte maneira. FastReport repete a banda "Dados de 1o. Nível" enquanto houver

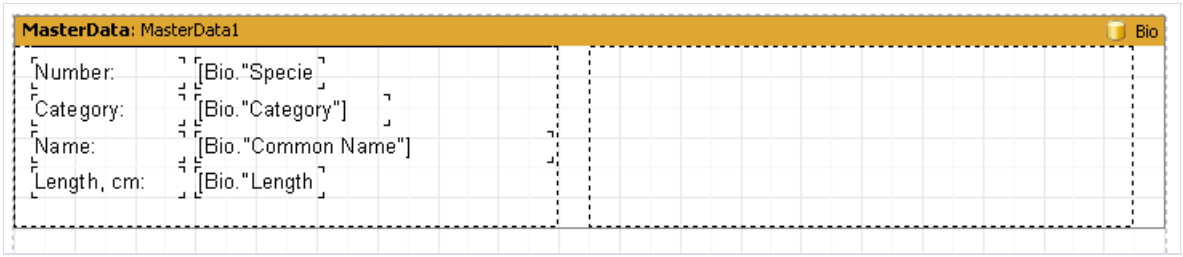
espaço em branco na parte inferior da página. Depois disso, é criada uma nova coluna na mesma página e as bandas de dados continuam a ser repetidas a partir do topo da coluna nova. Isso difere de relatórios simples, nos quais é criada uma nova página em vez de uma coluna nova. Quando a segunda coluna tiver sido preenchida, é criada uma terceira coluna, e assim por diante – apesar de nosso exemplo possuir somente duas colunas. Quando todas as colunas tiverem sido preenchidas, o FastReport cria uma nova página e continua exibindo os dados, começando novamente pela primeira coluna.

Nosso relatório de duas colunas deve ter esta aparência:



A propriedade "Columns", disponível no inspetor de objetos para todas as bandas de dados, oferece outra maneira de configurar o número de colunas. No entanto, se esta propriedade for diferente de zero, o número de colunas é configurado somente para a banda de dados selecionada, e não para toda a página (como no exemplo anterior). O efeito resultante é exibir os dados primeiro da 'esquerda para a direita e depois de cima para baixo', em vez da saída 'de cima para baixo e depois esquerda para a direita' mostrada acima.

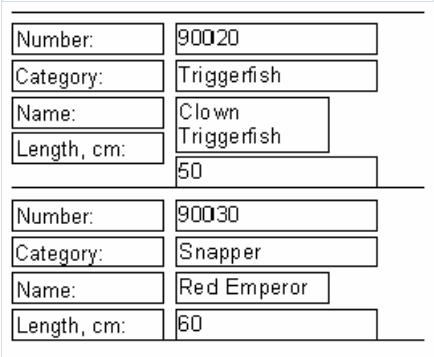
Desabilite as colunas na caixa de diálogo de Opções da Página (configure o Nùmero de colunas como 1) e insira 2 na propriedade `Columns` da banda de dados. Observe que a propriedade `ColumnWidth` também deve ser modificada do zero padrão para prevenir que a coluna 2 se sobreponha à coluna 1. De maneira opcional, a propriedade "ColumnGap" também pode ser alterada. O FastReport então exibe as bordas das colunas como linhas pontilhadas:



Este design exibe os dados na ordem 'esquerda para a direita e depois de cima para baixo'.

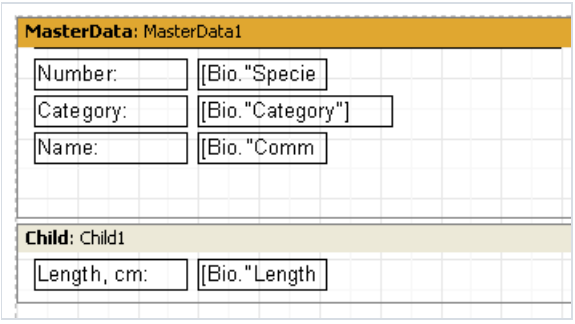
Bandas filho

Pode haver um problema quando um campo em um relatório tipo rótulo possui conteúdo de comprimento variável. Para simular isso em nosso exemplo, vamos reduzir a largura do objeto Bio."Common Name" para 2,5 cm e habilitar a propriedade "Esticar" neste objeto e também na banda "Dados de 1o. Nível". Habilite todas as linhas da borda para todos os objetos para que os efeitos da função de extensão sejam claramente visíveis. O design agora gera um relatório como este:

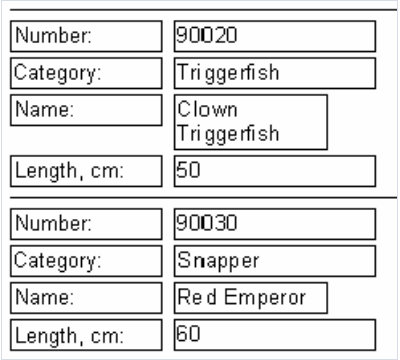


Aqui o primeiro objeto de campo Bio."Common Name" contém muito texto e é estendido por duas linhas. Isso causa o deslocamento para baixo do objeto de campo Bio."Length (cm)", localizado abaixo do primeiro. Isso acontece porque todos os objetos possuem a propriedade `ShiftMode` configurada como smAlways como padrão, o que significa que eles são deslocados para baixo se houver um objeto extensível (um objeto "Texto" com a propriedade "Esticar" habilitada) sobre eles. A distância do deslocamento depende de quanto o objeto acima é estendido.

Mas não desejamos que isso aconteça com nosso rótulo. Nós queremos que o objeto "Comprimento, cm:" também seja deslocado igualmente. Isso pode ser obtido ao usar uma banda de FastReport especial chamada de banda "Filho". Uma banda "Filho" é vinculada a (e exibida após) uma banda pai. Adicione uma banda "Filho" ao design e arraste os dois objetos "Texto" para dentro dela, conforme é mostrado aqui:



Vincule a banda DadosMestre à banda Filho ao configurar sua propriedade "Child" como "Child1" no inspetor de objetos. Agora, cada vez que a banda DadosMestre for impressa, a banda Filho é impressa imediatamente depois dela:

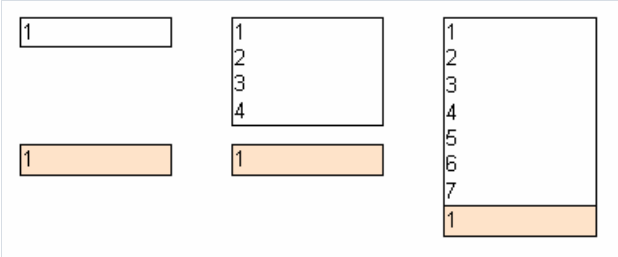


O título "Comprimento, cm:" agora está alinhado corretamente com seu campo de valor "50". Para prevenir que uma banda filho seja movida para a próxima página se não houver espaço livre suficiente na página (e se tornando 'órfã' de sua banda pai), habilite a propriedade `KeepChild` da banda pai no inspetor de objetos.

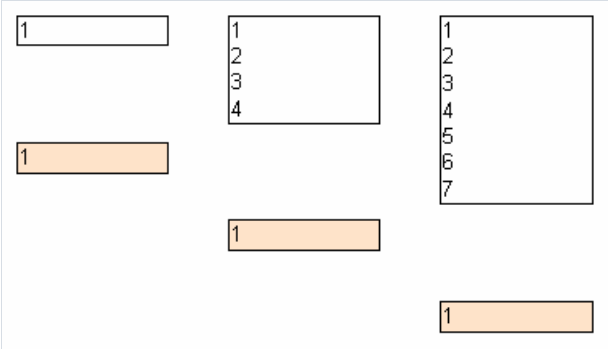
Deslocar objetos

Você já viu como 'smAlways' funciona na propriedade `ShiftMode`. Vamos dar uma olhada no próximo modo de deslocamento, 'smWhenOverlapped'. Neste modo, o deslocamento de objetos ocorre quando o objeto acima é estendido e é sobreposto ao objeto abaixo. Aqui há dois cenários:

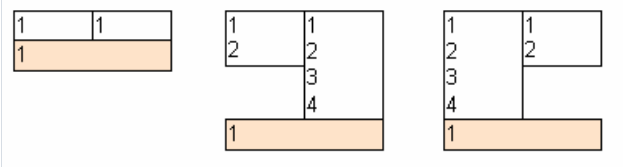
1: os três objetos no topo têm "Esticar" habilitado, e os três objetos inferiores têm `ShiftMode` configurado como 'smAlways'. Os objetos inferiores são deslocados apenas o suficiente para dar espaço ao objeto estendido acima deles:



2: os três objetos superiores têm "Esticar" habilitado, e os três objetos inferiors têm `ShiftMode` configurado como 'smWhenOverlapped'. Os objetos inferiores são deslocados para dar espaçõ para o objeto estendido, e também para manter a separação designada dos dois objetos:



Isso permite a criação de relatório complexos, especialmente em casos nos quais um objeto pode sobrepor diversos outros objetos acima dele ao mesmo tempo. No exemplo abaixo, todos os objetos superiores contêm texto extensível, e todos os objetos inferiores estão no modo 'smWhenOverlapped'. Os objetos inferiores sempre serão exibidos perto do objeto que estiver acima, não importando o comprimento do texto dos objetos superiores:



Relatório com dois níveis de dados (mestre-detalhe)

Até agora nossos relatórios de exemplo usaram somente uma banda de dados ("Dados de 1o. Nível" ou "Dados Mestre") para controlar a saída de dados. Isso era adequado para a saída de dados de uma única tabela de BD. FastReport também permite o design de relatórios com até seis níveis de dados, todos na mesma página de design. Além disso, é possível obter um número ilimitado de níveis de dados em relatórios ao usar o objeto "Sub-relatório" – este objeto será abordado mais adiante. Em geral, a maioria dos relatórios precisa de apenas um, dois ou três níveis de dados, e números maiores de níveis de dados são raros.

Vamos ver como fazer o design de um relatório com dois níveis de dados. O relatório irá exibir dados das tabelas de demonstração "Customer" e "Orders". A primeira tabela é uma lista de clientes e a segunda é uma lista de pedidos feitos por esses clientes. As tabelas contêm dados nos seguintes campos:

```
Customer:

CustNo   Company
1221     Kauai Dive Shoppe
1231     Unisco
1351     Sight Diver
...

Orders:

OrderNo   CustNo   SaleDate
1003      1351     12.04.1988
1023      1221     01.07.1988
1052      1351     06.01.1989
1055      1351     04.02.1989
1060      1231     28.02.1989
1123      1221     24.08.1993
...
```

Como você pode ver, a segunda tabela contém a lista de todos os pedidos feitos por todos os clientes. Para ver os pedidos da tabela Orders que foram feitos por um cliente da tabela Customers, as duas tabelas são vinculadas no campo "CustNo", que é comum a ambas as tabelas. A saída do relatório desses dados deve ter este aspecto:

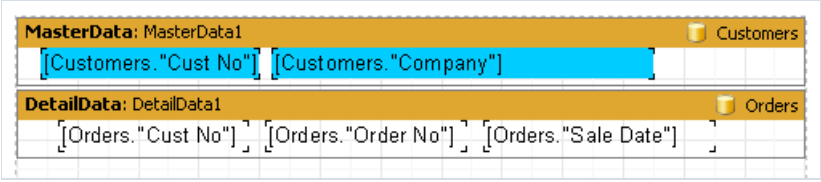
```
1221   Kauai Dive Shoppe
  1023   01.07.1988
  1123   24.08.1993
1231   Unisco
  1060   28.02.1989
1351   Sight Diver
  1003   12.04.1988
  1052   06.01.1989
  1055   04.02.1989
```

Crie um relatório novo. Coloque duas "Tabelas ADO" na página e configure suas propriedades:

```
ADOTable1:
TableName = 'Customer'
UserName = 'Customers'

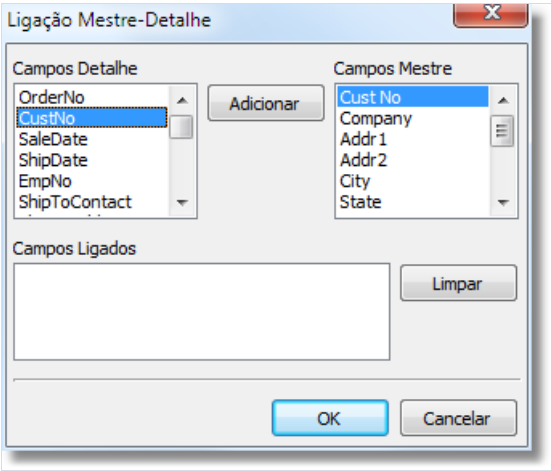
ADOTable2:
TableName = 'Orders'
UserName = 'Orders'
```

Agora adicione uma banda "Dados Mestre" e uma "Dados de Detalhe" à página:



Observe que a banda "DadosMestre" deve ser colocada acima da banda "DadosDeDetalhe"! Arraste-a para lá se necessário. Se a banda Mestre for colocada embaixo da banda Detalhe, o FastReport irá gerar uma mensagem de erro ao visualizar o relatório.

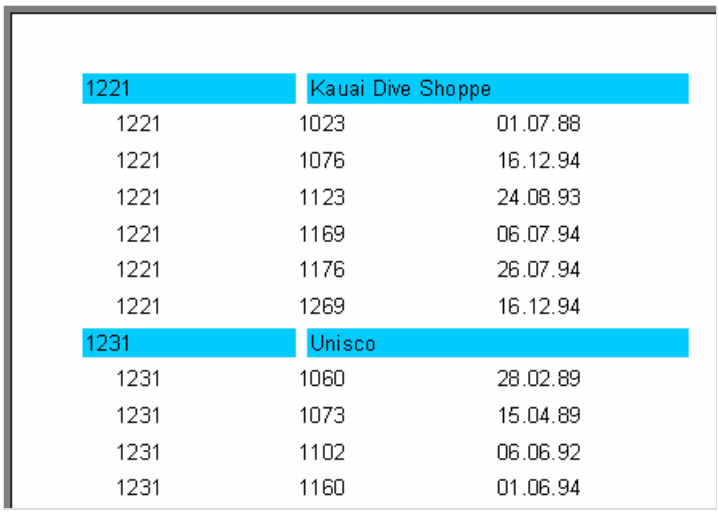
Se você visualizasse o relatório agora, você veria que a lista de pedidos permaneceria igual para todos os clientes e iria conter todos os registros da tabela "Orders". Isso aconteceria porque a propriedade `MasterSource` da tabela "Orders" não foi configurada. Configure "MasterSource – ADOTable1" no componente "ADOTable2". Agora configuramos uma relação 'mestre-detalhe'. Depois disso, selecionamos os campos a serem vinculados. Configure a propriedade `MasterFields` do componente "ADOTable2".



Precisamos vincular os campos "CustNo" das duas fontes. Para fazer isso, selecione os campos desejados e clique no botão "Adicionar". A vinculação do campo será exibida no painel inferior. Conclua ao fechar o editor clicando em OK.

Ao criar a visualização, FastReport faz o seguinte. Após a saída de um registro da tabela mestre (Customer), ele configura o filtro na tabela detalhe (Orders). Somente os registros que cumprem a condição 'Orders.CustNo = Customer.CustNo' permanecem na tabela. Isso significa que para cada cliente somente são exibidos na banda detalhe os pedidos que correspondem ao cliente autal. É importante enteder este conceito. Apesar das bandas de dados serem do tipo mestre ou detalhe, elas somente controlam o posicionamento dos dados na página de saída (ordem e número de exibições). Os dados exibidos pelos objetos nas bandas depende dos campos aos quais os objetos estão vinculados e da vinculação externa das duas tabelas.

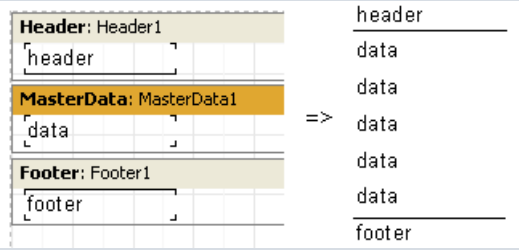
Esta é a saída final:



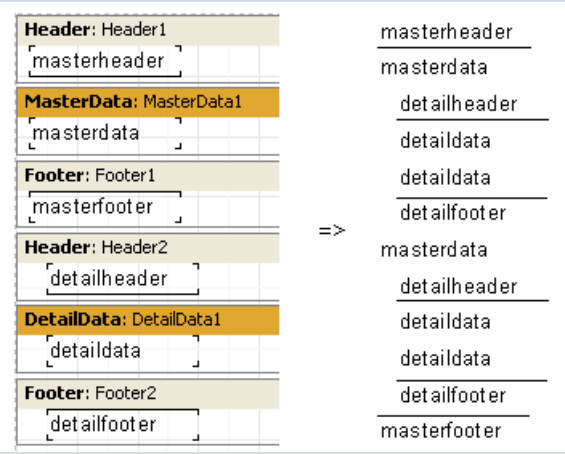
É possível construir relatórios com até 6 níveis de dados de maneira similar.

Cabeçalhos e rodapés de uma banda de dados

Cada banda de dados pode ter um cabeçalho e um rodapé. Os cabeçalhos são exibidos primeiro, depois todos os registros na banda de dados, e finalmente os rodapés. Aqui segue um exemplo de como os cabeçalhos e rodapés funcionam em um relatório simples:

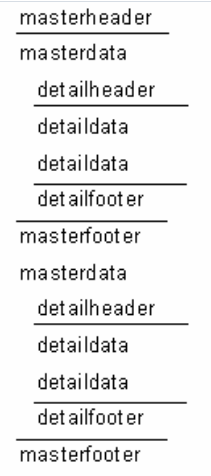


Vamos dar uma olhada em um exemplo mais complexo que utiliza dois níveis de dados – mestre e detalhe:



Como você pode ver, o cabeçalho é exibido antes de todos os registros de bandas de dados. Deste modo, o cabeçalho dos dados mestre é exibido primeiro, no início do relatório, e um cabeçalho de dados detalhe é exibido antes de cada grupo de bandas detalhe que pertencem à banda de registros mestre atual. O rodapé de detalhe é exibido depois do grupo de bandas detalhe que pertencem à banda de registros mestre, e o rodapé mestre não é exibido até que todos os registros da banda de dados mestre sejam exibidos.

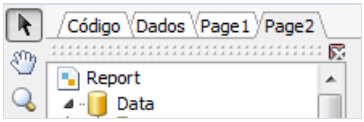
Ao usar a propriedade FooterAfterEach da banda de dados, é possível alterar este comportamento. Configurar esta propriedade como True (também é possível usar o menu de contexto da banda de dados – "Rodapé Após Cada Linha") resulta na exibição do rodapé após cada linha de dados. Isso pode ser útil no design de relatórios mestre-detalhe. Se FooterAfterEach da banda de dados mestre for configurado como True, o relatório será exibido assim:



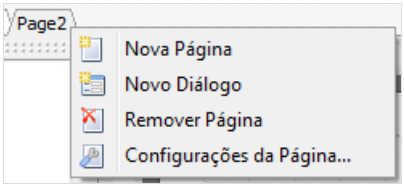
Relatórios com páginas múltiplas

Um relatório no FastReport pode conter diversas páginas de design. Designs com páginas múltiplas permitem o ajuste de propriedades como tamanho e orientação de cada página, assim como permitem variedade na colocação de objetos e bandas nas páginas. Quando este tipo de relatório é exibido, todas as bandas da primeira página de design são exibidas, e então as bandas da segunda página, etc.

Quando um relatório novo é criado no designer, ele contém uma página por padrão. É possível adicionar uma página nova ao clicar no botão na barra de ferramentas ou ao selecionar o comando de menu "Arquivo > Nova Página". Então você veria que é exibida uma nova guia de página no designer:

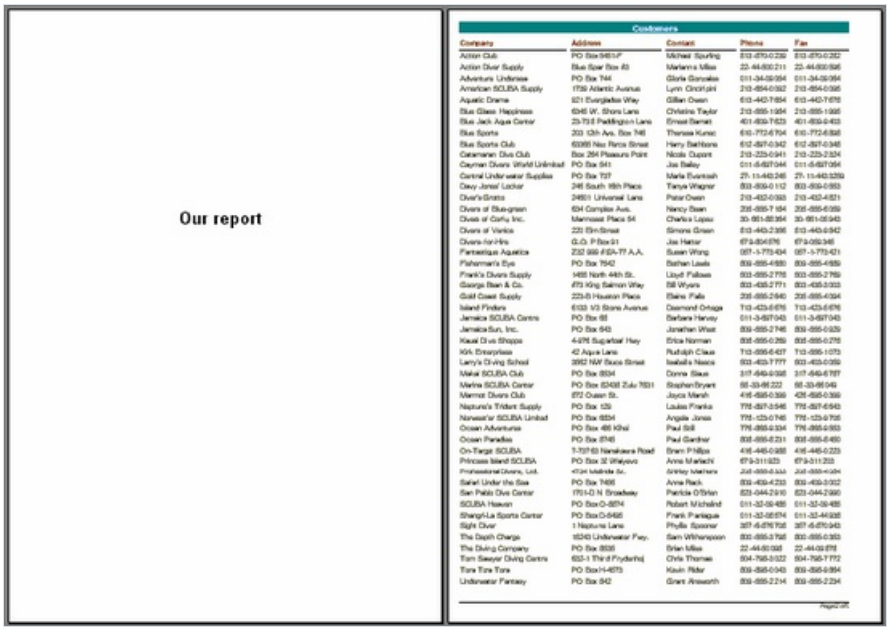


É fácil alternar as páginas ao clicar nas guias de páginas. As guias de páginas podem ser arrastadas ("arrastar e soltar") para mudar sua ordem de impressão com facilidade. Uma página desnecessária pode ser excluída com o botão na barra de ferramentas, ou ao selecionar o comando de menu "Editar > Remover Página". Também é possível usar o menu de contexto ao clicar com o botão direito na guia da página:



O número de páginas de design em um relatório é ilimitado. Como regra geral, páginas adicionais são usada como páginas de título ou, em relatórios mais complexos, para dados que têm origem em diversas fontes de dados.

Aqui há um exemplo simples de como criar uma página de título. Vamos usar nosso relatório anterior que possui um nível de dados. Adicione uma nova página a ele e ela será adicionada como uma segunda página. Mova-a para a frente do relatório ao clicar na guia Page2 e arrastá-la sobre a guia da primeira página, Page1. Isso altera a ordem das páginas. Selecione a página nova e coloque um objeto "Texto" contendo "Nosso relatório" no meio da página. Isso é tudo. O relatório com uma página de título está concluído:



No entanto, preste atenção a um recurso de relatórios com páginas múltiplas. Se a propriedade

"PrintOnPreviousPage" for habilitada no inspetor de objettos para a segunda página de saída, então os objetos da segunda página de saída serão impressos no espaço em branco da primeira página de saída e não na nova página de saída.

Propriedades RowCount e PageCount

Às vezes surge a necessidade de exibir dados estáticos diversas vezes, por exemplo ao imprimir cartões de visitas ou cartões postais "em branco". Para isso as bandas de dados possuem a propriedade `RowCount` e a página do relatório possui a propriedade `PageCount`.

Essas duas propriedades controlam o grau de repetição de bandas/páginas no relatório, sem serem influenciadas pelos dados do relatório.

Relatório com grupos

No exemplo anterior criamos um relatório de dois níveis baseado nos dados de duas tabelas. É possível construir outro relatório com o mesmo aspecto no FastReport, que desta vez é baseado em um conjunto de dados obtido de uma consulta conjunta.

Para fazer isso é necessário uma consulta SQL que retorna dados de ambas as tabelas ordenado de maneira específica. Em nosso exemplo, as tabelas serão unidas nos campos "CustNo", que estão presentes em ambas as tabelas. A consulta pode ser:

```
select * from customer, orders
where orders.CustNo = customer.CustNo
order by customer.CustNo
```

A linha "order by" é necessária para ordenar os registros no campo "CustNo". Esta consulta retorna um conjunto de dados parecido com isto:

```
CustNo   Company             OrderNo   SaleDate
1221     Kauai Dive Shoppe   1023      01.07.1988
1221     Kauai Dive Shoppe   1123      24.08.1993
1231     Unisco              1060      28.02.1989
1351     Sight Diver         1003      12.04.1988
1351     Sight Diver         1052      06.01.1989
1351     Sight Diver         1055      04.02.1989
```

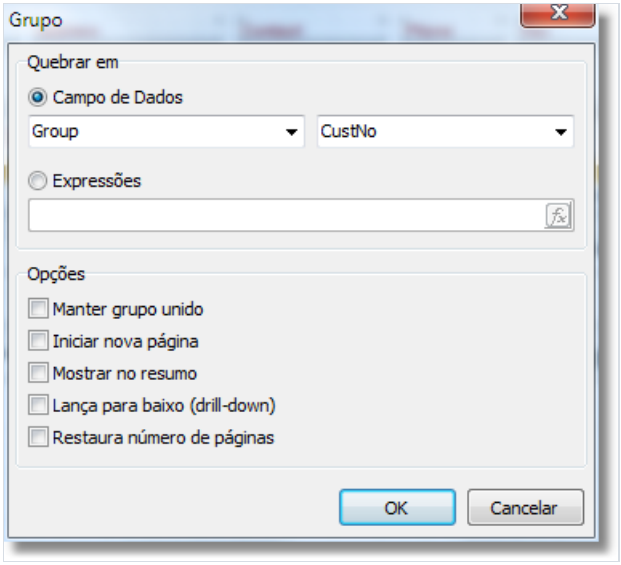
Como é possível fazer o design de um relatório de vários níveis usando estes dados? No FastReport há uma banda especial, o "Cabeçalho de Grupo". Uma condição (um campo de BD ou uma expressão) é especificada para a banda; a banda é exibida sempre que o valor da condição for alterado. O exemplo a seguir ilustra isso.

Crie um relatório novo no designer do FastReport. Coloque um componente "Consulta ADO" na página e configure suas propriedades:

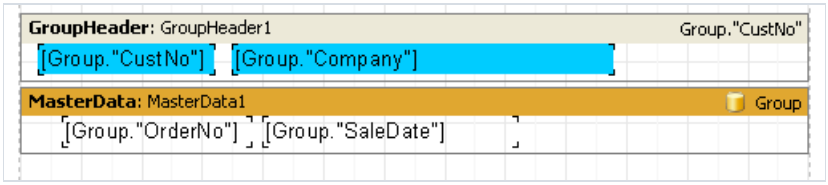
```
SQL =
  select * from customer, orders
  where orders.CustNo = customer.CustNo
  order by customer.CustNo

UserName = 'Group'
```

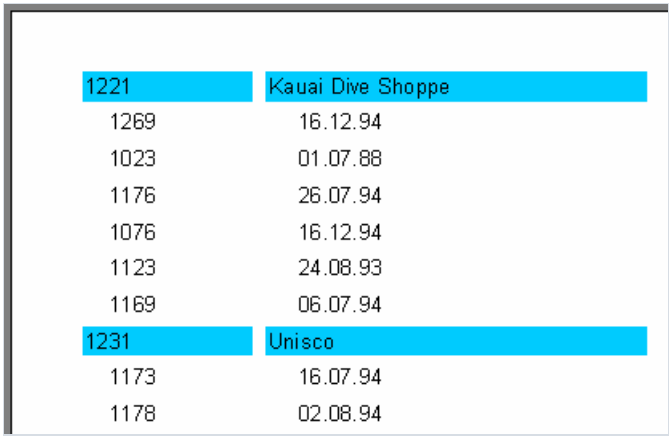
Depois adicione uma banda "Cabeçalho de Grupo" ao relatório. Defina uma condição (neste caso o campo de dados "Group.CustNo") no editor da banda "Cabeçalho de Grupo":



Também vincule a banda de dados à fonte de dados "Group" e arranje alguns objetos como mostrado abaixo (observe que o cabeçalho do grupo deve ser colocado acima da banda de dados):



Ao visualizar o relatório, obtemos uma saída similar a isto:



Como você pode ver, a banda "Cabeçalho de Grupo" é exibida somente quando o valor do campo ao qual é vinculada é alterado. Caso contrário, a banda de dados conectada ao conjunto de dados "Group" é exibida. Se compararmos este relatório ao relatório mestre-detalhe criado anteriormente, é óbvio que os números de pedidos não são ordenados. Isso pode ser corrigido com facilidade ao alterar a cláusula 'order by' da consulta SQL:

```
select * from customer, orders
where orders.CustNo = customer.CustNo
order by customer.CustNo, orders.OrderNo
```

É possível fazer o design de relatórios que possuam grupos aninhados de maneira similar. A profundidade de grupos aninhados é ilimitada. Relatórios que usam grupos têm algumas vantagens sobre relatórios do tipo mestre-detalhe:

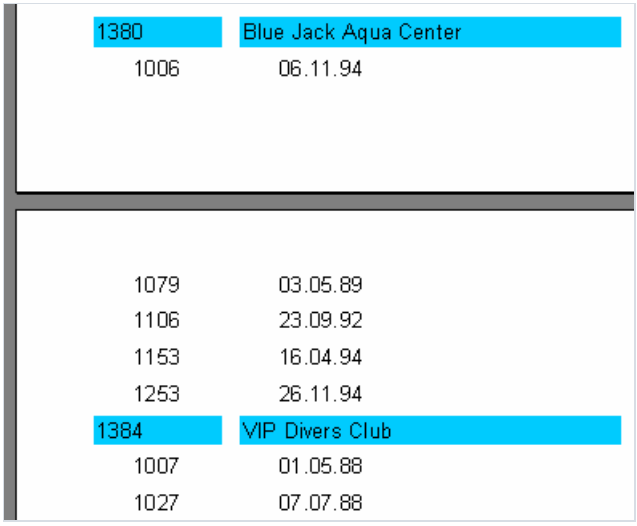
- todo o relatório precisa de somente um conjunto de dados (consulta)
- o número de níveis de agrupamento de dados é ilimitado

- é possível ordenar os dados
- uso mais otimizado dos recursos de BD (a consulta retorna apenas um conjunto de dados para saída e a filtragem é executada pela consulta).

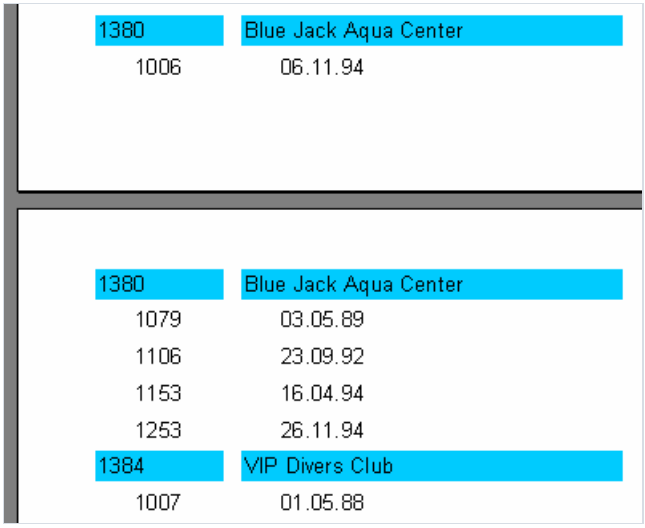
A única desvantagem é a necessidade de escrever consultas na linguagem SQL. No entanto, o conhecimento básico de SQL é praticamente obrigatório para qualquer programador que trabalhe com bancos de dados.

Outros recursos de grupos

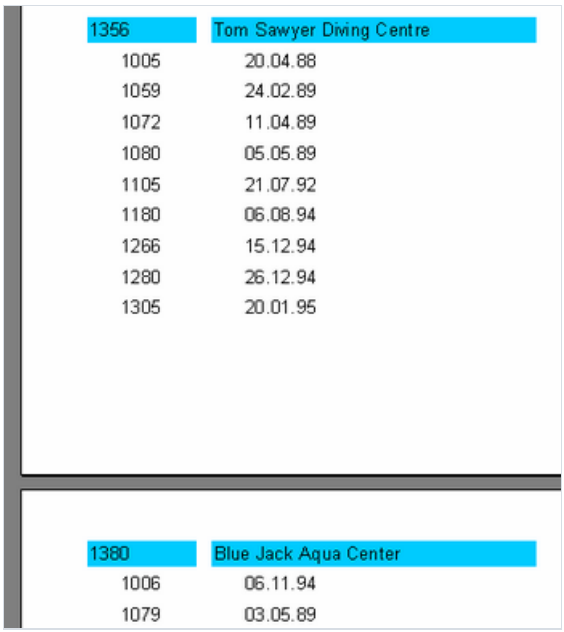
Vamos ver como o grupo é exibido quando o relatório precisa de uma página nova:



Ao olhar para o relatório, não é claro a qual cliente se refere a lista de pedidos no topo da segunda página. FastReport permite que o cabeçalho do grupo seja exibido na página nova (o que em nosso caso identifica o cliente). Para fazer isso, habilite a propriedade `ReprintOnNewPage` da banda "Cabeçalho do Grupo" usando o inspetor de objetos ou o menu de contexto. Isso irá alterar o relatório desta maneira:



Há outra maneira de evitar quebrar grupos nos limites de páginas, que é habilitar a propriedade `KeepTogether` do cabeçalho do grupo no inspetor de objetos ou no menu de contexto. Dessa maneira, se o grupo inteiro não couber no espaço livre na página de saída, ele é movido inteiro para uma página nova. Em nosso exemplo, ele é exibido deste modo:



Pode haver muito espaço vazio em algumas páginas, mas, sempre que possível, o grupo é exibido de forma completa em uma página.

A propriedade `StartNewPage` do cabeçalho de grupo permite a exibição de grupos em páginas separadas. Isso pode levar a um desperdício de papel, mas pode ser útil em algumas situações.

Redefinir números de páginas

A banda "Cabeçalho de Grupo" possui a propriedade `ResetPageNumbers`, que nos permite redefinir números de páginas ou imprimir um grupo. Para que serve?

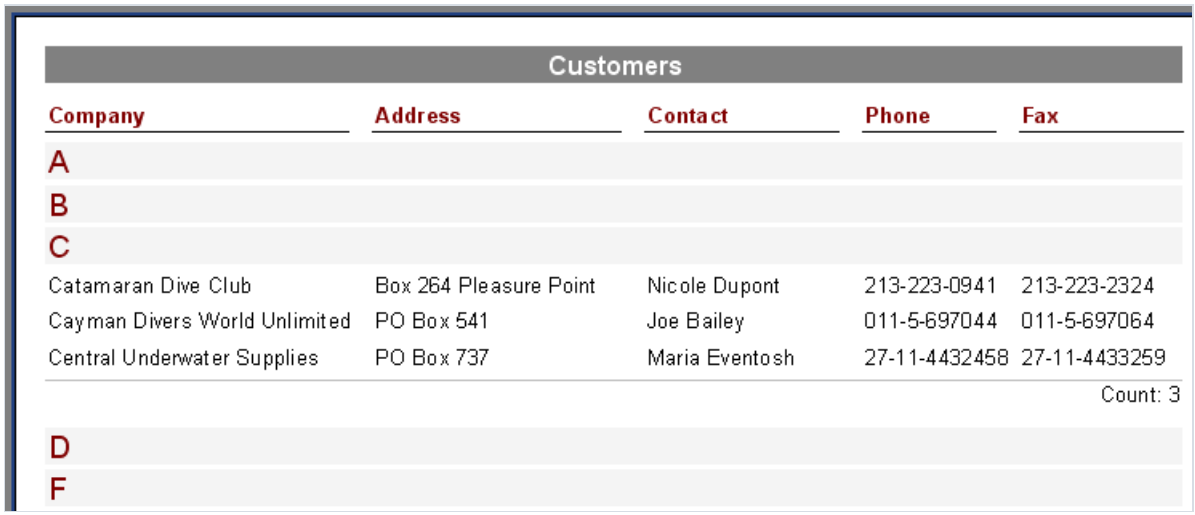
Aqui está um exemplo: você criou um relatório que coloca o nome do cliente em um cabeçalho de grupo e os pedidos do cliente na banda de dados. Agora você precisa imprimir o relatório e enviá-lo a todos os seus clientes, e cada cliente deve receber apenas as páginas do relatório que se referem a ele. Lamentavelmente, a numeração das páginas é contínua no relatório, então um cliente que receber as páginas 50 a 52 perguntará "Onde estão as primeiras 49 páginas?". Para evitar este tipo de situação, é necessário numerar as páginas de cada cliente com sua própria sequência. No relatório, cada grupo terá páginas numeradas a partir de 1.

Observe: se configurar `ResetPageNumbers` como True, também deverá configurar `StartNewPage` como True, para que cada grupo inicie em uma página nova. Para imprimir o número da página ou as páginas totais, você deve usar as variáveis de sistema [Page] e [TotalPages], e não [Page#] e [TotalPages#].

Grupos de drill down

O cabeçalho de grupo possui uma propriedade chamada `DrillDown`. Se ela for configurada como True, o grupo se torna interativo. Isso significa que é possível clicar no cabeçalho do grupo na janela de visualização e o grupo irá se expandir (exibir todos os registros no grupo) ou recolher (exibir somente o cabeçalho e, se `ShowFooterIfDrillDown` for True, o rodapé).

Segue um exemplo deste tipo de grupo com um cabeçalho expandido:



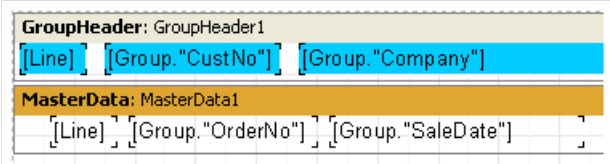
Customers

Company	Address	Contact	Phone	Fax
A				
B				
C				
Catamaran Dive Club	Box 264 Pleasure Point	Nicole Dupont	213-223-0941	213-223-2324
Cayman Divers World Unlimited	PO Box 541	Joe Bailey	011-5-697044	011-5-697064
Central Underwater Supplies	PO Box 737	Maria Eventosh	27-11-4432458	27-11-4433259
				Count: 3
D				
F				

É possível controlar se todos os grupos são recolhidos ou expandidos quando o relatório é executado pela primeira vez. Como padrão, um grupo permanece recolhido, mas é possível configurar `ExpandDrillDown` como True se desejar que seja expandido. Também é possível usar o menu de contexto da visualização para expandir ou recolher todos os grupos ao mesmo tempo.

Numeração de linhas

Vamos usar nosso exemplo para mostrar como numerar as linhas em um grupo. Para fazer isso, adicionamos um objeto "Texto" que contém uma variável de sistema [Line] a ambas as bandas (a maneira mais fácil de fazer isso é ao arrastar e soltar da guia "Variáveis" no painel "Árvore de Dados").

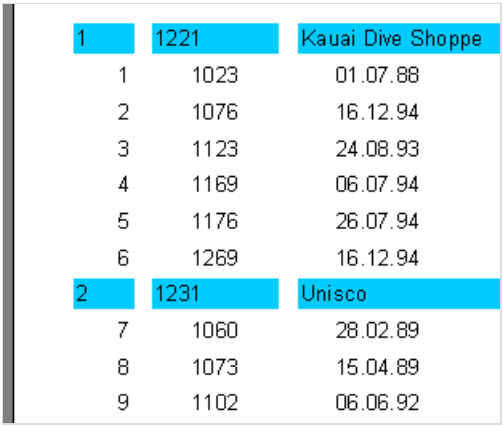


Ao visualizer o relatório, podemos ver que ambos os níveis de dados agora possuem seus próprios números de linha:



1	1221	Kauai Dive Shoppe
1	1023	01.07.88
2	1076	16.12.94
3	1123	24.08.93
4	1169	06.07.94
5	1176	26.07.94
6	1269	16.12.94
2	1231	Unisco
1	1060	28.02.89
2	1073	15.04.89
3	1102	06.06.92
4	1160	01.06.94

Para numerar de maneira contínua as linhas de dados do segundo nível, use a variável [Line#] em vez de [Line] no objeto "Texto" na banda de dados. O resultado então terá este aspecto:



1	1221	Kauai Dive Shoppe
1	1023	01.07.88
2	1076	16.12.94
3	1123	24.08.93
4	1169	06.07.94
5	1176	26.07.94
6	1269	16.12.94
2	1231	Unisco
7	1060	28.02.89
8	1073	15.04.89
9	1102	06.06.92

Funções de agregação

Na maioria dos casos, relatórios de grupo precisam exibir algum resumo das informações (como: "total de um grupo", "número de elementos do grupo", etc.). FastReport fornece funções de agregação para calcular valores de agregação em um intervalo de dados. As funções de agregação são:

Function	Description
`SUM`	retorna o total de uma expressão
`MIN`	retorna o valor mínimo de uma expressão
`MAX`	retorna o valor máximo de uma expressão
`AVG`	retorna o valor médio de uma expressão
`COUNT`	retorna o número de linhas (fileiras) no intervalo de dados

A sintaxe de todas as funções de agregação (exceto `COUNT`) é similar àquela da função `SUM`:

```
SUM(expressão, banda, sinalizadores)
SUM(expressão, banda)
SUM(expressão)
```

Os parâmetros são:

`expressão` – a expressão a ser calculada

`banda` – o nome da banda de dados na qual é executado o cálculo

`sinalizadores` – um campo de bits com os valores

1 : incluir bandas invisíveis no cálculo
2 : acumular o agregado como uma soma acumulada (não redefinir o agregado quando o intervalo de dados atual for redefinido)
3 : (ambas as opções anteriores)

Uma expressão é o único parâmetro obrigatório, os outros dois são opcionais. No entanto, para evitar cometer erros é recomendado que os parâmetros da banda sejam sempre fornecidos.

A função de agregação `COUNT` possui a seguinte sintaxe:

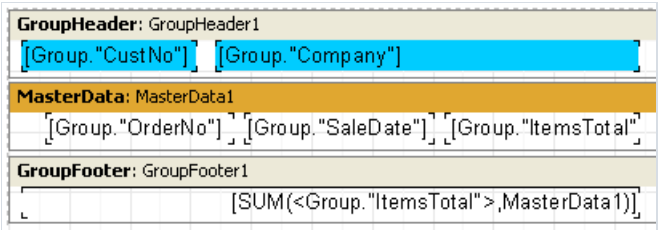
```
COUNT(banda, sinalizadores)
COUNT(banda)
```

na qual os parâmetros têm o mesmo significado explicado acima.

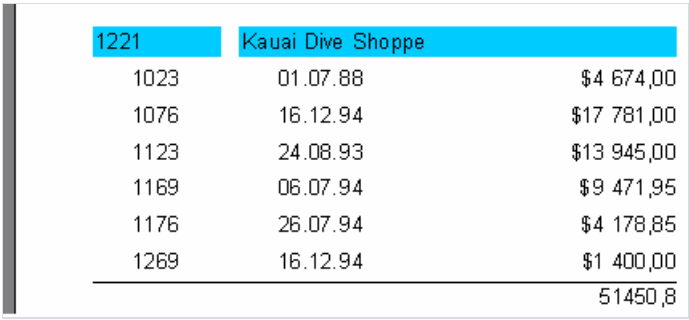
Há uma regra geral para todas as funções de agregação: uma função de agregação somente pode ser calculada sobre uma banda de dados, e ela somente pode ser usada no rodapé da banda, que pode pertencer a estas

categorias: rodapé, rodapé da página, rodapé do grupo, rodapé da coluna ou rodapé do relatório (banda de resumo).

Como funcionam as funções de agregação? Abordaremos isso com nosso relatório de grupo como exemplo. Vamos adicionar alguns elementos novos ao relatório:



O campo Group."ItemsTotal" na banda de dados exibe o total de pedidos atual. Coloque um objeto "Texto" no rodapé do grupo com a função de agregação `SUM` exibida acima. Ele exibirá o total de todos os pedidos feitos por um cliente. Usando uma calculadora, podemos verificar que o resultado está correto:



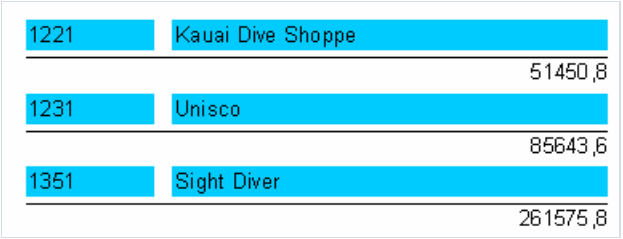
1221	Kauai Dive Shoppe	
1023	01.07.88	$4 674,00
1076	16.12.94	$17 781,00
1123	24.08.93	$13 945,00
1169	06.07.94	$9 471,95
1176	26.07.94	$4 178,85
1269	16.12.94	$1 400,00
		51450,8

Funções de agregação funcionam desta maneira: antes de exibir o relatório, FastReport examina o conteúdo do objeto "Texto" para encontrar funções de agregação. As funções encontradas são vinculadas às bandas de dados em seus parâmetros (em nosso exemplo, `SUM` é vinculada à banda "DadosMestre1"). Na exibição do relatório (quando a banda de dados é exibida) são calculados os valores das funções de agregação vinculadas. Em nosso caso, os valores do campo Group."ItemsTotal" são acumulados. Quando o rodapé do grupo com a função de agregação tenha sido exibido, o valor da função é redefinido para zero, e o ciclo é repetido para o próximo grupo, e assim por diante.

Qual é o propósito do parâmetro opcional `Sinalizadores` em funções de agregação? Os relatórios podem ocultar algumas ou até todas as bandas de dados. Podemos, no entanto, precisar calcular as funções de agregação de todas as bandas de dados, estejam elas visíveis ou não. Em nosso exemplo, configure a propriedade `Visible` da banda de dados como false, prevenindo sua exibição. Para que esta banda de dados oculta seja incluída nos cálculos, é necessário configurar o terceiro parâmetro opcional na chamada da função como 1:

```
[SUM(<Group."ItemsTotal">,MasterData1,1)]
```

Isso produz um relatório com o seguinte aspecto:

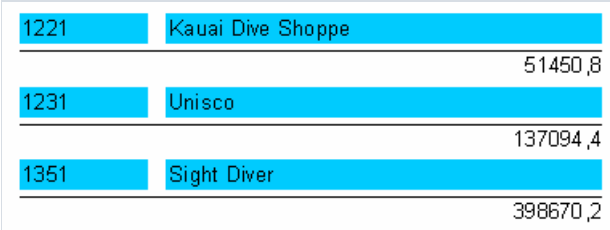


1221	Kauai Dive Shoppe	
		51450,8
1231	Unisco	
		85643,6
1351	Sight Diver	
		261575,8

Se o parâmetro `Sinalizadores` for configurado como 2, o valor da função de agregação não é redefinido imediatamente após a exibição: a função se torna um cálculo "contínuo" para cada saída sucessiva. Vamos modificar a chamada da função desta maneira:

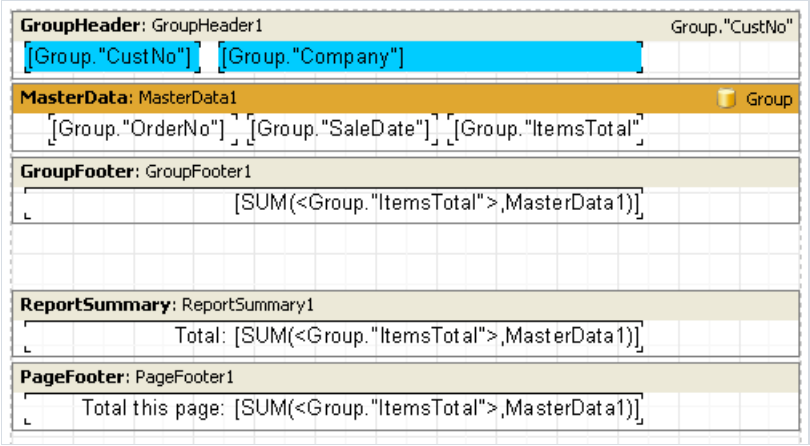
```
[SUM(<Group."ItemsTotal">,MasterData1,3)]
```

O valor "3" é uma combinação de bits de "1" e "2", o que significa que precisamos incluir as bandas invisíveis sem redefinir o total após cada grupo. Como resultado obtemos:

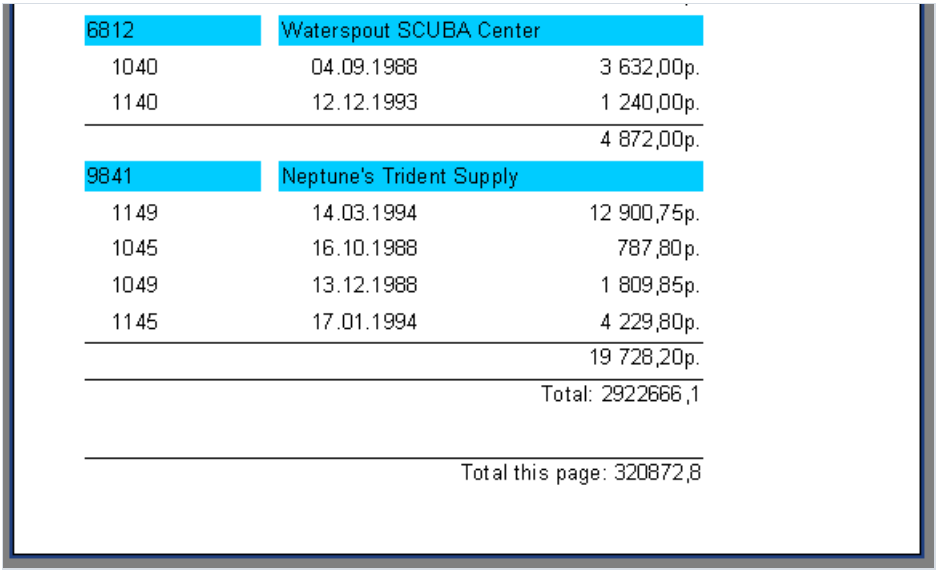


Totais da página e do relatório

Com frequência é necessário exibir totais de resumo de uma página ou um relatório inteiro. Também é possível usar funções de agregação nesta situação. Demonstraremos isso ao fazer algumas alterações em nosso exemplo:



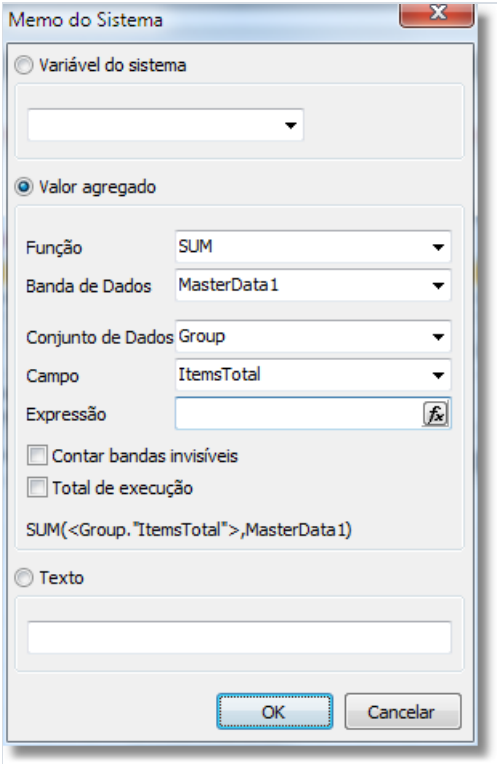
Como você pode ver, adicionamos uma banda "Sumário do Relatório" que contém um objeto "Texto" com a função de agregação SUM tanto na banda "Sumário do Relatório" como na banda "Rodapé da Página". Isso é tudo que precisamos fazer:



Inserting aggregate function

Até agora inserimos as funções de agregação manualmente em objetos "Texto". Agora vamos dar uma olhada em outras maneiras de inserir funções de agregação.

Primeiramente, podemos usar o objeto "Texto do Sistema" para exibir uma função de agregação. Na verdade, este objeto é parecido com um objeto "Texto" e possui seu próprio editor especial para especificar variáveis de sistema ou funções de agregação com mais facilidade:



Selecione um tipo de função, depois uma banda de dados (sobre a qual será calculada a função de agregação) e finalmente um campo de BD ou uma expressão cujo valor deve ser calculado. Também é possível configurar os sinalizadores "Contar bandas invisíveis" e "Total de execução", se necessário.

O segundo método é usar um objeto "Texto" e clicar no botão em seu editor. Isso abre uma caixa de diálogo parecida com o editor do objeto "Texto do Sistema". Ao clicar no botão OK, é inserida uma chamada da função de agregação no texto do objeto.

Formatar, realçar

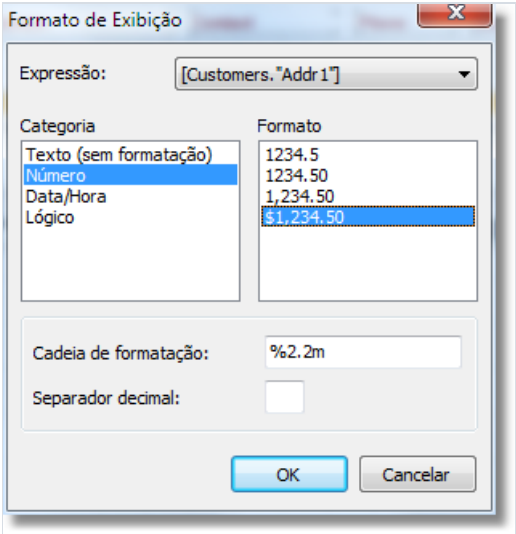
Formatar valores

Um recurso de funções de agregação é que os valores numéricos de retorno não são formatados, como é mostrado no primeiro exemplo, que usa "SUM":

1176	26.07.94	$4 178,85
1269	16.12.94	$1 400,00
		51450,8

Os campos de dados normalmente retornam um valor formatado, que é exibido pelo objeto "Texto" sem nenhuma alteração. Para aplicar a formatação ao resultado de `SUM`, vamos usar as ferramentas de formatação de valores do FastReport.

Selecione o objeto que contém a função Sum e abra o editor de formatação ao clicar em 'Formato de Exibição...' em seu menu de contexto ou usar a propriedade `DisplayFormat` no inspetor de objetos.



Este editor lista as categorias de formatação à esquerda, os formatos correspondentes à direita e a cadeia de formatação e o separador decimal da categoria e formato selecionados abaixo. Vamos selecionar a categoria "Número" e o formato "$1,234.50". A cadeia de formatação e o separador decimal podem ser alterados. Se o separador decimal for deixado em branco, é utilizado o valor da configuração regional atual.

Após clicar em OK e visualizar o relatório, você verá que a função Sum no relatório agora está formatada corretamente:

1176	26.07.94	$4 178,85
1269	16.12.94	$1 400,00
		$51450,80

Observe a caixa de combinação no topo da caixa de diálogo. Se houver mais de uma expressão em um objeto, é possível configurar formatações diferentes para cada expressão.

Formatação embutida

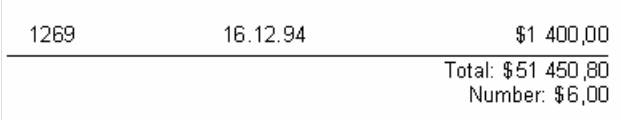
A formatação embutida permite configurar uma formatação diferente para cada expressão contida no objeto. Ela era usada em versões anteriores do FastReport e agora é obsoleta. Utilize a caixa de diálogo de formatação para configurar uma formatação diferente para cada expressão.

Usando o exemplo, redimensione o rodapé e seu objeto e altere o texto do objeto para isto:

```
Total: [SUM(<Group."ItemsTotal">,MasterData1)]
Number: [COUNT(MasterData1)]
```

O total e o número de pedidos serão exibidos no objeto.

Na visualização do relatório, ambos os valores são exibidos no formato monetário que configuramos anteriormente. Isso é errado:



Para exibir cada valor em seu formato correto, eles devem ser formatados individualmente. Para fazer isso usamos tags de formatação, que são colocadas antes do colchete de fechamento da expressão. Em nosso exemplo, desabilite a formatação no objeto (selecione a categoria "Texto (sem formatação)" no editor de formatação). Agora somente precisamos especificar o formato da primeira expressão, já que a segunda será exibida corretamente por padrão (como número inteiro). Modifique o texto do objeto deste modo:

```
Sum: [SUM(<Group."ItemsTotal">,MasterData1) #n%2,2m]
Number: [COUNT(MasterData1)]
```

Visualize o relatório para certifica-se de que o objeto é exibido corretamente:



A sintaxe geral de tags de formatação é:

`[expression #formattag]`

Observe que o caractere de espaço entre a expressão e o símbolo "#" é obrigatório! A tag de formatação em si pode ter este aspecto:

`#nformat_string` – formato numérico

`#dformat_string` – formato data/hora

`#bFalse,True` – formato booleano

Em todos os casos, `Format_string` é o argumento da função usado para a formatação. Desta maneira, a função `Format` é usada na formatação numérica, e a função `FormatDateTime` é usada para data/hora. Abaixo são mostrados diversos valores usados no FastReport:

para formatação numérica:

`%g` – número com dígitos mínimos após o ponto decimal

`%2.2f` – número com um número fixo de dígitos após o ponto decimal

`%2.2n` – como o anterior, mas com separador de milhares

`%2.2m` – formato monetário, aceito pelo SO Windows, depende das configurações regionais no painel de controle

para formatação de data/hora:

`dd.mm.yyyy` – data como '23.12.2003'

`dd mmm yyyy` – data como '23 Nov 2003'

`dd mmmm yyyy` – data como '23 Novembro 2003'

`hh:mm` – hora como '23:12'

`hh:mm:ss` – hora como '23:12:00'

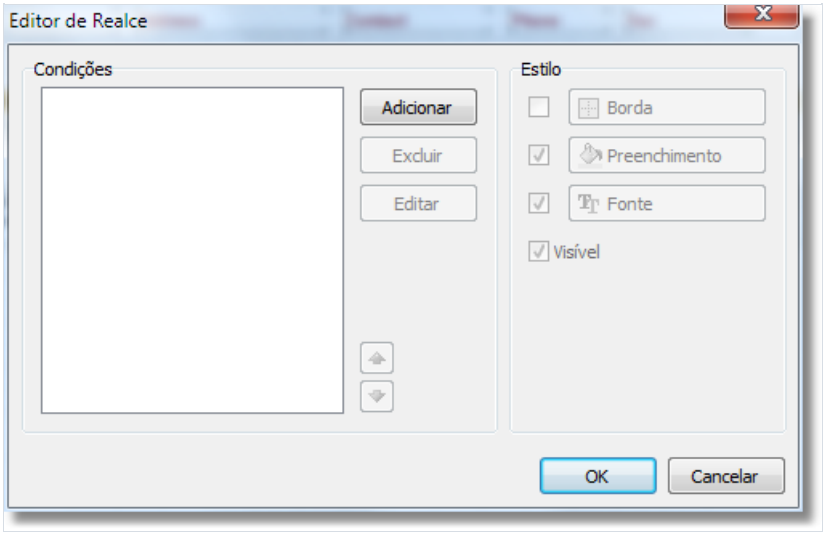
`dd mmmm yyyy, hh:mm` – data e hora como '23 Novembro 2003, 23:12'

Pode ser usada uma vírgula ou um hífen em vez do ponto em `format_string` para a formatação numérica. Este símbolo é usado como separador entre o número inteiro e a fração do valor. Nenhum outro caractere é permitido.

Para a formatação do tipo `#b` (booleana), a cadeia format_string é inserida como dois valores separados por vírgula. O primeiro valor corresponde a "False" e o segundo a "True".

Realce condicional

Há a possibilidade de alterar a aparência do objeto "Texto" dependendo das condições existentes. Por exemplo, um objeto pode ser realçado com a cor vermelha se ele possuir um valor negativo. Este recurso é chamado de "realce condicional". Para configurá-lo, selecione o objeto "Texto" e clique no botão na barra de ferramentas "Texto". A seguinte caixa de diálogo é exibida:



É possível definir uma ou mais condições e configurar o estilo de cada condição. O estilo pode conter uma ou mais configurações:

· borda;

· preenchimento;

· fonte;

· visibilidade do objeto.

Você pode indicar quais configurações devem ser alteradas quando a condição for cumprida. Para isso é necessário marcar a configuração desejada usando a caixa de seleção.

Para criar uma condição nova, clique no botão "Adicionar". O editor de expressões é exibido. Nele é possível escrever qualquer expressão que retornar um resultado booleano. Em muitos casos você usará a variável `Value`, que contém o valor de impressão atual.

Vamos dar uma olhada no seguinte exemplo: temos um objeto "Texto", o qual imprimimos a quantidade de produtos em estoque:

```
[Products."UnitsInStock"]
```

Queremos deixar o objeto vermelho se a quantidade de produtos = 0. Para isso criaremos a seguinte condição:

```
Value = 0
```

Atenção: se você selecionou C++ Script como a linguagem de script (veja maiores detalhes no capítulo Script), você deve escrever a condição usando C++ Script:

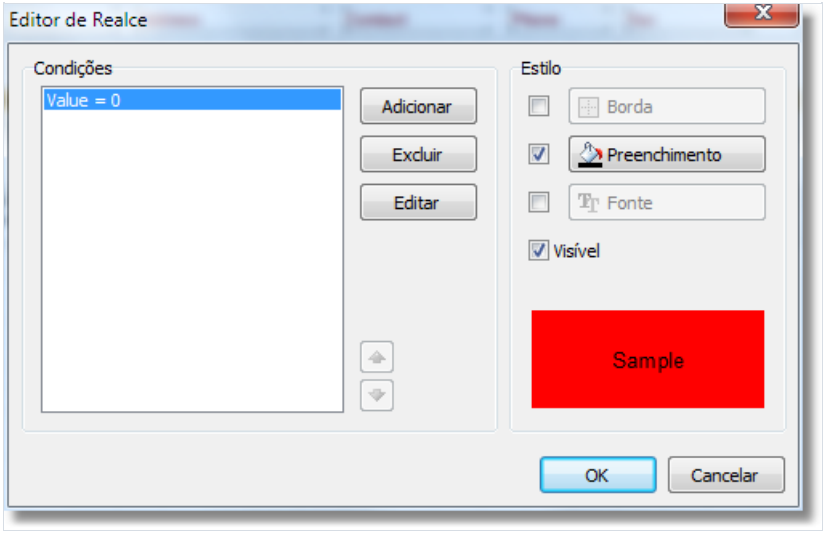
```
Value == 0
```

Neste caso usamos a variável `Value`, que possui um valor impresso. Se houver diversas expressões em um objeto,

esta variável terá o valor da última expressão. Em vez de `Value`, é possível usar uma coluna de dados:

```
<Products."UnitsInStock"> = 0
```

Configure o estilo da condição de maneira que somente preenchimento pode ser usado, e escolha a cor vermelha:



Ao imprimir um objeto com valor zero, ele será vermelho. Vamos tornar nosso exemplo mais complexo, adicionando outra condição. Se as unidades em estoque forem menores que 10, elas devem ser impressas em amarelo. Para fazer isso, abra o editor de condições e clique no botão "Adicionar". A segunda condição será como esta:

```
Value < 10
```

Se diversas condições tiverem sido indicadas, FastReport verifica todas as condições, começando pela primeira. Se uma condição for cumprida, o FastReport aplica suas configurações de estilo ao objeto, e o processo é concluído. É importante colocar as condições na ordem correta. A ordem que vimos neste exemplo está correta:

1. Value = 0
2. Value < 10

Se trocarmos as condições, o realce não funcionará corretamente.

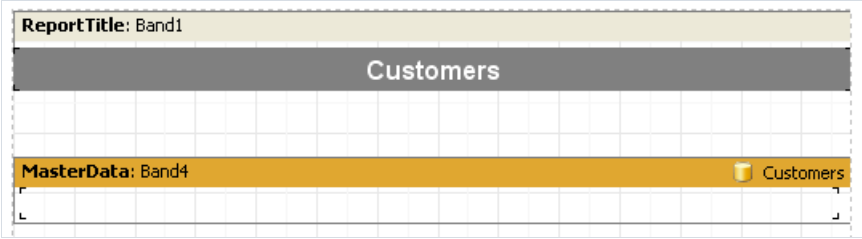
1. Value < 10
2. Value = 0

Neste caso, "Value = 0" não será executado, já que quando o valor for zero, a primeira condição será cumprida. Para alterar a ordem das condições, use os botões e .

Colorindo fileiras de dados alternadas

Ao usar o realce condicional, é fácil criar relatórios que possuam um visual "listrado", nos quais as linhas de dados são coloridas de forma alternada. Para ter menos esforço, vamos usar o exemplo "Lista de Clientes" que criamos anteriormente.

Remova todos os objetos "Texto" da banda "DadosMestre". Coloque um objeto "Texto" na banda de dados e estenda-o para cobrir quase todo o espaço da banda:



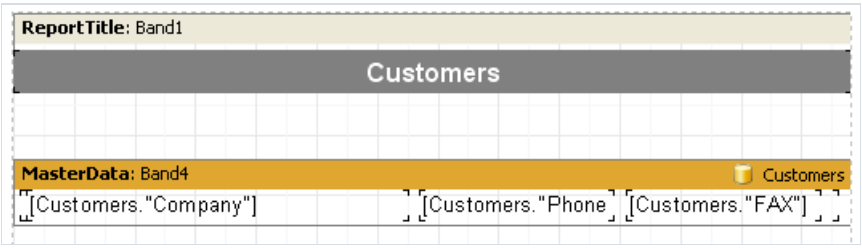
Este objeto irá modificar sua cor em dependência do número da linha de dados. Selecione o objeto e configure a expressão condicional a seguir na guia Realce do editor de objetos:

```
<Line> mod 2 = 1
```

Observação: se você selecionou C++ Script como a linguagem de script (veja maiores detalhes no capítulo Script), você deve escrever a condição usando C++ Script:

```
<Line> % 2 == 1
```

Selecione cinza não muito saturado e claro como a cor de realce. Agora é possível adicionar outros objetos à banda de dados em cima do primeiro objeto "Texto" vazio:



Na visualização o relatório produz esta saída:



Customers		
Action Club	813-870-0239	813-870-0282
Action Diver Supply	22-44-500211	22-44-500596
Adventure Undersea	011-34-09054	011-34-09064
American SCUBA Supply	213-654-0092	213-654-0095
Aquatic Drama	613-442-7654	613-442-7678
Blue Glass Happiness	213-555-1984	213-555-1995
Blue Jack Aqua Center	401-609-7623	401-609-9403
Blue Sports	610-772-6704	610-772-6898

Relatórios aninhados (sub-relatórios)

Às vezes é necessário criar relatórios muito complexos, que contêm blocos de dados adicionais inseridos em pontos específicos no design. Apesar de muitos relatórios deste tipo poderem ser criados usando um arranjo de bandas do FastReport, às vezes isso se torna complicado demais. Nessas circunstâncias é necessário usar o objeto "Sub-relatório".

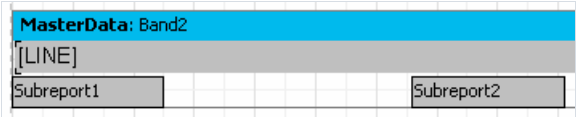
Quando um objeto "Sub-relatório" é inserido em um design no FastReport, ele automaticamente adiciona uma página nova que é conectada ao "Sub-relatório". Este relatório aninhado é parecido a um relatório de páginas múltiplas em termos da estrutura de design. A única diferença é que o relatório aninhado é exibido em um local específico na página de design básica, e não após a página. Quando este relatório é exibido, quando o objeto "Sub-relatório" é encontrado, o mecanismo de relatórios exibe toda a página do sub-relatório conectada. Depois disso, a exibição continua com o resto da página de design básica.

Além disso, objetos "Sub-relatório" podem ser inseridos em uma página de design de sub-relatório, aumentando a profundidade do aninhamento. Um exemplo de um relatório aninhado pode ser encontrado no programa de demonstração, no relatório "Sub-relatório".

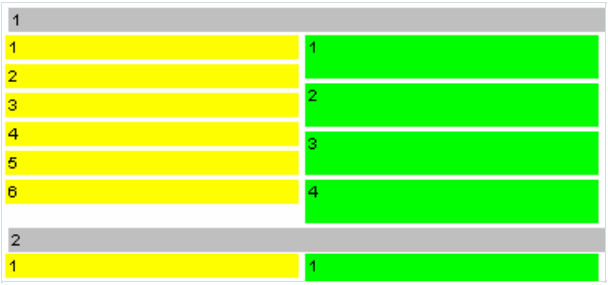
É importante observar que a capacidade de FastReport de usar sub-relatórios permite o aninhamento profundo de dados. Lembre-se de que o número de níveis de dados no FastReport é limitado a somente seis quando bandas de dados Detalhe são usadas em vez do objeto "Sub-relatório".

Sub-relatórios lado a lado

É possível colocar dois ou mais objetos "Sub-relatório" lado a lado na mesma banda de dados:



Este design habilita relatórios nos quais a saída de dados de cada "Sub-relatório" possui fileiras/registros de diversos comprimentos, alturas e graus de extensão:

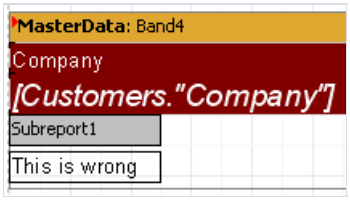


Conforme é ilustrado, FastReport continua exibindo a página de design básica somente após concluir o Sub-relatório mais longo. Também é possível usar a propriedade Vertical Alignment para ajustar o alinhamento do objeto "Texto" em cada sub-relatório.

Limitações na utilização de sub-relatórios

Já que sub-relatórios são colocados na página de design básica, eles não podem conter nenhuma das bandas a seguir: "Título do Relatório/Rodapé do Relatório", "Título da Página/Rodapé da Página/Plano de Fundo" ou "Título da Coluna/Rodapé da Coluna". Se estas bandas forem colocadas em uma página com um relatório aninhado, elas não serão reconhecidas. Por esta mesma razão não há motivo em alterar as opções de páginas de relatórios aninhados, já que as opções da página de relatório básica substituem as opções de páginas aninhadas.

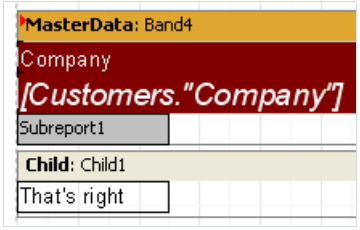
Não coloque objetos abaixo do objeto "Sub-relatório":



Se fizer isso, os objetos criados no sub-relatório irão se sobrepor a tudo que for colocado abaixo do objeto sub-relatório na página de design principal, e a saída será parecida com isto:



Para exibir objetos abaixo ou após um relatório aninhado, utilize uma banda filho:

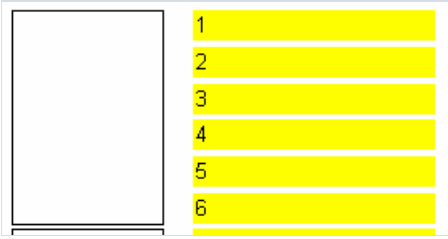


Este método também é usado quando diversos Sub-relatórios devem ser colocados um embaixo do outro. Utilize uma banda filho para cada Sub-relatório e conecte-os em cadeia, configurando a propriedade child de Child1 como Child2, e assim por diante.

Opção PrintOnParent

O objeto "Sub-relatório" possui a propriedade `PrintOnParent` que às vezes pode ser útil. Esta propriedade é configurada como False por padrão.

Normalmente um sub-relatório é exibido como um conjunto de bandas na página de relatório básica. Quando isso acontece, a altura da banda pai que contém o objeto "Sub-relatório" não é controlada pelas bandas no sub-relatório, ou seja, ela não pode ser estendida. Se a propriedade `PrintOnParent` do sub-relatório for configurada como True no inspetor de objetos ou no menu de contexto, os objetos no sub-relatório são impressos físicamente na banda que contém o objeto "Sub-relatório". Esta banda pode ser estendida e pode ter objetos estendidos colocados nela:



Script

Um script é um programa escrito em uma linguagem de alto nível que faz parte do relatório. Conforme o relatório é executado, o script também é executado. Um script pode gerenciar dados de maneiras que não são possíveis usando as operações normais do nucleo do FastReport; por exemplo, um script pode ocultar dados redundantes dependendo de uma condição predefinida. Também é possível utilizar um script para controlar as propriedades de formulários de diálogo que fazem parte de um relatório.

Um script é escrito em uma das linguagens suportadas pelo mecanismo de scripts (FastScript). Estas são as linguagens que são suportadas atualmente:

- PascalScript
- C++Script
- BasicScript
- JScript

Os recursos a seguir são suportados pelo mecanismo FastScript:

- conjunto padrão da linguagem: variáveis, constantes, procedimentos, funções (que podem ser aninhadas e podem possuir variáveis, constantes, parâmetros padrão), todos os operadores padrão (incluindo case, try, finally, except, with), tipos (integral, fracional, lógico, caractere, linha, arrays multidimensionais, variante), classes (com métodos, eventos, propriedades, índices e propriedades padrão)
- verificação de compatibilidade de tipos
- acesso a todos os objetos do relatório

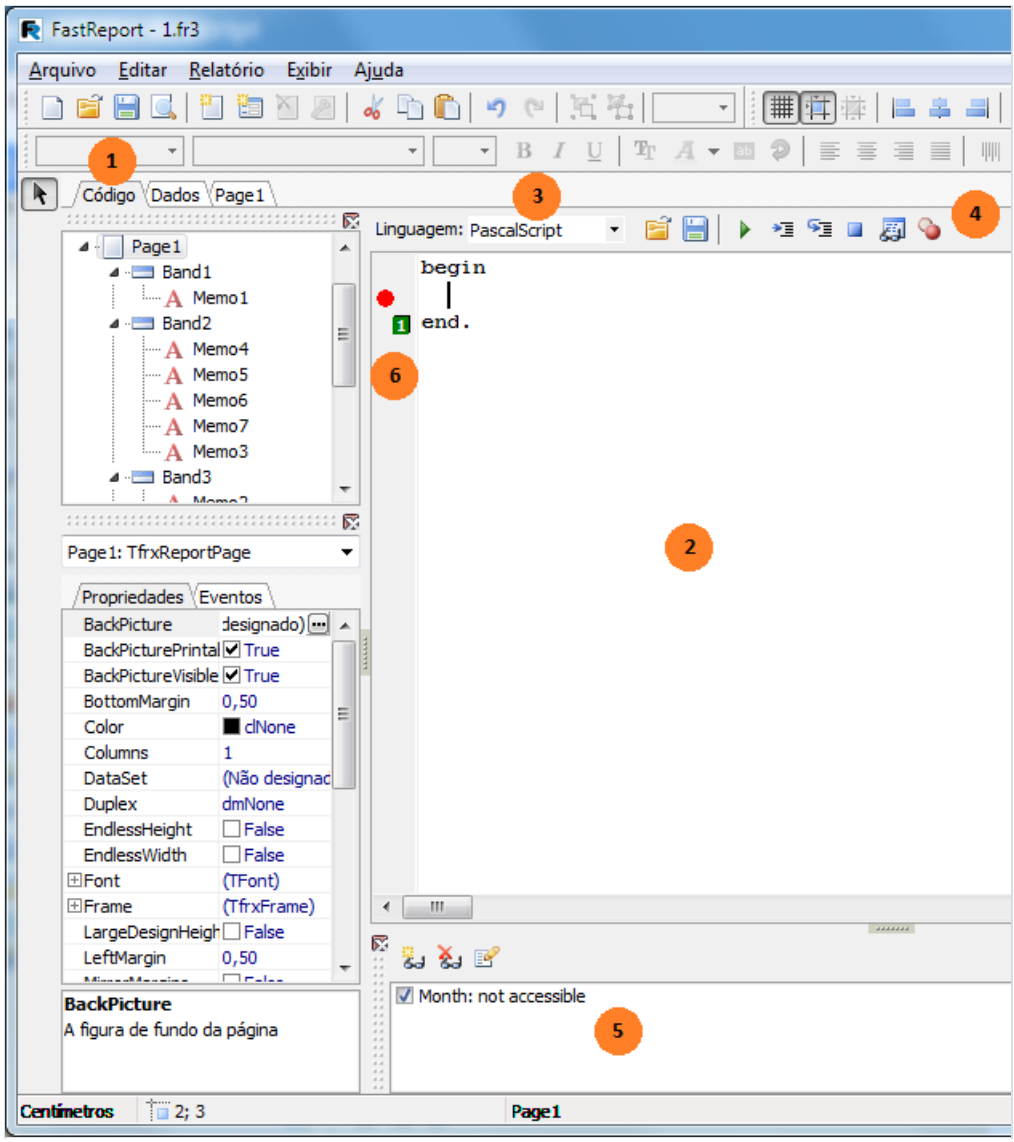
No entanto, o FastScript não suporta o seguinte:

- declarações deste tipo: registros, classes
- ponteiros, conjuntos (porém é possível usar o operador 'IN' em expressões como "a in ['a'..'c','d']")
- tipo shortstring
- saltos incondicionais (GOTO)

Os scripts podem ser criados no designer do FastReport, que contém um editor de scripts com realce de sintaxe. Também há um depurador embutido que possui as seguintes funções: "Step", "Breakpoint", "Run to cursor" e "Evaluate".

Um gostinho de script

As ferramentas para trabalhar com os scripts estão localizadas na guia "Código" do designer do FastReport. Ao abrir esta guia, o IDE tem este aspecto:



Legenda dos rótulos acima:

1 – guia "Código"

2 – painel do editor de scripts

3 – lista suspensa para selecionar a linguagem na qual é escrito o script

4 – barra de ferramentas do depurador

- executar o relatório no modo de depuração (F9)

- executar até o cursor (F4)

- executar a linha de código regular (Seguir Dentro, F7)

- interromper a execução de um script (Ctrl+F2)

- visualizar o valor de uma expressão (Avaliar, Ctrl+F7)

 - alternar o ponto de parada (F5)

5 – painel "Relógios"

6 – os indicadores e pontos de parada são exibidos nesta coluna; a linha do código em execução também é realçada aqui

A lista de teclas de atalho que podem ser usadas no editor de scripts:

Tecla	Significado
teclas de direção	movem o cursor
PageUp, PageDown	ir para a página anterior/próxima
Ctrl+PageUp	ir para o início do texto
Ctrl+PageDown	ir para o final do texto
Home	ir para o início da linha
End	ir para o fim da linha
Enter	ir para a próxima linha
Delete	excluir símbolo na posição do cursor; excluir texto selecionado
Backspace	excluir o símbolo à esquerda do cursor
Ctrl+Y	excluir a linha atual
Ctrl+Z	desfazer a última ação (até 32 eventos)
Shift+teclas de direção	selecionar um bloco de texto
Ctrl+A	selecionar todo o texto
Ctrl+U	deslocar o bloco selecionado 2 símbolos para a esquerda
Ctrl+I	deslocar o bloco selecionado 2 símbolos para a direita
Ctrl+C, Ctrl+Insert	copiar o bloco selecionado para a área de transferência
Ctrl+V, Shift+Insert	colar texto da área de transferência
Ctrl+X, Shift+Delete	cortar o bloco selecionado para a área de transferência
Ctrl+Shift+<número>	definir um indicar com <número> 0..9 na linha atual
Ctrl+<número>	saltar para o indicador <número>
Ctrl+F	pesquisar uma linha

Tecla	Significado
Ctrl+R	substituir uma linha
F3	pesquisa/substituição repetida a partir da posição do cursor
F4	definir ponto de parada para a execução do script (Executar até o cursor)
Ctrl+F2	redefinir o script
Ctrl+F7	visualizar o valor da variável (Avaliar)
F9	executa o script (Executar)
F7 or F8	executar a linha de código (Seguir dentro)
Ctrl + Espaço	exibir lista de métodos e propriedades do objeto
Ctrl+Shift+Delete	excluir a palavra à direita do cursor
Ctrl+Shift+Backspace	excluir a palavra à esquerda do cursor

Estrutura de um script

A estrutura de um script depende da linguagem utilizada. No entanto, há alguns elementos que as linguagens têm em comum: o título e o corpo do script, e o procedimento principal que é executado quando o relatório é executado. Abaixo há exemplos de scripts nas quatro linguagens suportadas:

Estrutura de PascalScript:

```
#language PascalScript // opcional
program MyProgram;     // opcional
// o capítulo "uses" deve estar localizado antes de qualquer outro capítulo
uses 'unit1.pas', 'unit2.pas';
var                    // o capítulo "variables" pode ser colocado em qualquer lugar
  i, j: Integer;
const                  // capítulo "constants"
  pi = 3.14159;
procedure p1;          // procedimentos e funções
var
  i: Integer;
  procedure p2;        // procedimento aninhado
  begin
  end;
begin
end;
begin                  // procedimento principal.
end.
```

Estrutura de C++Script:

```
#language C++Script    // opcional
// o capítulo "include" deve ser colocado antes de qualquer outro capítulo
#include "unit1.cpp", "unit2.cpp"
int i, j = 0;          // o capítulo "variables" pode ser colocado em qualquer lugar
#DEFINE pi = 3.14159   // capítulo "constants"
void p1()              // funções
{                      // sem procedimentos aninhados
}
{                      // procedimento principal.
}
```

Estrutura de JScript:

```
#language JScript      // opcional
// o capítulo "import" deve ser colocado antes de qualquer outro capítulo
import "unit1.js", "unit2.js"
var i, j = 0;          // o capítulo "variables" pode ser colocado em qualquer lugar
function p1()          // funções
{                      //
}
                       // procedimento principal.
p1();
for (i = 0; i < 10; i++) j++;
```

Estrutura de BasicScript:

```
#language BasicScript  ' opcional
' o capítulo "imports" deve ser colocado antes de qualquer outro capítulo
imports "unit1.vb", "unit2.vb"
Dim i, j = 0           ' o capítulo "variables" pode ser colocado em qualquer lugar
Function p1()          ' funções
{                      '
}
                       ' procedimento principal.
For i = 0 To 10
  p1()
Next
```

Mais adiante iremos dar uma olhada em scripts escritos na linguagem "PascalScript". Quando um relatório novo é criado, esta linguagem é selecionada por padrão.

Script "Olá, Mundo!"

Já vimos um exemplo de um relatório "Olá, Mundo!". Agora vamos ver como criar um script simples que exibe uma janela com a mesma saudação.

Abra o designer e clique no botão "Novo Relatório" para que o FastReport crie automaticamente um modelo básico. Abra a guia "Código" e escreve o script a seguir:

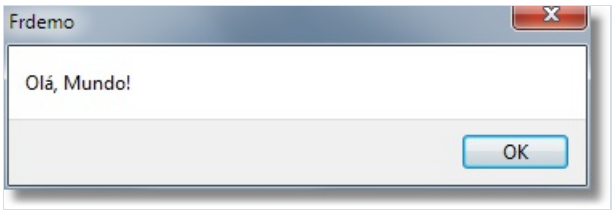
PascalScript:

```
begin
  ShowMessage('Olá, Mundo!');
end.
```

C++ Script:

```
{
  ShowMessage("Olá, Mundo!");
}
```

Depois disso execute o relatório. Conforme esperado, o FastReport exibe uma pequena caixa de diálogo com a saudação:



Vamos explicar alguns detalhes. Criamos um script que é composto de um único bloco "begin...end". Desta forma, nosso script possui uma estrutura muito simples; ele é composto apenas de um procedimento principal (consulte a seção "Estrutura de um script" acima). O procedimento principal é executado assim que o relatório é executado. Neste caso, ele exibiu uma caixa de diálogo de saudação, e o procedimento é concluído após fechar a caixa de diálogo. Depois do término da execução do procedimento principal, a criação normal do relatório é iniciada.

Usando objetos no script

Qualquer objeto do relatório pode ser acessado por um script. Deste modo, se houver uma página "Page1" e um objeto "Memo1", por exemplo, eles podem ser usados no script ao chamá-los usando seus nomes:

PascalScript:

```
Memo1.Color := clRed
```

C++Script:

```
Memo1.Color = clRed
```

A lista de objetos do relatório que podem ser acessados pelo script é exibida no painel "Árvore do Relatório". Quais propriedades do objeto estão disponíveis para um script? A resposta é simples: todas que estão visíveis no inspetor de objetos. O inspetor de objetos também mostra dicas de cada propriedade na parte inferior. Ambos os painéis (a árvore do relatório e o inspetor de objetos) permanecem disponíveis ao trabalhar em um script. Para obter ajuda detalhada sobre as propriedades do objeto e os métodos, consulte o arquivo de ajuda do FastReport que é incluído no kit de distribuição.

Aqui há um exemplo simples. Coloque um objeto "Texto" com o nome "MyTextObject" que contém "Test" na página de design do relatório. Então escreva este script:

PascalScript:

```
begin
  MyTextObject.Color := clRed
end.
```

C++Script:

```
{
  MyTextObject.Color = clRed
}
```

Execute o relatório e veja que a cor do objeto é vermelha.

Chamando as variáveis da lista de variáveis do relatório

Toda variável que é definida na lista de variáveis do relatório (item de menu "Relatório > Variáveis...") pode ser referenciada em um script. O nome da variável deve estar entre chevrons:

PascalScript:

```
if <minha variável> = 10 then ...
```

C++ Script:

```
if (<minha variável> == 10) { ... }
```

Uma maneira alternativa é utilizar a função `Get`:

PascalScript:

```
if Get('minha variável') = 10 then ...
```

C++ Script:

```
if (Get("minha variável") == 10) { ... }
```

O valor de uma variável é alterado apenas através do procedimento `Set`:

PascalScript:

```
Set('minha variável', 10);
```

C++ Script:

```
Set("minha variável", 10);
```

Vale a pena observar que para atribuir um valor de string à variável, você deve colocar o valor entre aspas:

PascalScript:

```
Set('minha variável', '''' + 'String' + '''');
```

C++ Script:

```
Set("minha variável", "\"String\"");
```

Variáveis de sistema, como `Page#`, devem ser referenciadas exatamente da mesma maneira:

PascalScript:

```
if <Page#> = 1 then ...
```

C++ Script:

```
if (<Page#> == 1) { ... }
```

Referenciando os campos de BD

Assim como nas variáveis, é necessário usar chevrons ao referenciar campos de BD em um relatório:

PascalScript:

```
if <Table1."Field1"> = Null then...
```

C++ Script:

```
if (<Table1."Field1"> == Null) { ... }
```

De forma alternativa, é possível usar a função `Get` para acessar campos de BD (na verdade esta função é usada implicitamente por FastReport ao calcular expressões dentro de chevrons).

Utilizar funções de agregação no script

É uma peculiaridade das funções de agregação que elas devem ser usadas dentro de objetos "Texto"; uma vez usadas desta maneira, elas podem ser usadas no script em si. Se uma função de agregação aparecer apenas em um script (sem aparecer em um objeto "Texto"), é gerada uma mensagem de texto. Isso acontece porque uma função de agregação deve estar conectada a uma banda específica. Se ela estiver conectada, ela irá funcionar corretamente.

Exibir o valor de uma variável em um relatório

Variáveis podem ser declaradas e usadas localmente dentro de um script. Uma vez declarada, uma variável de script pode ter um valor atribuído a ela. Aqui segue um exemplo simples de uma variável de script sendo utilizada:

PascalScript:

```
var
  MyVariable: String;
begin
  MyVariable := 'Olá!';
end.
```

C++ Script:

```
string MyVariable;
{
  MyVariable = "Olá!";
}
```

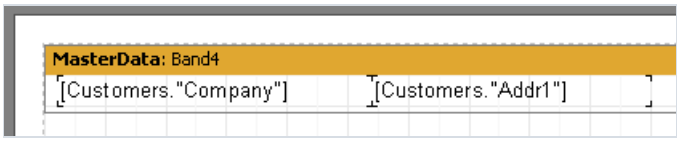
O valor da variável pode ser exibido em um objeto "Texto", por exemplo, ao digitar '[MyVariable]' no objeto.

O nome de uma variável deve ser único. Isso significa que o nome não deve duplicar o nome de outro objeto do relatório, de uma função padrão ou constante. Se houver um erro em um script, é exibida uma mensagem quando o relatório for executado e a criação do relatório é parada.

Eventos

Até agora nos vimos scripts com apenas um procedimento principal que é executado quando o relatório é executado. No procedimento principal é possível efetuar configurações iniciais e inicializar variáveis. No entanto, isso não é suficiente para obter o controle total sobre o processo da geração de relatórios. Para obter o máximo de controle possível sobre a geração de relatórios, cada objeto possui diversos eventos aos quais é possível atribuir manipuladores (ou seja, procedimentos no script). Por exemplo, conectar um manipulador à banda de dados permite a filtragem de registros, de modo que a banda pode ser ocultada ou revelada de acordo com condições específicas.

Vamos demonstrar o processo da criação do relatório e os eventos disparados através de um relatório simples que contém uma página e possui uma banda "Dados Mestre", com dois objetos "Texto" na banda:



Conforme é descrito acima, o procedimento principal do script é chamado no início da execução do relatório. Depois disso começam as partes essenciais da criação do relatório. Primeiro é chamado o evento "OnStartReport" do objeto "Relatório". Então, antes de criar uma página de saída, o evento "OnBeforePrint" é chamado. Este evento é chamado uma vez para cada página de design no modelo do relatório (páginas de design não devem ser confundidas com as páginas de saída de um relatório!). Em nosso exemplo, o evento é chamado uma vez, já que o design do relatório é composto de apenas uma página de design.

Então os eventos das bandas de dados são chamados na seguinte ordem:

1. o evento "OnBeforePrint" da banda é chamado
2. o evento "OnBeforePrint" de cada objeto contido na banda é chamado
3. cada objeto é preenchido com dados (em nosso exemplo, com valore dos campos "Company" e "Addr1")
4. o evento "OnAfterData" de cada objeto é chamado
5. ações como posicionar objetos na banda (se houver objetos extensíveis entre eles), calcular a altura da banda e estendê-la (se ela for extensível) são executadas
6. o evento "OnAfterCalcHeight" da banda é chamado
7. é criada uma nova página de saída se a banda não possuir espaço suficiente no espaço em branco da página
8. a banda e todos os seus objetos são exibidos na página de saída
9. o evento "OnAfterPrint" de cada banda é chamado
10. o evento "OnAfterPrint" da banda em si é chamado

As bandas continuam a ser impressas enquanto a fonte conectada à banda tiver dados. Depois que a impressão do relatório é concluída, é chamado o evento "OnAfterPrint" da página do relatório, e finalmente o evento "OnStopReport" do objeto "Relatório".

Dessa maneira, ao usar os eventos de objetos diferentes, praticamente cada etapa do processo de criação de um relatório pode ser gerenciado. O mais importante ao usar eventos é um entendimento profundo do processo de saída de bandas, que é discutido nas próximas nove seções. A maior parte das ações pode ser executada usando apenas o evento "OnBeforePrint" da banda; qualquer modificação efetuada em um objeto é exibida imediatamente.

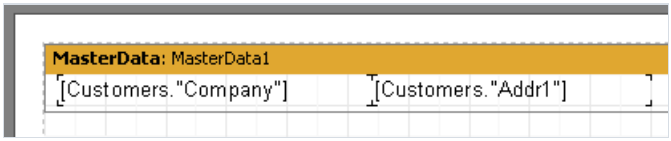
No entano, se a banda for extensível, é impossível afirmar em que página será impressa a banda neste evento, já que o cálculo da altura da banda é executada na etapa 5. No entanto, isso pode ser feito no evento "OnAfterCalcHeight" na etapa 6 ou no evento "OnAfterPrint" na etapa 9. Observe que no último evento a banda já vai ter sido exibida, então a modificação de objetos não terá nenhum efeito visível.

É essencial entender claramente "onde e quando" as bandas são exibidas, e entender o timing (ordem de chamada) de cada um de seus eventos. Isso também é válido para cada um dos objetos contidos nas bandas.

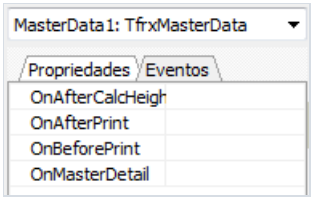
Exemplo de uso do evento "OnBeforePrint"

Para demonstrar esse evento, crie um relatório que apresenta uma lista de clientes. Este relatório irá incluir apenas as empresas cujo nome comece com "A".

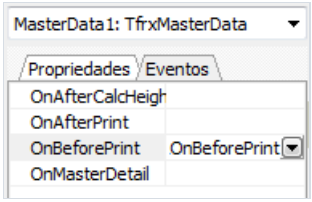
Abra o designer de relatórios e crie um relatório como este:



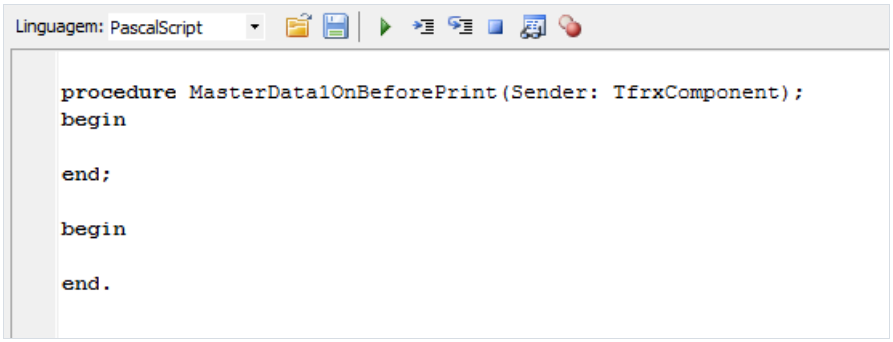
Selecione a banda de dados e abra a guia "Eventos" no inspetor de objetos:



Para criar um manipulador de eventos "OnBeforePrint" (que é o mais apropriado para nós), clique duas vezes no campo em branco à direita do nome do evento:



Isso adiciona um manipulador em branco ao script e o designer abre a guia "Código".



Tudo que precisamos fazer agora é digitar o seguinte cógido no corpo do manipulador:

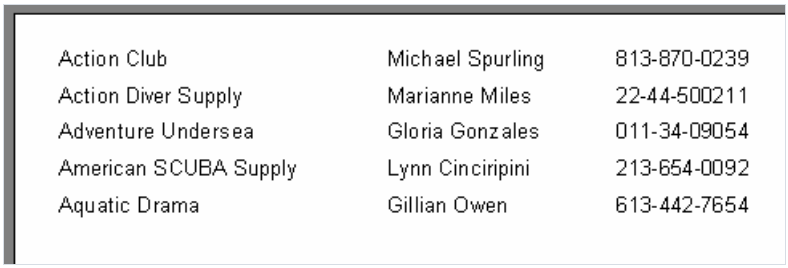
PascalScript:

```
if Copy(<Customers."Company">, 1, 1) = 'A' then
  MasterData1.Visible := True
else
  MasterData1.Visible := False;
```

C++Script:

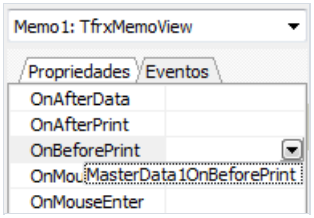
```
if (Copy(<Customers."Company">, 1, 1) == "A")
  MasterData1.Visible = true;
else
  MasterData1.Visible = false;
```

Execute o relatório e certifique-se de que o script está funcionando corretamente:



Action Club	Michael Spurling	813-870-0239
Action Diver Supply	Marianne Miles	22-44-500211
Adventure Undersea	Gloria Gonzales	011-34-09054
American SCUBA Supply	Lynn Cinciripini	213-654-0092
Aquatic Drama	Gillian Owen	613-442-7654

Vamos explicar diversas coisas. Um manipulador pode ser atribuído aos eventos de mais de um objeto – o parâmetro `Sender` exibe qual objeto iniciou o evento. Para atribuir um manipulador existente ao um evento, digite-o diretamente no inspetor de objetos ou selecione-o na lista suspensa:



É fácil excluir o link para um manipulador - selecione o manipulador atribuído no inspetor de objetos e aperte a tecla "Delete".

Imprimir um total de grupo no cabeçalho do grupo

Este método é usado com frequência e requer o uso de um script, já que o valor de um total de grupo somente é conhecido após processar todos os registros no grupo. Para exibir o total no cabeçalho do grupo (ou seja, antes que o grupo seja exibido no relatório) é utilizado o seguinte algoritmo:

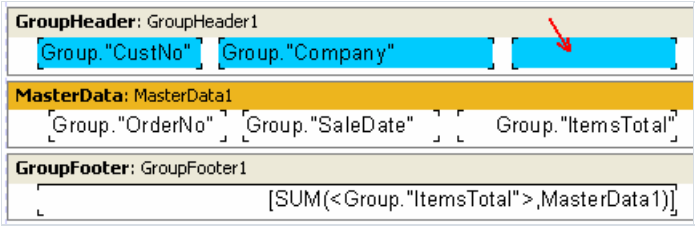
- ative a opção de relatório dois passos (item de menu "Relatório > Opções...")
- no primeiro passo, calcule o total de cada grupo e salve-os em um array
- no segundo passo, extraia os valores do array e exiba-os nos cabeçalhos dos grupos

Vamos demonstrar dois métodos de executar esta tarefa. Crie um relatório novo. Coloque o componente "Consulta ADO" na página e configura sua propriedade SQL:

```
SQL =
  select * from customer, orders
  where orders.CustNo = customer.CustNo
  order by customer.CustNo, orders.OrderNo

UserName = 'Group'
```

Habilite dois passos nas configurações do relatório (item de menu "Relatório > Opções..."). Adicione duas bandas ao relatório: "Cabeçalho de Grupo" e "Dados Mestre". No editor da banda "Cabeçalho de Grupo", insira o campo de dados Group."CustNo". Conecte a banda de dados à fonte de dados "Group" e arranje alguns objetos da seguinte maneira:



Para exibir o total, usamos o objeto com a seta no design (em nosso exemplo ele é chamado de "Memo8").

O primeiro método.

Usaremos a classe `TStringList` como um array para armazenar os totais – vamos armazenar os valores numéricos como cadeias. O primeiro item na StringList corresponderá ao total do primeiro grupo, etc. Uma variável de número inteiro (que iremos incrementar após imprimir cada grupo) é usada para calcular o número de índice do grupo.

Então nosso script terá o seguinte aspecto:

PascalScript:

```
var
  List: TStringList;
  i: Integer;

procedure frReport1OnStartReport(Sender: TfrxComponent);
begin
  List := TStringList.Create;
end;

procedure frReport1OnStopReport(Sender: TfrxComponent);
begin
  List.Free;
end;

procedure Page1OnBeforePrint(Sender: TfrxComponent);
begin
  i := 0;
end;

procedure GroupHeader1OnBeforePrint(Sender: TfrxComponent);
begin
  if Engine.FinalPass then
    Memo8.Text := 'Sum: ' + List[i];
end;

procedure GroupFooter1OnBeforePrint(Sender: TfrxComponent);
begin
  if not Engine.FinalPass then
    List.Add(FloatToStr(SUM(<Group."ItemsTotal">,MasterData1)));
  Inc(i);
end;

begin
end.
```

C++ Script:

```
TStringList List;
int i;

void frReport1OnStartReport(TfrxComponent Sender)
{
  List = TStringList.Create();
}

void frReport1OnStopReport(TfrxComponent Sender)
{
  List.Free();
}

void Page1OnBeforePrint(TfrxComponent Sender)
{
  i = 0;
}

void GroupHeader1OnBeforePrint(TfrxComponent Sender)
{
  if (Engine.FinalPass)
    Memo8.Text = "Sum: " + List[i];
}

void GroupFooter1OnBeforePrint(TfrxComponent Sender)
{
  List.Add(FloatToStr(SUM(<Group."ItemsTotal">,MasterData1)));
  i++;
}

{
}
```

Os nomes dos procedimentos no script mostram quais eventos utilizamos. Eles são: "Report.OnStartReport", "Report.OnStopReport", "Page1.OnBeforePrint", "GroupHeader1.OnBeforePrint" e "GroupFooter1.OnBeforePrint". Os primeiros dois eventos são chamados no início e no final do relatório, respectivamente. Para criar manipuladores para estes dois eventos, selecione o objeto "Report" na lista suspensa do inspetor de objetos e suas propriedades serão exibidas no inspetor de objetos. Então abra a guia "Eventos" do inspetor de objetos e crie os manipuladores.

Por que não criamos a variável `List` no procedimento principal do script? Nos a criamos no evento "OnStartReport" porque variáveis criadas dinamicamente devem ser destruídas após concluir o relatório. É lógico criar variáveis dinâmicas no evento "OnStarReport" e destruí-las no evento "OnStopReport". Em outros casos (quando a memória não precisa ser liberada na conclusão do script) é possível usar o procedimento principal do script para a inicialização de variáveis.

A criação e destruição da variável `List` é simples. Agora vamos ver como o script funciona. No início da página, o contador do grupo atual (a variável "i") é redefinida como zero e é incrementada após a impressão de cada grupo (no evento "GroupFooter1.OnBeforePrint"). O total calculado é adicionado a `List` neste evento antes de incrementar o contador. O evento "GroupHeader1.OnBeforePrint" não faz nada no primeiro passo (condição if "Engine.FinalPass"), mas durante o segundo passo (quando `List` foi preenchida com valores) o total que corresponde ao grupo atual é recuperado de `List` e a saída é enviada para o objeto "Memo8" para exibir o total no cabeçalho do grupo. No relatório concluído, o total é exibido desta maneira:

1221	Kauai Dive Shoppe	Sum: 51450,8
1023	01.07.88	4 674,00
1076	16.12.94	17 781,00
1123	24.08.93	13 945,00
1169	06.07.94	9 471,95
1176	26.07.94	4 178,85
1269	16.12.94	1 400,00
		51 450,80

Este algoritmo é bastante simples. No entanto, ele pode ser simplificado ainda mais.

O segundo método.

Vamos usar a coleção de variáveis do relatório como um array para armazenar os totais dos grupos. Lembre-se de que as variáveis do relatório são acessadas através das funções `Get` e `Set`. Usar estas funções também nos poupa de ter que criar e destruir estas variáveis de maneira explícita. Este será nosso script:

PascalScript:

```
procedure GroupHeader1OnBeforePrint(Sender: TfrxComponent);
begin
  if Engine.FinalPass then
    Memo8.Text := 'Sum: ' + Get(<Group."CustNo">);
end;

procedure GroupFooter1OnBeforePrint(Sender: TfrxComponent);
begin
  Set(<Group."CustNo">,
    FloatToStr(SUM(<Group."ItemsTotal">,MasterData1)));
end;

begin
end.
```

C++ Script:

```
void GroupHeader1OnBeforePrint(TfrxComponent Sender)
{
  if (Engine.FinalPass)
    Memo8.Text = "Sum:" + Get(<Group."CustNo">);
}

void GroupFooter1OnBeforePrint(TfrxComponent Sender)
{
  Set(<Group."CustNo">,
    FloatToStr(SUM(<Group."ItemsTotal">,MasterData1)));
}

{
}
```

Como você pode ver, este script é um pouco mais simples. O código no manipulador "GroupFooter1.OnBeforePrint" define o valor de uma variável que possui um nome derivado do número do cliente (também é possível usar qualquer outro identificador que identifique o cliente de maneira não ambígua, por exemplo <Group."Company">). Se não existir nenhuma variável com este nome, o script a cria automaticamente; de outro modo, se ela existir, seu valor é atualizado. O valor da variável apropriada é recuperado no manipulador "GroupHeader1.OnBeforePrint".

Evento "OnAfterData"

Este evento é disparado depois que um objeto do relatório tenha sido preenchido com os dados da fonte à qual está conectado. Use este evento para acessar o valor de um campo de BD ou uma expressão contida no objeto. Este valor é colocado na variável de sistema `Value`, que é disponibilizada somente neste evento. Então, se dois objetos "Textos" possuirem as expressões [Table1."Field1"] e [<Table2."Field1"> + 10], os valores destas expressões podem ser utilizados ao se referir à variável `Value` dos objetos:

PascalScript:

```
if Value > 3000 then
  Memo1.Color := clRed
```

C++ Script:

```
if (Value > 3000)
  Memo1.Color = clRed;
```

o que é mais simples do que escrever algo parecido com isto:

PascalScript:

```
if <Table1."Field1"> > 3000 then
  Memo1.Color := clRed
```

C++ Script:

```
if (<Table1."Field1"> > 3000)
  Memo1.Color = clRed;
```

Usar `Value` em vez de uma expressão habilita você a escrever um manipulador multiuso para o evento "OnAfterData", que pode ser conectado a diversos objetos.

Observe também outra coisa: se um objeto possuir diversas expressões (por exemplo '[expr1] [expr2]'), o valor da última expressão é transferido à variável `Value`.

O evento "OnAfterData" é ideal para calcular a altura e largura de objetos como "Texto". Isto é, se em um script for necessário obter a altura exata de um objeto "Texto" estendido que contém uma expressão, é possível usar este código no evento "OnAfterData":

PascalScript:

```
var
  MemoHeight: Extended;
begin
  MemoHeight := TfrxMemoView(Sender).CalcHeight;
end;
```

C++ Script:

```
  float MemoHeight;
  MemoHeight = TfrxMemoView(Sender).CalcHeight;
```

Se este código fosse usado no evento "OnBeforePrint", o resultado seria a altura do objeto que contém a expressão antes da expressão ser avaliada, e não seu valor real na impressão.

Objetos de serviço

Há outros objetos de sistema que podem ser utilizados em script, junto aos objetos de relatório normais como páginas, bandas, "Texto" e outros. Eles podem ser úteis no gerenciamento da construção do relatório. O objeto `Engine`, usado no capítulo anterior, é um objeto deste tipo, e todos são listados aqui:

- `Report` : o objeto "Relatório"
- `Engine` : o link ao "Mecanismo" do relatório
- `Outline` : o link ao elemento "Árvore do Rélatório" em um relatório visualizado

Vamos dar uma olhada em cada um destes objetos.

Objeto "Report"

Este objeto representa um link para o relatório atual. As propriedades deste objeto podem ser vistas ao selecionar o elemento "Report" na janela "Árvore do Relatório".

Métodos:

Método	Descrição
function Calc(const Expr: String): Variant	retorna o valor de "Expr", que é uma expressãopor exemplo, Report.Calc('1+2') retorna "3"qualquer expressão válida do FastReport pode ser passada no parâmetro
function GetDataSet(const Alias: String): TfrxDataSet	retorna o conjunto de dados que possuir o nome especificadoo conjunto de dados deve ser exibido na lista de conjuntos de dados do relatório (caixa de diálogo "Relatório > Dados...")

Objeto "Engine"

Os métodos e as propriedades deste objeto.

Propiedade	Tipo	Descrição
CurColumn	Integer	o índice da coluna atual em um relatório com diversas colunasé possível atribuir um valor a esta propriedade.
CurX	Extended	a posição de impressão atual no eixo Xé possível atribuir um valor a esta propriedade.
CurY	Extended	a posição de impressão atual no eixo Y é possível atribuir um valor a esta propriedade.
DoublePass	Boolean	é igual a "True" se o relatório for de dois passosé análoga a Report.EngineOptions.DoublePass.
FinalPass	Boolean	é igual a "True" quando estiver no último passo de um relatório de dois passos
PageHeight	Extended	altura de impressão da área, em pixels
PageWidth	Extended	largura de impressão da área, em pixels
StartDate	TDateTime	tempo de execução do relatorioigual à variável de sistema
StartTime	TDateTime	tempo de execução do relatorioigual à variável de sistema
TotalPages	Integer	o número de páginas em um relatórioigual à variável de sistema o relatório deve ser de dois passos se esta variável for usada
SecondScriptcall	Boolean	sinalizador que retorna o status 'repeat-call' de um evento (em alguns casos um evento pode ser chamado repetidamente durante o agrupamento)se for igual a True, o script já foi chamado

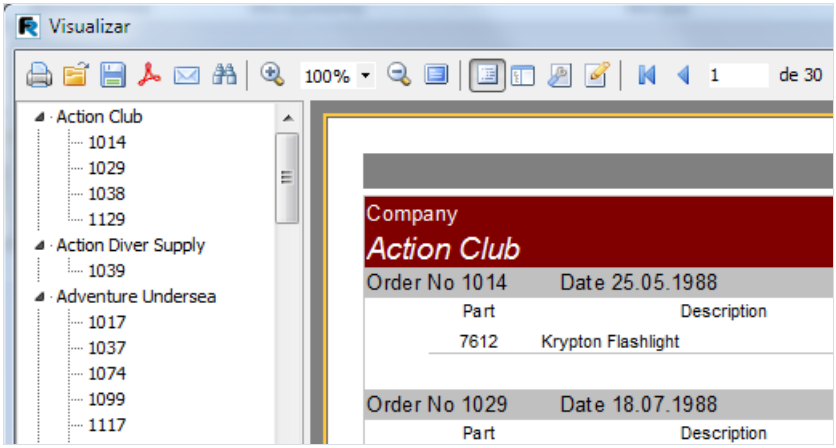
Métodos:

Método	Descrição
procedure AddAnchor(const Text: String)	adiciona "Text" à lista de âncorasveja mais abaixo
procedure NewColumn	cria uma nova coluna em um relatório com várias colunasapós a última coluna, uma quebra de página é inserida automaticamente
procedure NewPage	cria uma página nova (quebra de página)
procedure ShowBand(Band: TfrxBand)	exibe uma banda com o nome especificadoapós exibir a banda, a posição "CurY" é incrementada automaticamente
function FreeSpace: Extended	retorna a altura do espaço em branco que sobra na página em pixels.

Método	Descrição
function GetAnchorPage(const Text: String): Integer	retorna o número da página onde foi colocada a âncora especificada

Objeto "Outline"

Este objeto representa o elemento de controle "Árvore do Relatório" em um relatório visualizado.



"Outline" exibe uma estrutura tipo árvore do relatório concluído. Quando um nó da árvore é clicado, o painel de visualização abre a página referida no nó. Para exibir "Outline", ele deve ser habilitado ao clicar no botão na barra de ferramentas da janela de visualização ou ao configurar a propriedade "Report.PreviewOptions.OutlineVisible" como True. A largura de "Outline" em pixels também pode ser configurada ali: "Report.PreviewOptions.OutlineWidth".

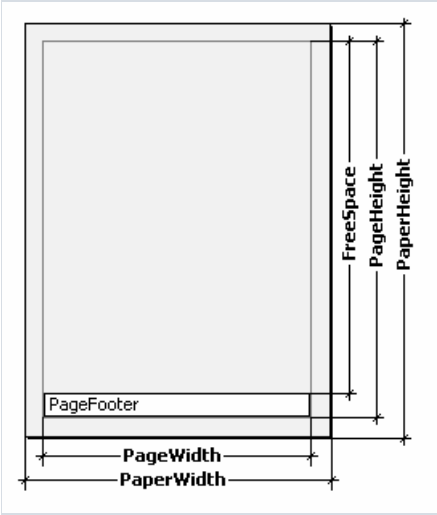
Estes são os métodos de "Outline":

Método	Descrição
procedure AddItem(const Text: String)	adiciona um elemento que possui o nome "Text" na posição atual da árvorea página atual do relatório e a posição na página são vinculados ao elemento
procedure LevelRoot	move a posição atual na árvore para o nível raiz
procedure LevelUp	eleva a posição atual na árvore por um nível

Utilização do objeto "Engine"

Já dissemos que o objeto `Engine` representa o mecanismo do relatório que gerencia a construção do relatório. O processo de arranjar as bandas na página pode ser gerenciado ao usar as propriedades e métodos de `Engine`. Primeiro vamos ver um pouco de teoria.

O diagrama abaixo mostra diversas dimensões da página do relatório.



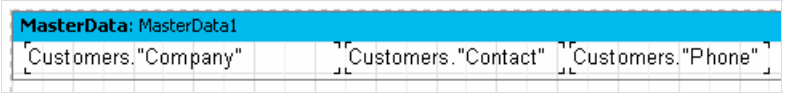
As dimensões físicas da página são as propriedades `PaperWidth` e `PaperHeight`, que se tornam visíveis no inspetor de objetos quando a página for selecionada. Então o tamanho de uma página A4 é 210 x 297 mm.

`PageWidth` e `PageHeight` são as dimensões da área de impressão, que normalmente é menor que as dimensões físicas da página. O tamanho da área de impressão depende das própriedades da página do relatório `LeftMargin`, `TopMargin`, `RightMargin` e `BottomMargin`. O tamanho da área de impressão em pixels é retornado pelas funções `Engine.PageWidth` e `Engine.PageHeight`.

Finalmente, `FreeSpace` é a altura do espaço livre em uma página. Se houver uma banda "Rodapé da Página" na página, sua altura é incluída no cálculo de `FreeSpace`. A altura é retornada em pixels na função `Engine.FreeSpace`. Observe que após exibir a próxima banda, o espaço livre é reduzido na página, e isto é levado em conta ao calcular `FreeSpace`.

Como são construídas as páginas do relatório? O núcleo do FastReport exibe bandas em uma página enquanto houver espaço livre suficiente. Quando não sobrar mais espaço livre, a banda "Rodapé da Página" é impressa (se necessário) e uma nova página em branco é criada. Como já foi dito, após exibir a próxima banda, a altura do espaço livre é reduzida. Além disso, a exibição da próxima banda começa na posição atual, que é definida pelas coordenadas no eixo X e Y. A posição atual é retornada por `Engine.CurX` e `Engine.CurY`, respectivamente. Após imprimir a próxima banda, CurY é incrementada com a altura da banda impressa. Após a criação de uma nova página, "CurY" é igual a "0". "CurX" é alterada ao imprimir relatórios com diversas colunas.

`Engine.CurX` e `Engine.CurY` estão disponíveis não somente para a leitura, mas também para a gravação. Isso significa que as bandas podem ser deslocadas ao incrementar ou reduzir estes valores. Por exemplo, em um relatório parecido com este:



ele pode ser impresso desta maneira:



Action Club	Michael Spurling	813-870-0239
Action Diver Supply	Marianne Miles	22-44-500211
Adventure Undersea	Gloria Gonzales	011-34-09054
American SCUBA Supply	Lynn Cinciripini	213-654-0092
Aquatic Drama	Gillian Owen	613-442-7654
Blue Glass Happiness	Christine Taylor	213-555-1984

Isso é obtido ao escrever um manipulador do evento "OnBeforePrint" da banda:

PascalScript:

```
procedure MasterData1OnBeforePrint(Sender: TfrxComponent);
begin
  Engine.CurX := Engine.CurX + 5;
end;
```

C++ Script:

```
void MasterData1OnBeforePrint(TfrxComponent Sender)
{
  Engine.CurX = Engine.CurX + 5;
}
```

Alterar "CurY" pode sobrepor as bandas, por exemplo:



É obtido por este script:

PascalScript:

```
procedure MasterData1OnBeforePrint(Sender: TfrxComponent);
begin
  Engine.CurY := Engine.CurY - 15;
end;
```

C++ Script:

```
void MasterData1OnBeforePrint(TfrxComponent Sender)
{
  Engine.CurY = Engine.CurY - 15;
}
```

O método `Engine.NewPage` insere uma quebra de página em qualquer ponto necessário em um relatório, depois da qual a impressão continua do topo da nova página de saída. Em nosso exemplo é possível inserir uma quebra após imprimir o segundo registro:

PascalScript:

```
procedure MasterData1OnAfterPrint(Sender: TfrxComponent);
begin
  if <Line> = 2 then
    Engine.NewPage;
end;
```

C++ Script:

```
void MasterData1OnAfterPrint(TfrxComponent Sender)
{
  if (<Line> == 2)
    Engine.NewPage();
}
```

Observe que usamos o evento "OnAfterPrint" (ou seja, após imprimir a página). Também observe que a variável de sistema "Line" retorna o número sequencial do registro.

O método `Engine.NewColumn` insere uma quebra de coluna em relatórios com diversas colunas. Se não houver mais colunas livres na página, uma nova página é criada.

Anchors

Uma âncora é um dos elementos no sistema de hiperlinks que permite saltar para qualquer elemento conectado ao objeto do relatório concluído ao clicar nele (na janela de visualização).

A âncora pode ser configurada através do método "Engine.AddAnchor". Uma âncora possui um nome e uma posição na página de um relatório. Para saltar para uma âncora com um nome especificado, digite a seguinte linha na propriedade URL de qualquer objeto do relatório:

```
#AnchorName
```

ou

```
#[AnchorName]
```

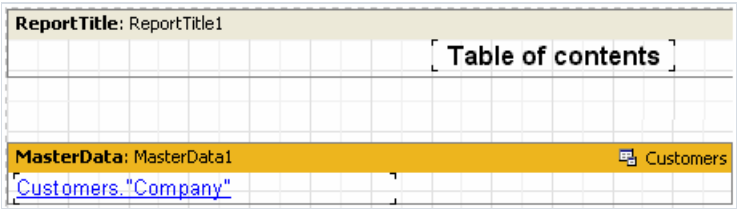
No segundo caso o FastReport substitui um valor pela expressão.

Ao clicar no objeto, um salto é executado para o local do relatório onde foi adicionada a âncora.

Use âncoras ao construir uma tabela de "Conteúdo", por exemplo com links para capítulos correspondentes. Vamos ilustrar isso no exemplo a seguir. Primeiro precisamos da familiar tabela "Clientes".

Nosso relatório terá diversas páginas (com duas páginas de design). Colocaremos a tabela de "Conteúdo" na primeira página e a lista de clientes na segunda página. Um clique em qualquer linha de Conteúdo executa um salto até o elemento correspondente do relatório.

A primeira página de design:

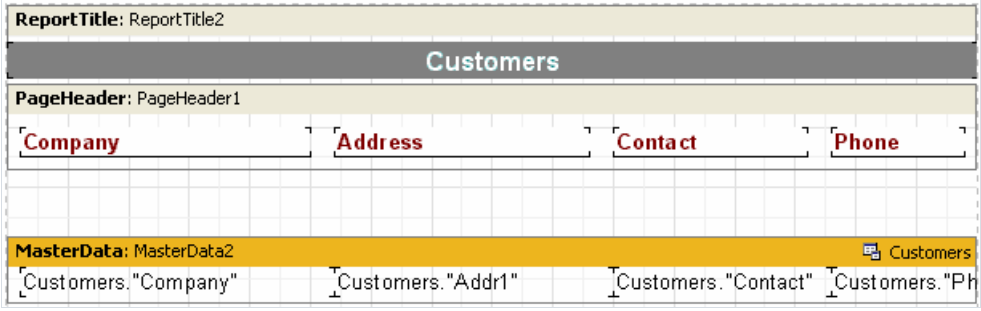


Coloque o seguinte texto na propriedade URL do objeto "Texto" contido na banda de dados

```
#[Customers."Company"]
```

e configure as propriedades da fonte como azul e sublinhado para simular o aspecto de um hiperlink.

A segunda página de design:



Para adicionar uma âncora, crie um manipulador de eventos "MasterData2.OnBeforePrint" no script:

PascalScript:

```
procedure MasterData2OnBeforePrint(Sender: TfrxComponent);
begin
  Engine.AddAnchor(<Customers."Company">);
end;
```

C++ Script:

```
void MasterData2OnBeforePrint(TfrxComponent Sender)
{
  Engine.AddAnchor(<Customers."Company">);
}
```

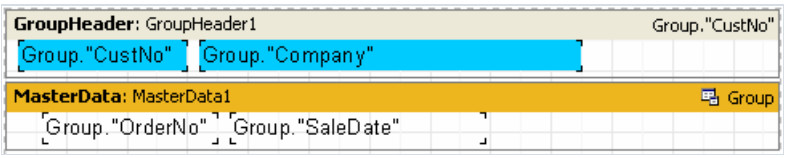
Isso é tudo que é necessário. Visualize o relatório e certifique-se de que todos os "hiperlinks" estejam funcionando.

O último item a ser mencionado é a função `Engine.GetAnchorPage`. Esta função retorna o número da página à qual foi adicionada a âncora correspondente, e é util ao criar a tabela "Conteúdo". O relatório deve ter dois passos ou a função não pode ser utilizada.

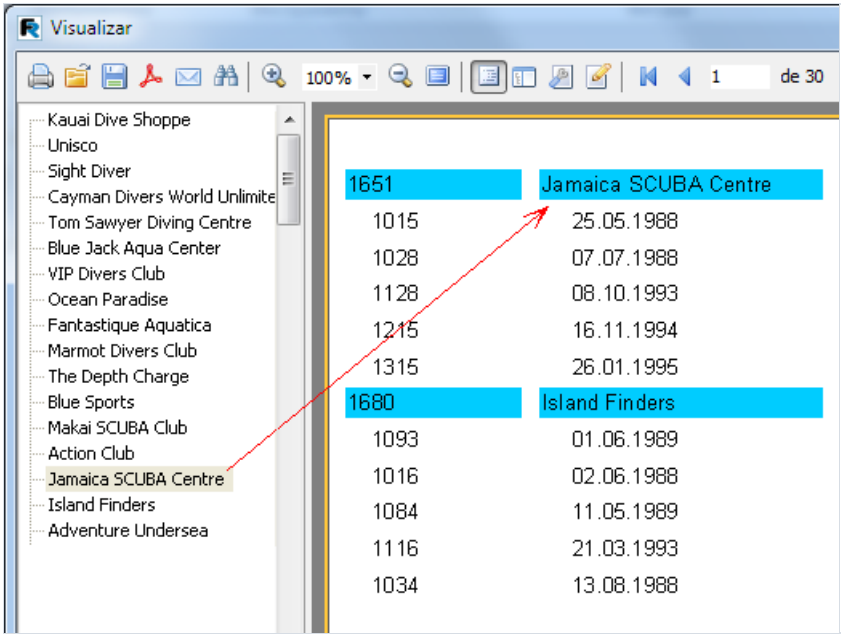
Utilização do objeto "Outline"

Como foi afirmado anteriormente, o objeto "Outline" representa a árvore do relatório que pode ser exibida na janela de visualização. Clicar em um elemento da árvore executa um salto para a página de saída do relatório que contém o elemento correspondente. Não é necessário usar um script para trabalhar com um "Outline", já que algumas bandas são capazes de criar uma árvore automaticamente. Vamos dar uma olhada em dois exemplos que mostram como "Outline" pode ser utilizado com a ajuda de bandas e um script.

Quase todas as bandas possuem a propriedade `OutlineText` para conter uma expressão que cria a árvore automaticamente. A expressão é avaliada ao criar o relatório, e seu valor é adicionado à árvore quando a banda é impressa. Desta maneira a hierarquia de elementos na árvore é similar à hierarquia de bandas no relatório, o que significa que a árvore terá elementos principais e subordinados que correspondem às bandas principais e subordinadas no relatório (por exemplo, onde o relatório tiver dois níveis de dados ou tiver grupos). Vamos usar nosso exemplo anterior para mostrar a operação de uma árvore em um relatório que possui grupos.

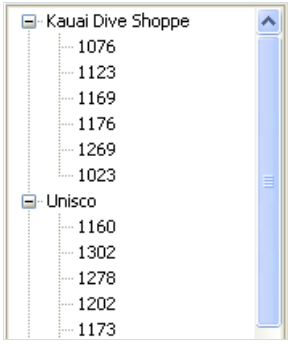


Configure o valor da propriedade da banda "GroupHeader1.OutlineText" como "<Group."Company">". Para deixar a árvore visível quando a janela de visualização for aberta, configure a propriedade `PreviewOptions.OutlineVisible` do objeto "Report" como True. Ao visualizar o relatório, você verá o seguinte:



Clicar em qualquer elemento da árvore executa um salto para a página correspondente no relatório, com o elemento selecionado no topo da janela.

Vamos adicionar o segundo nível à árvore do relatório. Apenas configure a propriedade `OutlineText` da banda "DadosMestre" como "<Group."OrderNo">" e a árvore será alterada desta maneira:



É evidente é possível obter uma navegação até os números dos pedidos, e que a hierarquia dos elementos na árvore é similar àquela no relatório.

Agora criaremos uma árvore parecida, mas usando um script em vez da propriedade `OutlineText`. No relatório, limpe as propriedades `OutlineText` de ambas as bandas e crie dois manipuladores de eventos:

"GroupHeader1.OnBeforePrint" e "MasterData1.OnBeforePrint":

PascalScript:

```
procedure GroupHeader1OnBeforePrint(Sender: TfrxComponent);
begin
  Outline.LevelRoot;
  Outline.AddItem(<Group."Company">);
end;

procedure MasterData1OnBeforePrint(Sender: TfrxComponent);
begin
  Outline.AddItem(<Group."OrderNo">);
  Outline.LevelUp;
end;

begin
end.
```

C++ Script:

```
void GroupHeader1OnBeforePrint(TfrxComponent Sender)
{
  Outline.LevelRoot;
  Outline.AddItem(<Group."Company">);
}

void MasterData1OnBeforePrint(TfrxComponent Sender)
{
  Outline.AddItem(<Group."OrderNo">);
  Outline.LevelUp;
}

{
}
```

Visualize o relatório para certificar-se de que ele funciona da mesma maneira como no exemplo anterior, no qual a árvore foi criada automaticamente. Vamos ver como a árvore é criada pelo script.

O método `Outline.AddItem` adiciona um nó filho ao nó da árvore atual e depois torna o nó filho o nó atual. Então, se `AddItem` fosse chamada diversas vezes em sequência, ela iria criar a "escada" mostrada aqui:

```
Item1
  Item2
    Item3
      ...
```

Os métodos de Outline `LevelUp` e `LevelRoot` são usados para controlar qual elemento é o atual. O primeiro método move o cursor para o elemento localizado um nível acima. Então este script:

```
Outline.AddItem('Item1');
Outline.AddItem('Item2');
Outline.AddItem('Item3');
Outline.LevelUp;
Outline.AddItem('Item4');
```

cria uma árvore como esta:

```
Item1
  Item2
    Item3
    Item4
```

Isso mostra que "Item4" é um elemento filho do elemento "Item2". No entanto, o método `LevelRoot` move o cursor até a raiz da árvore. Por exemplo, o script:

```
Outline.AddItem('Item1');
Outline.AddItem('Item2');
Outline.AddItem('Item3');
Outline.LevelRoot;
Outline.AddItem('Item4');
```

cria esta árvore:

```
Item1
  Item2
    Item3
Item4
```

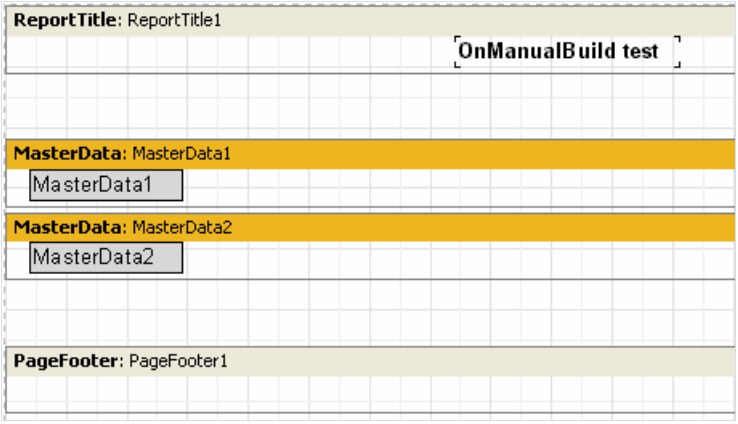
Ao entender isso se torna claro como o relatório funciona. Antes de cada título de grupo (nome da empresa) é exibida a raiz da árvore, que é transformada no elemento atual. Depois é exibida a lista de pedidos, e cada pedido é adicionado como um elemento filho da empresa. Para assegurar que todos os pedidos estejam localizados em um nível e não exibidos como uma "escada", é chamado o método `Outline.LevelUp` após cada adição de pedido para deslocar o cursor de volta ao nível da empresa.

Evento de página "OnManualBuild"

Normalmente, o núcleo do FastReport é responsável pela construção do relatório. Ele exibe as bandas do relatório em uma ordem específica e quantas vezes for necessário para os dados, criando assim um relatório completo. Às vezes é necessário exibir um relatório de forma não padrão, que o núcleo do FastReport não consegue realizar. Neste caso é possível construir um relatório manualmente usando o evento "OnManualBuild" da página de design do relatório. Se o manipulador deste evento for definido, então o núcleo do FastReport transfere controle a ele quando a saída de dados for necessária. Ao mesmo tempo, o núcleo do FastReport gerencia a exibição das bandas localizadas na página automaticamente, como "Título do Relatório", "Título da Página", "Título da Coluna", "Rodapé do Relatório", "Rodapé da Página, "Rodapé da Coluna" e "Plano de Fundo". O núcleo também gerencia a criação de novas páginas e colunas. O propósito do manipulador de eventos "OnManualBuild" é exibir bandas de dados, seus títulos e seus rodapés na ordem controlada pelo usuário.

Isso quer dizer que a essência do manipulador "OnManualBuild" é dar comandos ao núcleo do FastReport para exibir bandas em momentos específicos. O núcleo faz o resto sozinho: ele cria novas páginas quando não houver espaço livre na página atual, gerencia os scripts anexados a eventos, etc.

Vamos demonstrar um manipulador usando um exemplo simples. Este relatório possui duas bandas de dados mestre que não estão conectados a dados:



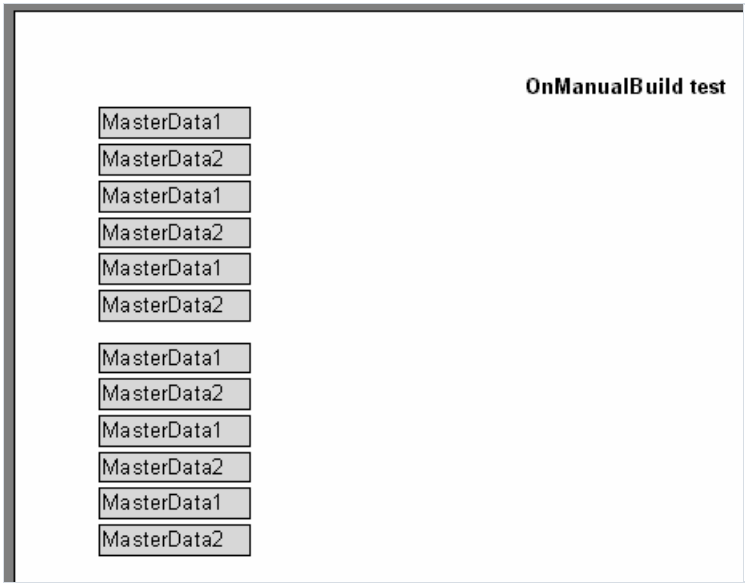
O manipulador exibirá estas bandas em ordem alternada (seis vezes para cada). Após exibir seis bandas, uma pequena lacuna será inserida.

PascalScript:

```
procedure Page1OnManualBuild(Sender: TfrxComponent);
var
  i: Integer;
begin
  for i := 1 to 6 do
  begin
    { exibir duas bandas }
    Engine.ShowBand(MasterData1);
    Engine.ShowBand(MasterData2);
    { criar uma lacuna }
    if i = 3 then
      Engine.CurY := Engine.CurY + 10;
  end;
end;
```

C++ Script:

```
void Page1OnManualBuild(TfrxComponent Sender)
{
  int i;
  for (i = 1; i <= 6; i++)
  {
    // exibir duas bandas
    Engine.ShowBand(MasterData1);
    Engine.ShowBand(MasterData2);
    // criar uma lacuna
    if (i == 3)
      Engine.CurY = Engine.CurY + 10;
  }
}
```



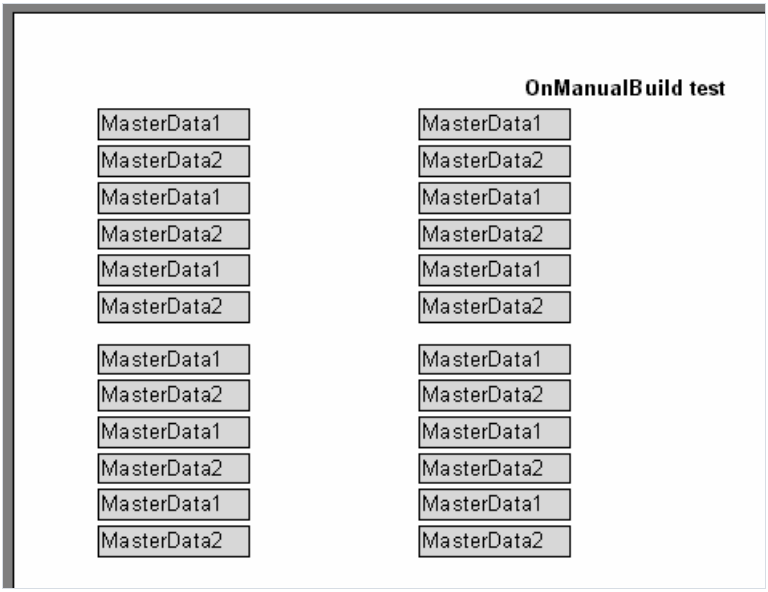
O exemplo a seguir exibe as mesmas bandas, com uma segunda cópia deslocada para a direita.

PascalScript:

```
procedure Page1OnManualBuild(Sender: TfrxComponent);
var
  i, j: Integer;
  SaveY: Extended;
begin
  SaveY := Engine.CurY;
  for j := 1 to 2 do
  begin
    for i := 1 to 6 do
    begin
      Engine.ShowBand(MasterData1);
      Engine.ShowBand(MasterData2);
      if i = 3 then
        Engine.CurY := Engine.CurY + 10;
    end;
    Engine.CurY := SaveY;
    Engine.CurX := Engine.CurX + 200;
  end;
end;
```

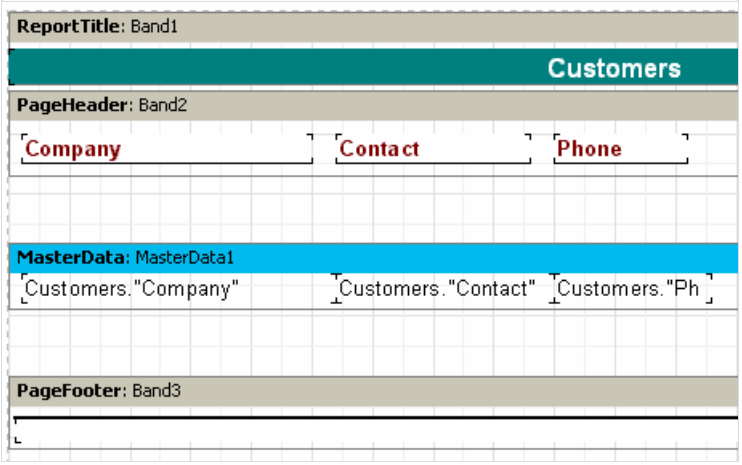
C++Script:

```
void Page1OnManualBuild(TfrxComponent Sender)
{
  int i, j;
  Extended SaveY;
  SaveY = Engine.CurY;
  for (j = 1; j <= 2; j++)
  {
    for (i = 1; i <= 6; i++)
    {
      Engine.ShowBand(MasterData1);
      Engine.ShowBand(MasterData2);
      if (i == 3)
        Engine.CurY = Engine.CurY + 10;
    }
    Engine.CurY = SaveY;
    Engine.CurX = Engine.CurX + 200;
  }
}
```



Como você pode ver, nestes exemplos controlamos apenas a saída das bandas de dados. O resto das bandas (em nosso caso, o "Título do Relatório") foi exibido automaticamente.

Finalmente mostraremos como construir um relatório com uma "Lista de clientes" (já demonstramos diversas versões anteriormente) utilizando o evento "OnManualBuild". Desta vez conecte a banda de dados à fonte de dados.



E use este script:

PascalScript:

```
procedure Page1OnManualBuild(Sender: TfrxComponent);
var
  DataSet: TfrxDataSet;
begin
  DataSet := MasterData1.DataSet;
  DataSet.First;
  while not DataSet.Eof do
  begin
    Engine.ShowBand(MasterData1);
    DataSet.Next;
  end;
end;
```

C++Script:

```
void Page1OnManualBuild(TfrxComponent Sender)
{
  TfrxDataSet DataSet;
  DataSet = MasterData1.DataSet;
  DataSet.First();
  while (!DataSet.Eof)
  {
    Engine.ShowBand(MasterData1);
    DataSet.Next();
  }
}
```

Visualize o relatório para certificar-se de que o script produz um relatório idêntico ao relatório padrão. Observe que temos um link para o conjunto de dados – nós conectamos uma variável do conjunto de dados à fonte de dados com este código:

```
DataSet := MasterData1.DataSet;
```

Se a banda DadosMestre não estiver conectada à fonte de dados, então o link à fonte de dados necessária pode ser feito da seguinte maneira:

```
DataSet := Report.GetDataSet('Customers');
```

Criação de objetos no script

É possível adicionar objetos novos a um relatório usando um script. Vamos demonstrar como isso é feito através de um exemplo simples. Crie um relatório em branco e insira este código no procedimento principal do script:

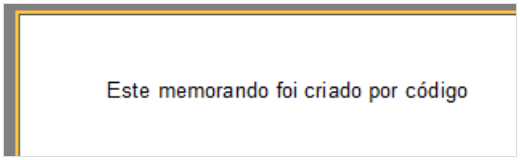
PascalScript:

```
var
  Band: TfrxReportTitle;
  Memo: TfrxMemoView;
begin
  Band := TfrxReportTitle.Create(Page1);
  Band.Height := 20;
  Memo := TfrxMemoView.Create(Band);
  Memo.SetBounds(10, 0, 100, 20);
  Memo.Text := 'Este memorando foi criado por código';
end.
```

C++ Script:

```
TfrxReportTitle Band;
TfrxMemoView Memo;
{
  Band = TfrxReportTitle.Create(Page1);
  Band.Height = 20;
  Memo = TfrxMemoView.Create(Band);
  Memo.SetBounds(10, 0, 100, 20);
  Memo.Text = "Este memorando foi criado por código";
}
```

Visualize o relatório:



Observe que não destruímos os objetos FastReport que criamos neste exemplo. Isso não é necessário, já que objetos do FastReport são destruídos automaticamente pelo aplicativo Delphi após concluir o relatório. Também observe que quando criamos objetos Delphi padrão em um script (como `TStringLists`), também precisamos destruí-los no script, já que isso não é feito automaticamente pelo aplicativo.

Relatórios tipo tabela de referência cruzada

Este tipo de relatório possui uma estrutura em tabela, o que significa que ele consiste de fileiras e colunas. Ao criar o design não é possível saber quantas linhas e colunas terá o relatório de saída. É por isso que um relatório não é estendido apenas para baixo (como nos tipos de relatório descritos anteriormente), mas também para os lados. Um exemplo típico de um relatório tipo tabela de referência cruzada é mostrado abaixo.

Vamos dar uma olhada nos elementos da tabela:

	1	2	3	4
a	a1	a2	a3	a4
b	b1	b2	b3	b4

Na ilustração vemos uma tabela com duas linhas (fileiras) e quatro colunas, onde "a" e "b" são títulos de linhas, "1", "2", "3" e "4" são títulos de colunas, e "a1".."a4" e "b1".."b4" são células. Para criar um relatório deste tipo, precisamos de apenas um conjunto de dados (de uma consulta ou uma tabela) que possua três campos e contenha os seguintes valores:

```
a 1 a1
a 2 a2
a 3 a3
a 4 a4
b 1 b1
b 2 b2
b 3 b3
b 4 b4
```

Você pode ver que o primeiro campo contém a letra da linha, o segundo contém o número da coluna e o terceiro contém o conteúdo da célula na interseção da linha e coluna especificadas. Ao exbir o relatório, FastReport cria uma tabela na memória e a preenche com dados. Deste modo a tabela é estendida dinamicamente, criando linhas e colunas onde elas ainda não existirem.

Títulos podem estar presentes em mais de um nível, conforme é ilustrado aqui:

	10		20	
	1	2	1	2
a	a10.1	a10.2	a20.1	a20.2
b	b10.1	b10.2	b20.1	b20.2

Neste exemplo, o número (ou índice) da coluna é composto, ou seja, ele é composto de dois valores. Este relatório é gerado dos seguintes dados:

```
a 10 1 a10.1
a 10 2 a10.2
a 20 1 a20.1
a 20 2 a20.2
b 10 1 b10.1
b 10 2 b10.2
b 20 1 b20.1
b 20 2 b20.2
```

Aqui o primeiro campo contém o índice da linha como no caso anterior, o segundo e terceiro campo contêm os

índices das colunas e o último campo contém o valor da célula. Veja como FastReport constrói a tabela de memória ao usar dados de tabela de referência cruzada com títulos complexos:

	10	10	20	20
	1	2	1	2
a	a10.1	a10.2	a20.1	a20.2
b	b10.1	b10.2	b20.1	b20.2

Ao exibir o relatório desta tabela de memória, FastReport une as células de título que possuem o mesmo valor e estão localizadas no mesmo nível.

Aqui há um relatório de tabela de referência cruzada mais complexo, que incorpora totais intermediários e gerais:

	10			20			Total
	1	2	Total	1	2	Total	
a	a10.1	a10.2	a10.1+a10.2	a20.1	a20.2	a20.1+a20.2	sum(a)
b	b10.1	b10.2	b10.1+b10.2	b20.1	b20.2	b20.1+b20.2	sum(b)
Total	a10.1+ b10.1	a10.2+ b10.2	a10.1+b10.1+ a10.2+b10.2	a20.1+ b20.1	a20.2+ b20.2	a20.1+b20.1+ a20.2+b20.2	sum(a)+ sum(b)

Este relatório é derivado dos mesmos dados dos relatórios anteriores. Os valores nas células que são destacadas com a cor nova são calculados automaticamente, e não estão presentes no conjunto de dados original.

Construção de um relatório de tabela de referência cruzada

Agora vamos passar da teoria para a prática. Vamos construir um relatório de tabela de referência cruzada que exiba os salários dos funcionários em um período de quatro anos. Para fazer isso, precisamos da tabela "crosstest". A tabela contém dados como:

```
Name   Year   Salary
Ann    1999   3300
Ben    2002   2000
...
```

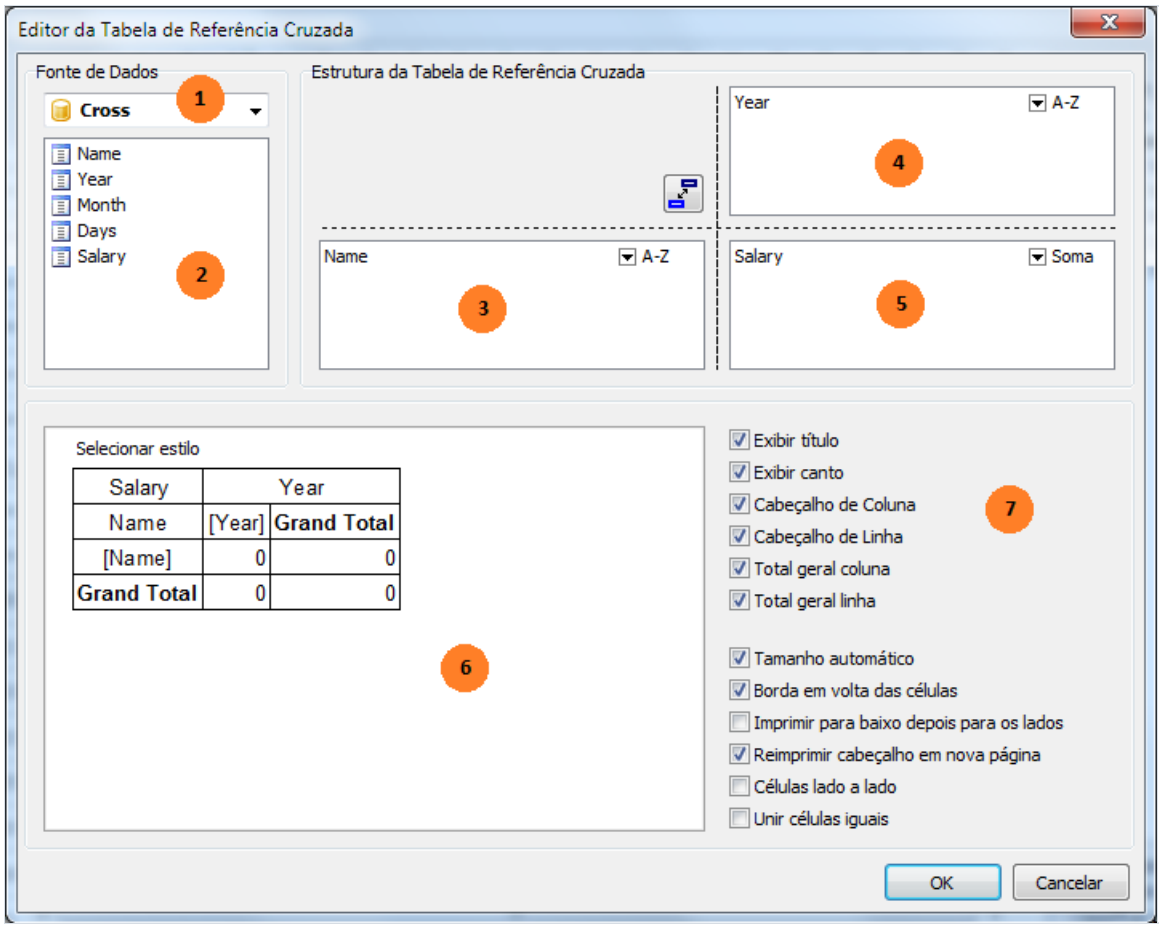
Crie um relatório novo. Coloque um componente "Tabela ADO" na página e configure suas propriedades:

```
Table1:
DatabaseName = 'c:\Program Files\FastReports\FastReport 6\Demos\Main'
TableName = 'crosstest.db'
```

Então selecione o objeto "Tabela de Referência Cruzada de BD" na barra de ferramentas de objetos do designer e clique na página de design para colocar o objeto nela:



Todas as configurações são efetuadas no editor de tabela de referência cruzada. Abra-o ao clicar duas vezes no objeto:



Legenda dos itens mostrados acima:

1 – uma lista suspensa das fontes de dados disponíveis

2 – a lista de campos na fonte de dados selecionada; os campos desta lista podem ser arrastados para as listas número 3, 4 ou 5

3 – a lista de campos que geram cabeçalhos de linhas (fileiras)

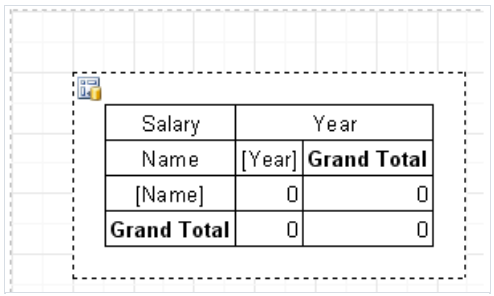
4 – a lista de campos que geram cabeçalhos de colunas

5 – a lista de campos que geram células da tabela

6 – visualização da estrutura da tabela

7 – opções da estrutura: exibição de títulos, totais, etc.

Somente o mouse pode ser usado neste editor para fazer alterações. Para nosso exemplo é apenas necessário arrastar campos da lista 2 para as listas 3, 4 e 5 (no diagrama acima). Depois disso feche o editor ao clicar no botão OK. O objeto tabela de referência cruzada agora exibe sua estrutura:



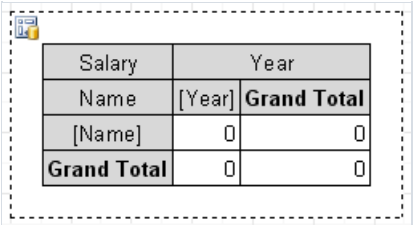
Salary	Year	
Name	[Year]	**Grand Total**
[Name]	0	0
Grand Total	0	0

Quando o relatório for visualizado, você verá uma tabela parecida com esta:

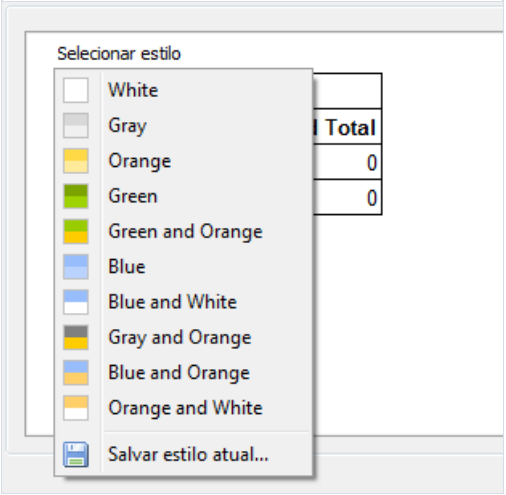
Salary	Year				
Name	1999	2000	2001	2002	**Grand Total**
Ann	3300	2700	3100	1700	10800
Ben	3900	2100		1800	7800
Catherine	6100	3200			9300
Den		3999	8100		12099
Grand Total	13300	11999	11200	3500	39999

Modificar a aparência

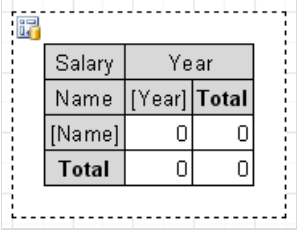
Vamos modificar a aparência do objeto tabela de referência cruzada. A primeira coisa que queremos fazer é mudar a cor dos títulos e exibir "Total" em vez de "Grand Total". Isso é muito fácil: para alterar a cor do título, clique nas células "Year", "Name" e "Grand Total" e selecione cinza usando o botão de plano de fundo na barra de ferramentas.



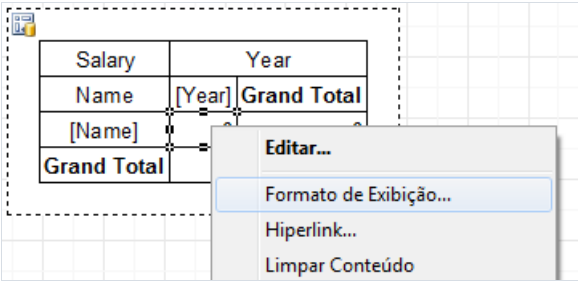
Também podemos usar um conjunto de estilos predefinidos. Eles estão disponíveis no editor de tabela de referência cruzada – clique em "Selecionar estilo" e escolha um.



Para alterar os dois textos "Grand Total", clique duas vezes em cada célula, o que abre o editor de texto onde podemos digitar "Total":



Para formatar os valores da moeda, selecione a primeira célula (interseção de [Name] e [Year] em nosso exemplo), clique com o botão direito para exibir o menu de contexto e selecione "Formato de Exibição...":



Selecione o formato necessário e feche o editor de formato. Tudo isso produz o seguinte relatório:

Salary	Year				
Name	1999	2000	2001	2002	**Total**
Ann	$3 300,00	$2 700,00	$3 100,00	$1 700,00	$10 800,00
Ben	$3 900,00	$2 100,00		$1 800,00	$7 800,00
Catherine	$6 100,00	$3 200,00			$9 300,00
Den		$3 999,00	$8 100,00		$12 099,00
Total	$13 300,00	$11 999,00	$11 200,00	$3 500,00	$39 999,00

Utilização de funções

Em nosso exemplo podemos ver a soma do salário de cada funcionário em um intervalo de quatro anos na linha "Total". É possível usar qualquer uma das seguintes funções de agregação:

`SUM` – soma de valores

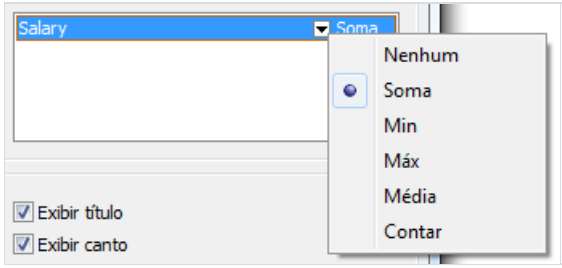
`MIN` – valor mínimo

`MAX` – valor máximo

`AVG` – valor médio

`COUNT` – número de valores

Vamos usar a função `MIN` em nosso exemplo. Abra o editor de tabela de referência cruzada e na lista 5 (o item de campo "Salary") clique na seta para baixo.



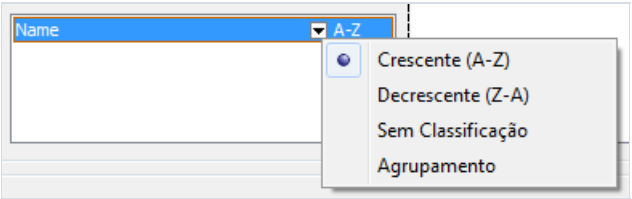
Selecione a função `MIN` na lista suspensa. Agora podemos mudar o texto nas células de totais de "Total" para "Mínimo". O relatório concluído tem o seguinte aspecto:

	1999	2000	2001	2002	**Minimum**
Ann	3300	2700	3100	1700	**1700**
Ben	4300	2400		2000	**2000**
Catherine	6100	3200			**3200**
Den		3999	8100		**3999**
Minimum	**3300**	**2400**	**3100**	**1700**	**1700**

Classificação de valores

Por padrão, linhas e colunas são arranjadas em ordem ascendente, de maneira numérica ou alfabética dependendo do tipo de dados. É possível configurar modos de classificação independentes para linhas e colunas. Estes são os modos de classificação: "arranjar em ordem crescente", "arranjar em ordem decrescente" e "não classificar". Sem classificação, as linhas/colunas são exibidas na ordem padrão do banco de dados.

Vamos alterar a classificação das colunas em nosso exemplo. Coloque os anos em ordem decrescente. Para fazer isso, abra o editor de tabela de referência cruzada, selecione o elemento de coluna "Year" e altere o modo de classificação ao clicar na seta para baixo e selecionar Decrescente:



Feche o editor e visualize o relatório. Ele terá este aspecto:

	2002	2001	2000	1999	**Total**
Ann	1700	3100	2700	3300	**10800**
Ben	2000		2400	4300	**8700**
Catherine			3200	6100	**9300**
Den		8100	3999		**12099**
Total	**3700**	**11200**	**12299**	**13700**	**40899**

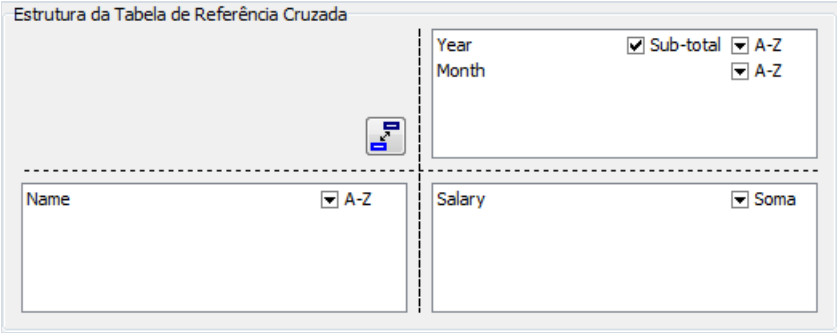
Tabelas com cabeçalhos compostos

Nosso exemplo anterior possuía um único valor por linha e cabeçalhos únicos de colunas. Vamos dar uma olhada em um design de tabela de referência cruzada que possui um cabeçalho composto com mais de um valor. Os dados de origem possuem o seguinte formato:

```
Name    Year    Month    Days    Salary
Ann     1999    2        3       1000
Ben     2002    1        5       2000
...
```

Adicionamos os campos "Month" e "Days" que contêm o número do mês e o número de dias úteis, respectivamente. É possível criar diversos relatórios com estes dados, por exemplo: salários de funcionários em cada ano, divididos por meses.

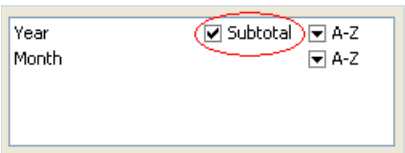
Que tipo de relatório iremos obter? Ele deve se parecido ao relatório do exemplo anterior, porém os dados anuais devem ser divididos por meses. O objeto tabela de referência cruzada deve ser configurado da mesma maneira como antes. Desta vez também arrastaremos o campo "Month" até a lista de cabeçalhos de colunas, conforme é mostrado aqui:



Na visualização vemos o relatório a seguir:

	1999					2000				2001				2002		Grand Total
	2	10	11	12	Total	1	2	3	Total	1	2	3	Total	1	Total	
Ann	1000		1100	1200	**3300**	1300	1400		**2700**		1500	1600	**3100**	1700	**1700**	**10800**
Ben		2100	2200		**4300**		2400		**2400**				**0**	2000	**2000**	**8700**
Catherine		3000	3100		**6100**			3200	**3200**				**0**		**0**	**9300**
Den					**0**	3999			**3999**	4000	4100		**8100**		**0**	**12099**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**	**5299**	**3800**	**3200**	**12299**	**4000**	**5600**	**1600**	**11200**	**3700**	**3700**	**40899**

Observe que o FastReport automaticamente adicionou uma coluna de totais intermediários, que são exibidos após cada ano. Isso pode ser desativado no editor da tabela de referência cruzada ao desmarcar o sinalizador "Subtotal" do elemento de coluna "Year":



Também observe que o último elemento de coluna na lista de cabeçalhos de colunas nunca possui um sinalizador "Subtotal" (incluindo no caso de um elemento único). Em nosso exemplo, não precisamos dos totais intermediários de cada mês, então o sinalizador "Subtotal" pode ser desmarcado.

Há outro recurso de totais intermediários, se eles forem usados: pode ser melhor colocar o cabeçalho do total

intermediário como "Ano + total do ano" em vez de "Total". No objeto tabela de referência cruzada na página do relatório, clique duas vezes na célula de total intermediário e digite no editor de texto:

```
Total for [Value]
```

Na criação do relatório, a expressão "[Value]" é substituída pelo valor real do cabeçalho da coluna na célula acima:

	1999				
	2	10	11	12	**Total for 1999**
Ann	1000		1100	1200	**3300**
Ben		2100	2200		**4300**
Catherine		3000	3100		**6100**
Den					**0**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**

Ajustar a largura da célula

Na ilustração acima vemos que FastReport ajusta as larguras das células de maneira automática, de modo que valores maiores de células caibam nas células. No entanto, isso pode não ser desejado em alguns casos, já que colunas muito largas podem ser feias. O que pode ser feito a respeito? Vamos dar uma olhada em três métodos de controle da largura das células.

O método mais simples para controlar a largura da célula é adicionar quebras de linha ao texto dos totais intermediários:

```
Total
for [Value]
```

A tabela resultante é mais compacta:

	1999				
	2	10	11	12	**Total for 1999**
Ann	1000		1100	1200	**3300**
Ben		2100	2200		**4300**
Catherine		3000	3100		**6100**
Den					**0**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**

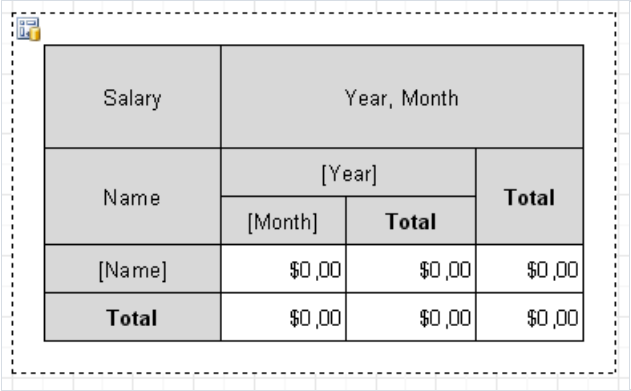
No entanto, há circunstâncias nas quais é difícil ou impossível quebrar linhas manualmente. Por isso o objeto tabela de referência cruzada possui as propriedades `MinWidth` e `MaxWidth` (que se referem à largura das células). Estas propriedades podem ser acessadas apenas através do inspetor de objetos.

Por padrão, `MinWidth` é 0 e `MaxWidth` é 200, o que suficiente na maioria dos casos. O segundo método de controlar a largura das células é alterar estes valores de acordo com necessidades especiais.

Então, em nosso exemplo podemos configurar `MinWidth` e `MaxWidth` como 50. Isso significa que uma célula de dados deve ter pelo menos 50 pixels de largura, mesmo se o valor da célula couber em um espaço com menos pixels. Para valores grandes de célula, a largura da célula está limitada ao valor de `MaxWidth`, e o texto na célula é quebrado conforme for necessário. Logo, nosso exemplo agora tem este aspecto:

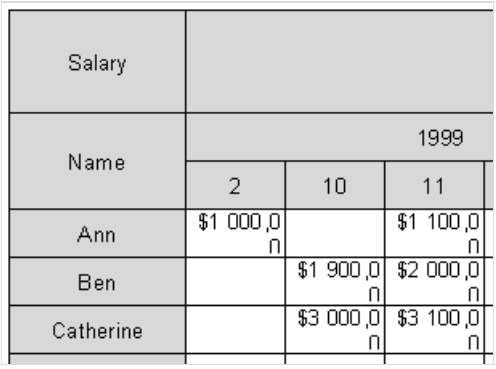
	1999				
	2	10	11	12	**Total for 1999**
Ann	1000		1100	1200	**3300**
Ben		2100	2200		**4300**
Catherine		3000	3100		**6100**
Den					**0**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**

O terceiro método de controlar a largura da célula é modificar as laguras das células manualmente. Para poder fazer isso, a propriedade Tamanho Automático deve ser configurada como False. A célula da tabela de referência cruzada pode então ser modificada usando o mouse. No objeto tabela de referência cruzada da página do relatório, o cursor do mouse muda quando estiver sobre bordas de células, permitindo que as bordas sejam arrastadas. Segue um exemplo do que pode ser obtido:



Salary	Year, Month		
Name	[Year]		**Total**
	[Month]	**Total**	
[Name]	$0,00	$0,00	$0,00
Total	$0,00	$0,00	$0,00

Lembre-se: se Tamanho Automático estiver desabilitado, a tabela de referência cruzada não irá ajustar as larguras e alturas das células de forma automática. Assim você pode obter algo parecido com isto na visualização do relatório:



Salary			
Name	1999		
	2	10	11
Ann	$1 000,00		$1 100,00
Ben		$1 900,00	$2 000,00
Catherine		$3 000,00	$3 100,00

Se isso acontecer, apenas aumente um pouco a largura das células.

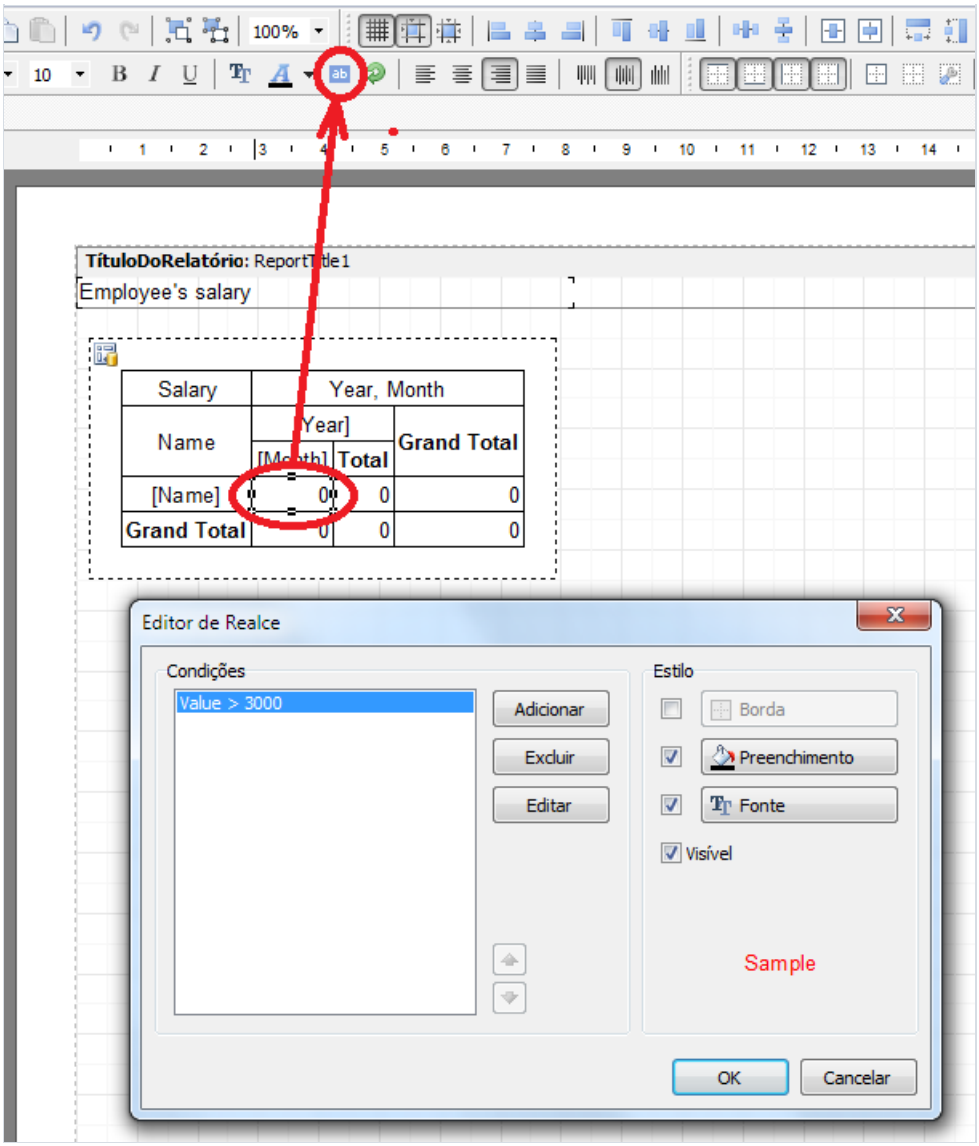
Cor da fonte e realce

Às vezes é necessário realçar valores ou mudar a cor da fonte. Isso foi abordado no exemplo do relatório de grupo, no qual usamos o realce condicional para objetos "Texto". No exemplo atual, o realce condicional também pode ser útil para nós.

Para adicionar realce a nosso relatório de exemplo, e assumindo que precisamos mudar a cor da fonte para valores maiores que 3000, selecione o objeto que representa a célula da tabela e configure os parâmetros de realce ao clicar no botão de realce na barra de ferramentas. A janela familiar do editor de realce será aberta, e nela podemos configurar esta condição:

```
Value > 3000
```

e configurar a cor da fonte como vermelho:



Isso é tudo. Feche o editor clicando no botão OK e visualize o relatório:

	1999					2000			
	2	10	11	12	**Total for 1999**	1	2	3	**Total for 2000**
Ann	1000		1100	1200	**3300**	1300	1400		**2700**
Ben		2100	2200		**4300**		2400		**2400**
Catherine		3000	3100		**6100**			3200	**3200**
Den					**0**	3999			**3999**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**	**5299**	**3800**	**3200**	**12299**

É possível realçar os valores totais de maneira similar, se necessário, e também células e linhas, usando os botões plano de fundo e cor da borda.

Gerenciar uma tabela de referência cruzada em um script

Se os métodos demonstrados acima não forem capazes de criar o aspecto necessário para o relatório, é possível usar um script. O objeto "Tabela de Referência Cruzada" possui os seguintes eventos:

Evento	Descrição
OnAfterPrint	o evento é chamado após imprimir uma tabela
OnBeforePrint	o evento é chamado antes de imprimir uma tabela
OnCalcHeight	o evento é chamado antes de calcular a altura de uma fileira na tabelao manipulador do evento pode ser configurado com a altura necessária ou "0" se a fileira precisa ser ocultada
OnCalcWidth	o evento é chamado antes de calcular a largura da coluna em uma tabelao manipulador do evento pode ser configurado com a largura necessária ou "0" se a coluna precisa ser ocultada
OnPrintCell	o evento é chamado antes de exibir uma célula da tabelao manipulador do evento pode modificar o design da célula ou seu contéudo
OnPrintColumnHeader	o evento é chamado antes de exibir o título da colunao manipulador do evento pode modificar o design ou o contéudo da célula título
OnPrintRowHeader	o evento é chamado antes de exibir o título da fileirao manipulador do evento pode modificar o design ou o contéudo da célula título

É possível usar os seguintes métodos do objeto "Tabela de Referência Cruzada" nestes eventos:

Método	Descrição
function ColCount: Integer	retorna o número de colunas na tabela
function RowCount: Integer	retorna o número de fileiras na tabela
function IsGrandTotalColumn(Index: Integer): Boolean	retorna "True" se a coluna 'Index' for um total
function IsGrandTotalRow(Index: Integer): Boolean	retorna "True" se a fileira 'Index' for um total
function IsTotalColumn(Index: Integer): Boolean	retorna "True" se a coluna 'Index' for um subtotal
function IsTotalRow(Index: Integer): Boolean	retorna "True" se a fileira 'Index' for um subtotal
procedure AddValue(const Rows, Columns, Cells: array of Variant)	adiciona um valor à tabela

Agora vamos demonstrar como realçar a terceira coluna (em nosso exemplo, a data "November 1999"). Selecione o objeto tabela de referência cruzada na página de design do relatório, clique na guia Eventos no inspetor de objetos, localize o evento "OnPrintCell" e crie um manipulador na página Código ao clicar duas vezes na lista vazia

à direita do nome do evento. O editor de script será aberto com a declaração básica criada para você. Então adicione o código necessário no bloco 'begin...end' vazio da declaração:

Pascal script:

```
procedure Cross1OnPrintCell(Memo: TfrxMemoView;
  RowIndex, ColumnIndex, CellIndex: Integer;
  RowValues, ColumnValues, Value: Variant);
begin
  if ColumnIndex = 2 then
    Memo.Color := clRed;
end;
```

C++ Script:

```
void Cross1OnPrintCell(TfrxMemoView Memo,
  int RowIndex, int ColumnIndex, int CellIndex,
  Variant RowValues, Variant ColumnValues, Variant Value)
{
  if (ColumnIndex == 2) { Memo.Color = clRed; }
}
```

Veremos o seguinte ao visualizar o relatório:

	1999				
	2	10	11	12	**Total**
Ann	1000		1100	1200	**3300**
Ben		2100	2200		**4300**
Catherine		3000	3100		**6100**
Den					**0**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**

Para realçar o título de uma coluna, crie um manipulador do evento "OnPrintColumnHeader" de maneira similar:

Pascal script:

```
procedure Cross1OnPrintColumnHeader(Memo: TfrxMemoView;
  HeaderIndexes, HeaderValues, Value: Variant);
begin
  if (VarToStr(HeaderValues[0]) = '1999') and
    (VarToStr(HeaderValues[1]) = '11') then
    Memo.Color := clRed;
end;
```

C++ Script:

```
void Cross1OnPrintColumnHeader(TfrxMemoView Memo,
  Variant HeaderIndexes, Variant HeaderValues, Variant Value)
{
  if ((VarToStr(HeaderValues[0]) == "1999") &&
     (VarToStr(HeaderValues[1]) == "11"))
  {
    Memo.Color = clRed;
  }
}
```

A visualização do relatório agora exibe:

	1999				
	2	10	11	12	**Total**
Ann	1000		1100	1200	**3300**
Ben		2100	2200		**4300**
Catherine		3000	3100		**6100**
Den					**0**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**

O script funciona deste modo: o manipulador do evento "OnPrintCell" é chamado antes de imprimir uma célula na área de dados da tabela (observe que as células título da tabela chamam o manipulador "OnPrintColumnHeader" ou "OnPrintRowHeader"). Os parâmetros do manipulador "OnPrintCell" incluem: um link para o objeto "Texto" que representa a célula da tabela (o parâmetro "Memo") e o "endereço" da célula como a localização da fileira, coluna e célula (a célula é relevante se sua tabela de referência cruzada conter células com vários níveis) como os parâmetros "RowIndex", "ColumnIndex" e "CellIndex", respectivamente. A lista de parâmetros também especifica os valores dos cabeçalhos como Variant (os parâmetros "RowValues" e "ColumnValues") e o parâmetro Variant "Value", que contém o conteúdo da célula.

Em nosso exemplo é mais fácil especificar o "endereço" usando "RowIndex" e "ColumnIndex". A numeração de colunas e fileiras começa em "0", então "ColumnIndex = 2" se refere à terceira coluna. Também é possível especificar a coluna correta ao ver os dados de seu conteúdo (precisamos do 11° mês de 1999):

Pascal script:

```
procedure Cross1OnPrintCell(Memo: TfrxMemoView;
  RowIndex, ColumnIndex, CellIndex: Integer;
  RowValues, ColumnValues, Value: Variant);
begin
  if (VarToStr(ColumnValues[0]) = '1999') and
    (VarToStr(ColumnValues[1]) = '11') then
    Memo.Color := clRed;
end;
```

C++ Script:

```
void Cross1OnPrintCell(TfrxMemoView Memo,
  int RowIndex, int ColumnIndex, int CellIndex,
  Variant RowValues, Variant ColumnValues, Variant Value)
{
  if ((VarToStr(ColumnValues[0]) == "1999") &&
     (VarToStr(ColumnValues[1]) == "11"))
  {
    Memo.Color = clRed;
  }
}
```

Os parâmetros "RowValues" e "ColumValues" são arrays do tipo Variant com base zero. O elemento "0" está no nível mais alto do título da tabela, o elemento "1" está no próximo nível, etc. Em nosso exemplo, "ColumnValues[0]" contém anos e "ColumnValues[1]" contém meses.

Por que a função `VarToStr` é necessária? Ela previne erros durante a conversão de tipos. Ao trabalhar com o tipo Variant, FastReport tenta converter cadeias para o formato numérico automaticamente, o que pode causar erros ao converter os valores das colunas "Total" e "Grand Total".

O manipulador do evento "OnPrintColumnHeader" é chamado durante a saída das células de título das colunas. A lista de parâmetros é parecida à do manipulador "OnPrintCell", apesar do "endereço" da célula (os parâmetros "HeaderIndexes" e "HeaderValues") ter um formato diferente. O parâmetro "HeaderValues" contém os mesmos valores que "ColumnValues" e "RowValues" do manipulador "OnPrintCell". O parâmetro "HeaderIndexes" também é um array de valores do tipo Variant, e ele contém o endereço da célula título em um formato diferente: o elemento "0" é o índice no nível mais alto do título da tabela, o elemento "1" está no próximo nível, etc. Para entender melhor o princípio de numeração das células, consulte a figura abaixo:

	0					1				2			
	0	1	2	3	4	0	1	2	3	0	1	2	3
0	1000		1100	1200	**3300**	1300	1400		**2700**		1500	1600	**3100**
1		2100	2200		**4300**		2400		**2400**				**0**
2		3000	3100		**6100**			3200	**3200**				**0**
3					**0**	3999			**3999**	4000	4100		**8100**
4	**1000**	**5100**	**6400**	**1200**	**13700**	**5299**	**3800**	**3200**	**12299**	**4000**	**5600**	**1600**	**11200**

Em nosso exemplo foi mais fácil usar o parâmetro "HeaderValues", mas seria possível usar o seguinte manipulador em seu lugar:

Pascal script:

```
procedure Cross1OnPrintColumnHeader(Memo: TfrxMemoView;
  HeaderIndexes, HeaderValues, Value: Variant);
begin
  if (HeaderIndexes[0] = 0) and (HeaderIndexes[1] = 2) then
    Memo.Color := clRed;
end;
```

C++ Script:

```
void Cross1OnPrintColumnHeader(TfrxMemoView Memo,
  Variant HeaderIndexes, Variant HeaderValues, Variant  Value)
{
  if ((HeaderIndexes[0] == 0) && (HeaderIndexes[1] == 2)) { Memo.Color = clRed; }
}
```

Ajustar o tamanho da fileira/coluna

A largura e altura das colunas e fileiras da tabela podem ser ajustadas através dos manipuladores de evento "OnCalcWidth" e "OnCalcHeight". Vamos ver como aumentar a largura da coluna que contém o 11° mês de 1999 com o exemplo a seguir. Crie um manipulador de evento "OnCalcWidth":

Pascal script:

```
procedure Cross1OnCalcWidth(ColumnIndex: Integer;
  ColumnValues: Variant; var Width: Extended);
begin
  if (VarToStr(ColumnValues[0]) = '1999') and
    (VarToStr(ColumnValues[1]) = '11') then
    Width := 100;
end;
```

C++ Script:

```
void Cross1OnCalcWidth(int ColumnIndex, variant ColumnValues, Extended &Width)
{
  if ((VarToStr(ColumnValues[0]) == "1999") &&
    (VarToStr(ColumnValues[1]) = "11"))
    {
      Width = 100;
    }
}
```

E o relatório será:

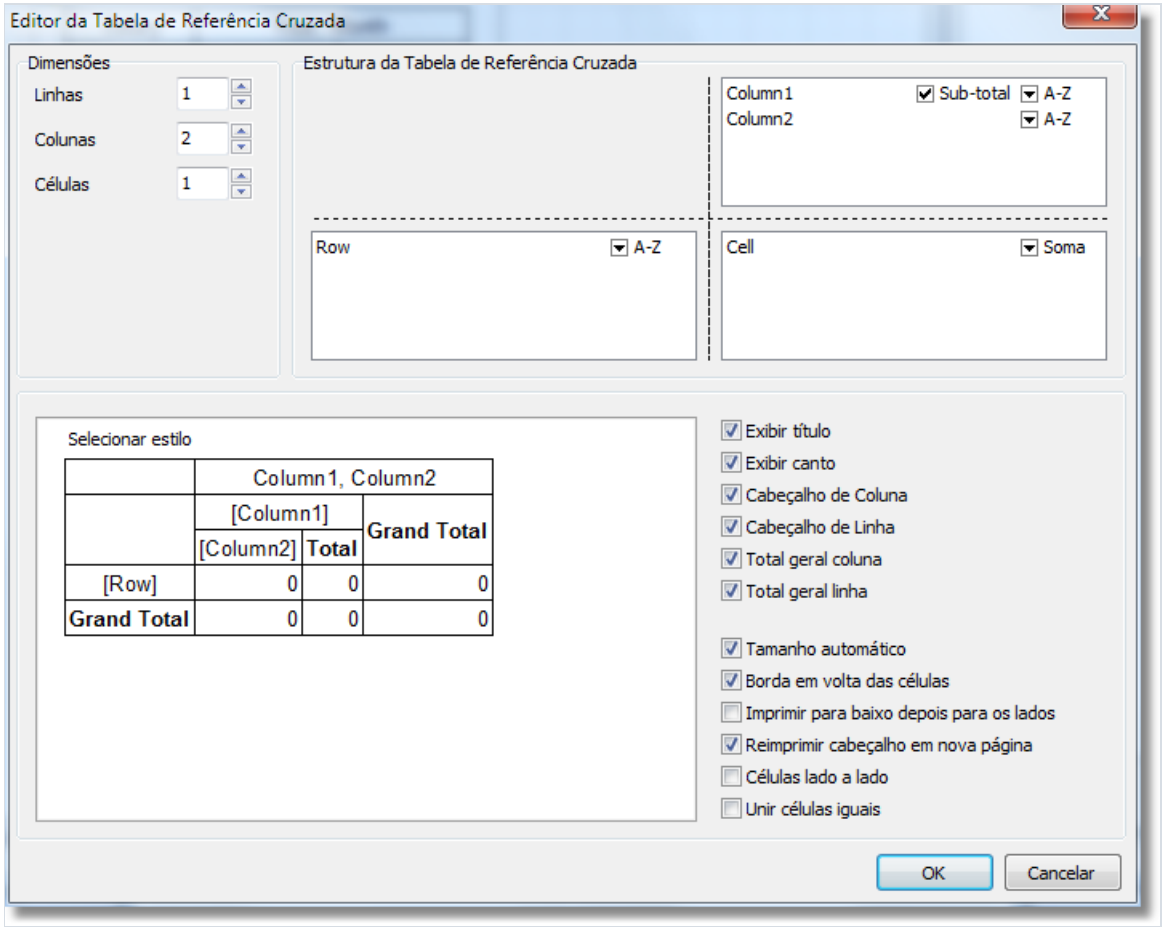
	1999				
	2	10	11	12	**Total**
Ann	1000		1100	1200	**3300**
Ben		2100	2200		**4300**
Catherine		3000	3100		**6100**
Den					**0**
Grand Total	**1000**	**5100**	**6400**	**1200**	**13700**

Para ocultar uma coluna em nosso exemplo, configure Width como zero. Observe que os totais não são calculados novamente, já que a tabela já está preenchida com valores nesse ponto.

Preencher uma tabela manualmente

Há duas versões do objeto tabela de referência cruzada, "Tabela de Referência Cruzada de BD" e "Tabela de Referência Cruzada". Até agora nós usamos o primeiro objeto conectado a dados de uma tabela de BD. O objeto é preenchido automaticamente quando o relatório é executado. Vamos dar uma olhada no segundo objeto, "Tabela de Referência Cruzada".

O objeto "Tabela de Referência Cruzada" não é conectado a uma tabela de BD. Por isso ela deve ser preenchida manualmente com dados. O objeto "Tabela de Referência Cruzada" possui um editor parecido ao objeto "Tabela de Referência Cruzada de BD", apesar de haver uma diferença, já que as dimensões dos títulos das tabelas e das células devem ser especificadas em vez de serem configuradas pelos campos do BD:



Vamos demonstrar a utilização de um objeto "Tabela de Referência Cruzada" com um exemplo. Coloque um objeto "Tabela de Referência Cruzada" na página de design do reltório e configure suas propriedades conforme é mostrado acima: o número de níveis do título das fileiras é 1, no título das colunas ele é 2 e nas células é 1. Vamos preencher a tabela com dados usando o manipulador de evento "OnBeforePrint":

PascalScript:

```
procedure Cross1OnBeforePrint(Sender: TfrxComponent);
begin
  with Cross1 do
  begin
    AddValue(['Ann'], [2001, 2], [1500]);
    AddValue(['Ann'], [2001, 3], [1600]);
    AddValue(['Ann'], [2002, 1], [1700]);
    AddValue(['Ben'], [2002, 1], [2000]);
    AddValue(['Den'], [2001, 1], [4000]);
    AddValue(['Den'], [2001, 2], [4100]);
  end;
end;
```

C++ Script:

```
void Cross1OnBeforePrint(TfrxComponent Sender)
{
  Cross1.AddValue(["Ann"], [2001, 2], [1500]);
  Cross1.AddValue(["Ann"], [2001, 3], [1600]);
  Cross1.AddValue(["Ann"], [2002, 1], [1700]);
  Cross1.AddValue(["Ben"], [2002, 1], [2000]);
  Cross1.AddValue(["Den"], [2001, 1], [4000]);
  Cross1.AddValue(["Den"], [2001, 2], [4100]);
}
```

No manipulador, os dados sao adicionados à tabela através do médoto `TfrxCrossView.AddValue`. Este método tem três parâmetros, e todos são um array do tipo Variant. O primeiro parâmetro é o valor da fileira, o segundo é o valor da coluna e o terceiro é o valor da célula. Observe que o número de valores em cada array deve corresponder às configurações do objeto! Em nosso exemplo, o objeto possui um nível no título da fileira, dois nível no título da coluna e um nível de células, então os parâmetros do array Variant do método AddValue precisam de um valor para as fileiras, dois valores para as colunas e um valor para as células.

Ao visualizar, o relatório é exibido deste modo:

	2001				2002		Grand Total
	1	2	3	**Total**	1	**Total**	
Ann		1500	1600	**3100**	1700	**1700**	**4800**
Ben				**0**	2000	**2000**	**2000**
Den	4000	4100		**8100**		**0**	**8100**
Grand Total	**4000**	**5600**	**1600**	**11200**	**3700**	**3700**	**14900**

O método `AddValue` também pode ser usado para o objeto "Tabela de Referência Cruzada de BD". Isso possibilita inserir dados que não sejam derivados da fonte de dados conectada ao objeto. Se dados forem adicionados desta maneira, eles também serão resumidos junto aos dados da fonte de dados.

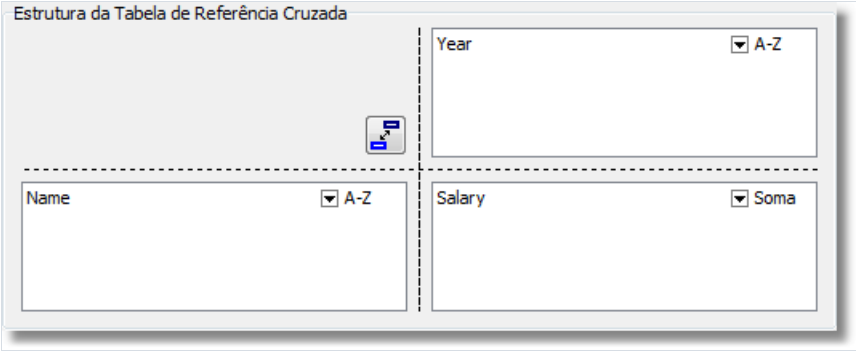
Adicionar objetos externos à tabela

É possivel colocar objetos externos (como linhas, formas e figuras) na tabela de referência cruzada. Pode ser necessário mostrar alguns dados de forma gráfica, por exemplo. Vamos dar uma olhada em um exemplo que usa formas para exibir uma barra de progresso rudimentar:

Salary	Year				
Name	1999	2000	2001	2002	**Grand Total**
Ann	3300	2700	3100	1700	10800
Ben	3900	2100		1800	7800
Catherine	6100	3200			9300
Den		3999	8100		12099
Grand Total	13300	11999	11200	3500	39999

Uma barra vermelho-escuro é exibida se o valor da célula for menor que 100, amarela se for menor que 3000 ou verde ser for maior que 3000.

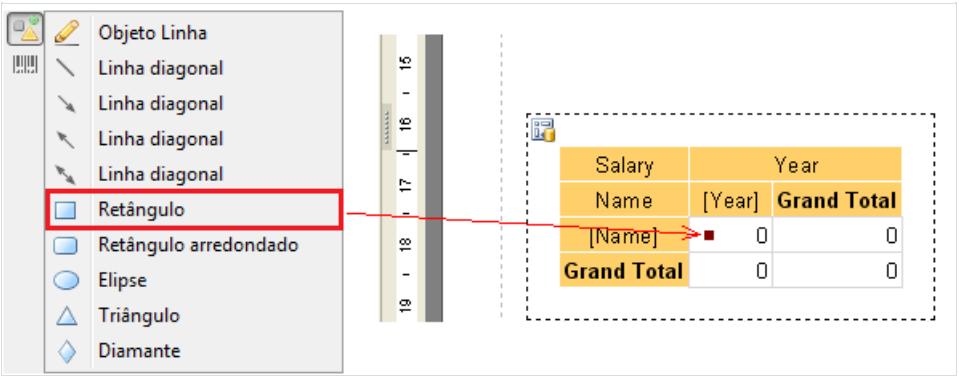
Vamos começar com nosso relatório. Coloque um objeto "Tabela de Referência Cruzada de BD" na página do relatório e configure suas propriedades desta maneira:



Desative a propriedade "Tamanho Automático" e configure a largura das colunas conforme é exibido abaixo:

Salary	Year	
Name	[Year]	**Grand Total**
[Name]	0	0
Grand Total	0	0

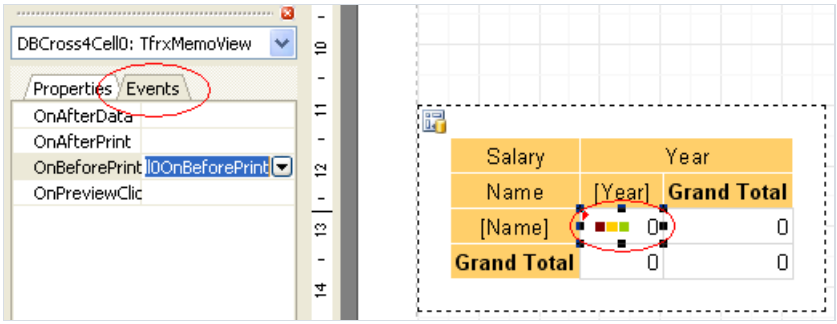
Agora adicione o objeto forma a nossa tabela. Selecione o objeto "Retângulo" na barra de ferramentas de objetos e coloque-o dentro da célula:



Modifique sua altura e largura para 0,2 cm e configure suas propriedades Top e left. Adicione dois retângulos

parecidos.

Agora crie um script que exibirá o número correto de formas coloridas (dependendo do valor da célula). Para fazer isso, selecione a célula e crie um manipulador de evento "OnBeforePrint":



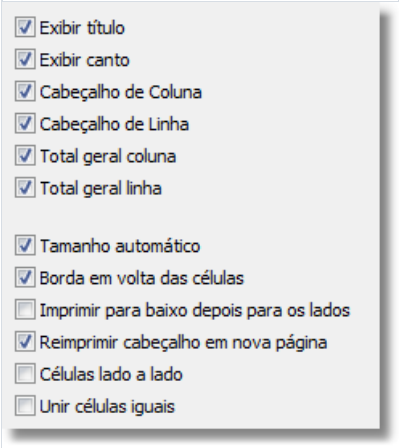
Escreva o código a seguir no manipulador de evento (preste atenção aos nomes das formas, de modo que eles correspondam a seus objetos):

```
procedure DBCross1Cell0OnBeforePrint(Sender: TfrxComponent);
begin
  // 'Value' é o valor atual da célula atual
  if Value < 100 then
  begin
    // primeiro objeto forma
    DBCross1Object1.Color := clMaroon; // vermelho-escuro
    // segundo objeto forma
    DBCross1Object2.Color := clWhite;
    // terceiro objeto forma
    DBCross1Object3.Color := clWhite;
  end
  else if Value < 3000 then
  begin
    DBCross1Object1.Color := $00CCFF;  // amarelo
    DBCross1Object2.Color := $00CCFF;
    DBCross1Object3.Color := clWhite;
  end
  else
  begin
    DBCross1Object1.Color := $00CC98;  // verde
    DBCross1Object2.Color := $00CC98;
    DBCross1Object3.Color := $00CC98;
  end;
end;
```

Isso é tudo. Visualize o relatório, que será parecido àquele que foi mostrado no início desta seção.

Algumas configurações úteis

Vamos dar uma olhada em outras configurações que estão disponíveis no editor de tabela de referência cruzada.



As primeiras seis opções permitem exibir ou ocultar diversos elementos da tabela.

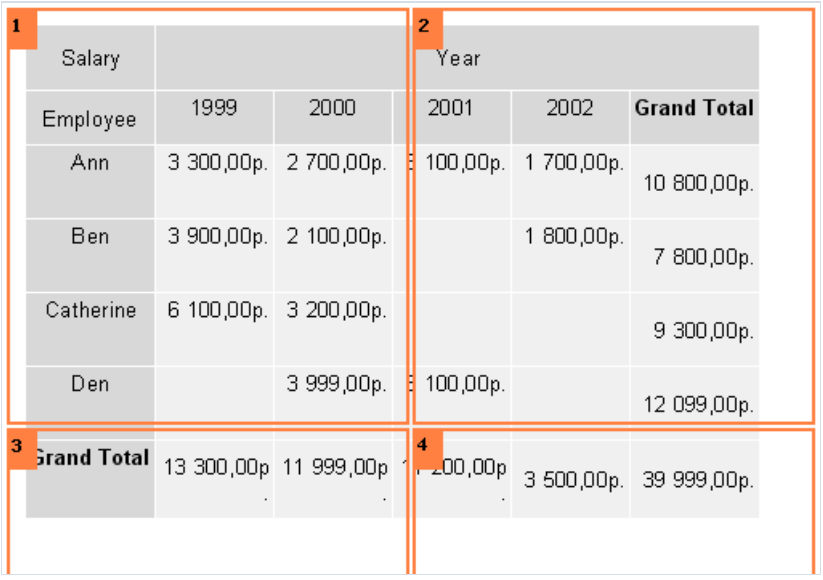
A opção "Tamanho automático" já é bem conhecida. Ela permite configurar a largura e altura da tabela de forma manual.

A opção "Borda em volta das células" permite desenhar um quadro em volta dos elementos da célula. Este é uma tabela de exemplo com uma borda (observe que as células em si não têm bordas):

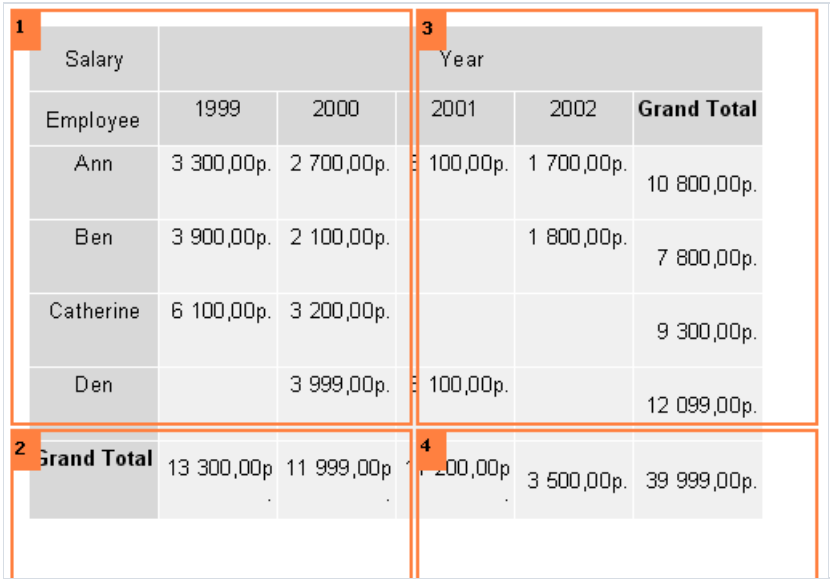
Salary	Year				
Name	1999	2000	2001	2002	**Grand Total**
Ann	3300	2700	3100	1700	10800
Ben	3900	2100		1800	7800
Catherine	6100	3200			9300
Den		3999	8100		12099
Grand Total	13300	11999	11200	3500	39999

A opção "Imprimir para baixo depois para os lados" determina como uma tabela é impressa em várias páginas. Aqui seguem dois exemplos que mostram o funcionamento desta função (observe a ordem das páginas):

1. "Imprimir para baixo depois para os lados" está desabilitado:



2. "Imprimir para baixo depois para os lados" está habilitado:



Salary			Year		
Employee	1999	2000	2001	2002	**Grand Total**
Ann	3 300,00р.	2 700,00р.	100,00р.	1 700,00р.	10 800,00р.
Ben	3 900,00р.	2 100,00р.		1 800,00р.	7 800,00р.
Catherine	6 100,00р.	3 200,00р.			9 300,00р.
Den		3 999,00р.	100,00р.		12 099,00р.
Grand Total	13 300,00р.	11 999,00р.	200,00р.	3 500,00р.	39 999,00р.

A opção "Reimprimir cabeçalhos em nova página" determina se cabeçalhos de tabelas são impressos em cada página nova de visualização.

A opção "Células lado a lado" é usada se houver dois ou mais valores em uma célula da tabela. Ela determina se esses valores da célula são impressos lado a lado ou empilhados um acima do outro (o padrão).

A opção "Unir células iguais" une as células horizontalmente se elas tiverem o mesmo valor:

Days	Year				
Name	1999	2000	2001	2002	**Grand Total**
Ann	3		4	2	12
Ben	4	2		2	8
Catherine	6	3			9
Den		4	7		11
Grand Total	13	12	11	4	40

Outras propriedades também estão disponíveis no inspetor de objetos:

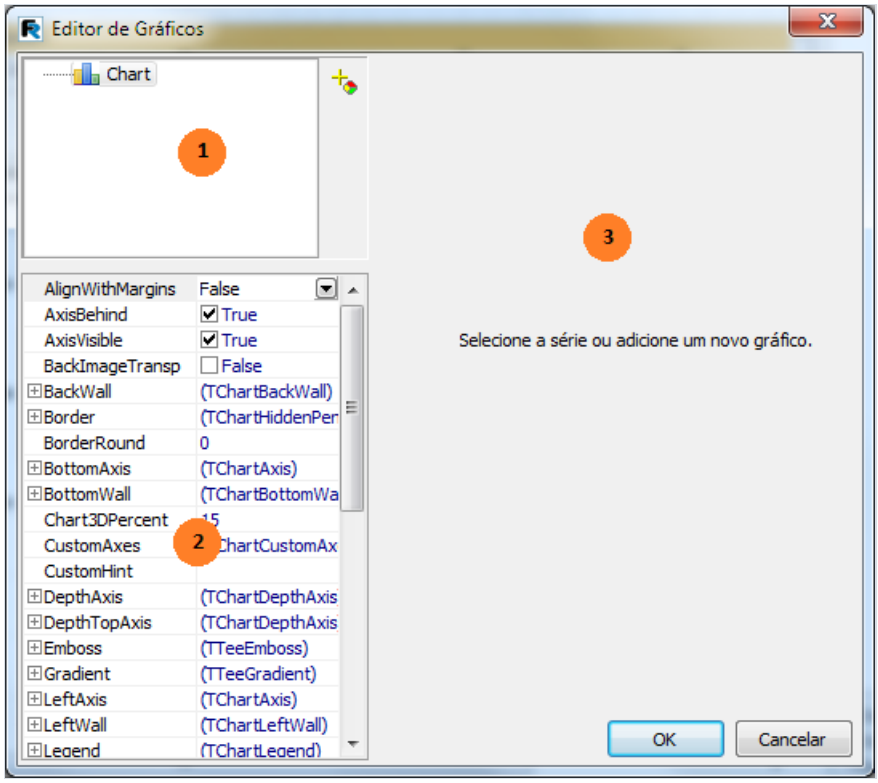
- `AddWidth`, `AddHeight`: adiciona uma quantidade de espaço especificado à largura ou altura da célula. Este número é incluído quando o mecanismo do FastReport calcula o tamanho da célula (a opção "Tamanho automático" deve estar habilitada)
- `NextCross`: um ponteiro para outro objeto tabela de referência cruzada que será impresso ao lado do atual
- `NextCrossGap`: o espaço entre dois objetos tabela de referência cruzada adjacentes

Gráficos

O FastReport pode inserir gráficos em um relatório. Vamos fazer um relatório de exemplo com um gráfico simples. O gráfico irá utilizar a tabela "Country". Esta tabela contém dados sobre países, suas áreas e populações.

```
Name        Area        Population
Argentina   2 777 815   32 300 003
Bolivia     1 098 575    7 300 000
...
```

Adicione um objeto "Gráfico" à página de design do relatório e defina seu tamanho como 18 cm de largura e 8 cm de altura. Abra seu editor de objeto ao clicar duas vezes nele.



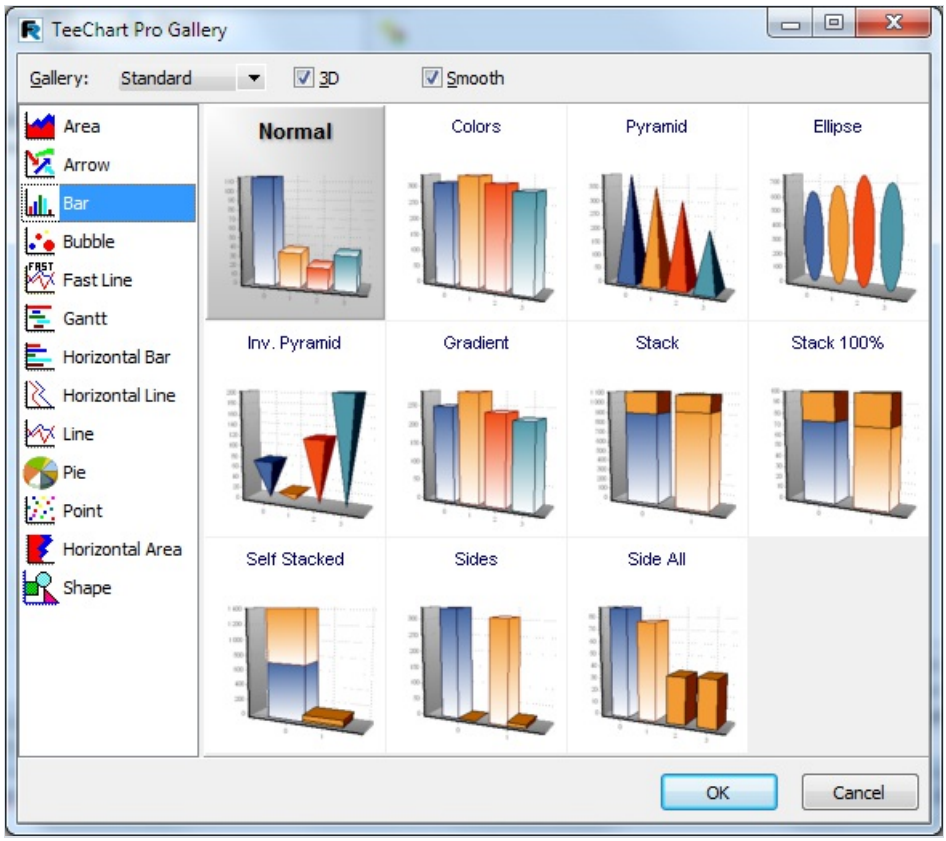
Legenda do editor de gráfico acima:

1 – estrutura do gráfico; um gráfico pode conter uma ou mais séries

2 – inspetor de objetos que exibe as propriedades do elemento selecionado na janela; configure as propriedades do gráfico aqui

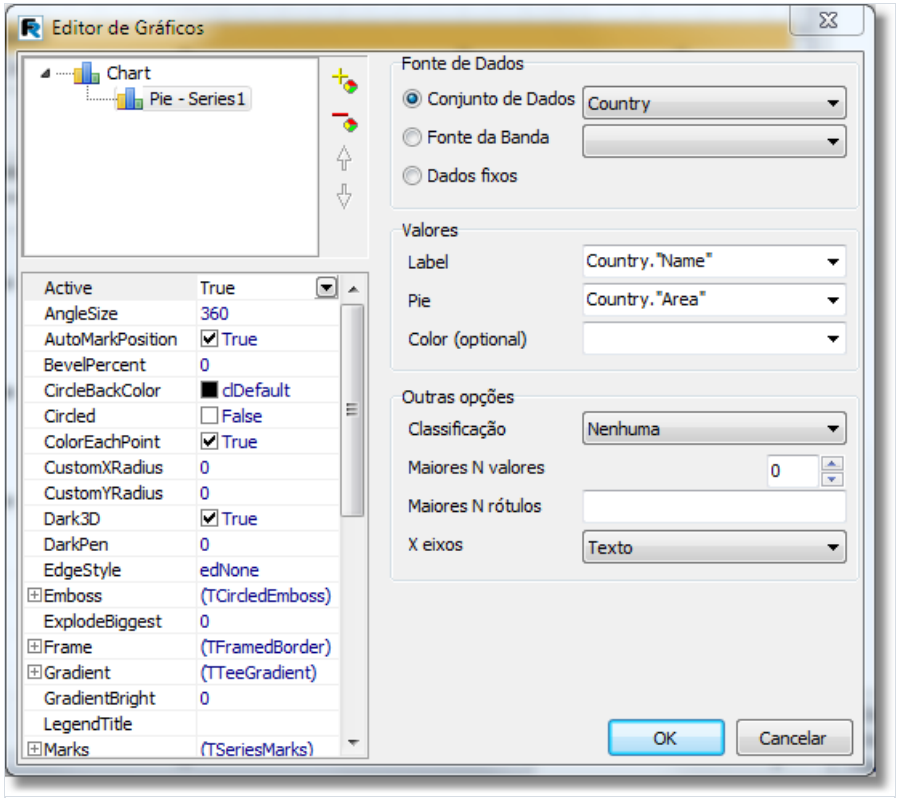
3 – área de opções para a conexão das séries aos dados; é ativada ao selecionar uma série na área 1

Quando for aberto pela primeira vez, o editor de gráfico será exibido como na imagem mostrada acima. A primeira tarefa é adicionar uma ou mais séries ao gráfico (apenas uma série em nosso exemplo). Faça isso ao clicar no botão adicionar e selecionar o gráfico de pizza na Galeria:

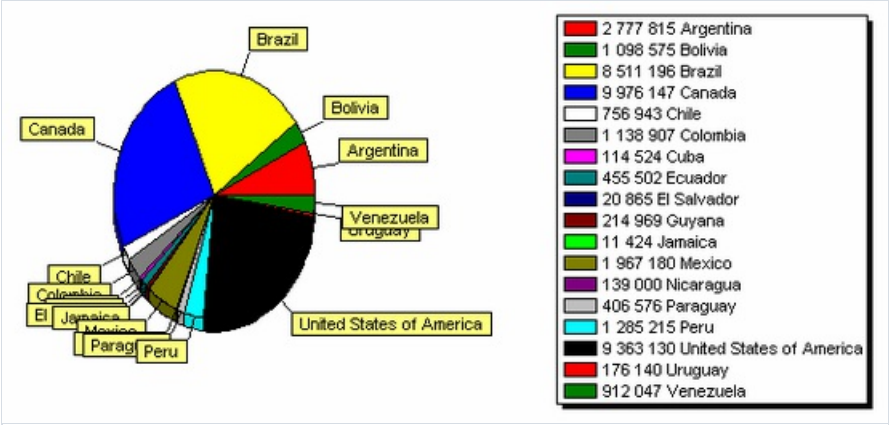


As setas verticais alteram a ordem das séries na lista. Para modificar o nome de uma série, selecione a série e, um segundo depois, clique nele novamente (observe que isso não é um clique duplo).

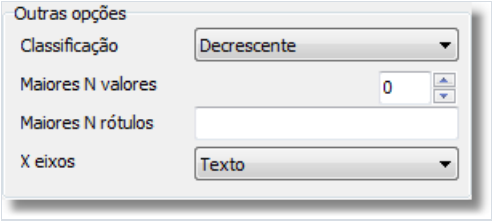
Há muitos tipos diferentes de séries disponíveis. Depois de adicionar uma série, a área de opções 3 é ativada. Nela você especifica quais dados devem ser usados para a plotagem do gráfico. Primeiro vamos escolher o conjunto de dados na lista suspensa "Conjunto de Dados". Então escolha os campos "Rótulo" e "Pizza", usando suas listas suspensas, conforme é mostrado abaixo:



Clique em OK para fechar o editor e visualizar o relatório:



O que pode ser melhorado neste relatório? Seria bom classificar as populações em ordem decrescente. Abra o editor do gráfico novamente, selecione a série na estrutura do gráfico e altere a ordem de classificação de Nenhuma para Descrescente:

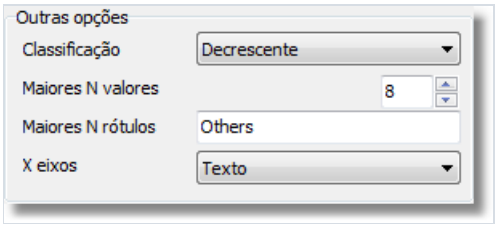


Se visualizarmos o relatório agora, veríamos que os dados na tabela de legendas foram classificados em ordem decrescente.

Limitar o número de valores do gráfico

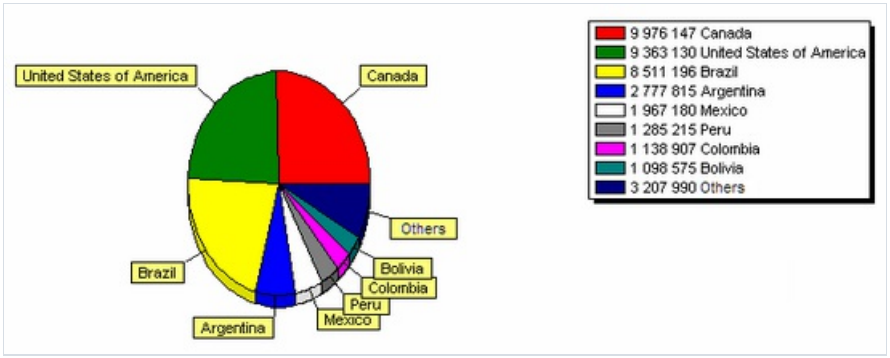
Nosso gráfico está cheio de valores pequenos, o que deixou muitos deles invisíveis. FastReport permite limitar o número de valores exibidos no gráfico. Todos os valores que excederem um número específico são exibidos como um único valor composto, que é criado da soma de todos esses valores.

Em nosso exemplo, o gráfico possui 18 valores, mas na verdade apenas 8 deles podem ser vistos. Abra o editor do gráfico e configure o limite como 8:



Se "Maiores N valores" for configurado como zero, não haverá nenhuma limitação. É necessário inserir um nome em "Maiores N rótulos" para que os valores agregados sejam identificados corretamente na lista de legendas. O modo de classificação é irrelevante, já que os valores sempre serão classificados como decrescentes como padrão.

Este será o aspecto do relatório:



Algumas configurações úteis

Vamos dar uma olhada em algumas configurações úteis para controlar a aparência do gráfico. Estas configurações podem ser acessadas somente no inspetor de objetos do editor do gráfico.

Estas propriedades básicas estão disponíveis quando 'Chart' for selecionado na Estrutura do Gráfico:

- `Gradient`: as configurações do preenchimento de plano de fundo como gradiente. Habilite a propriedade "Gradient.Visible" para exibir gradientes
- `Legend`: as configurações da Lista de Legendas. A Lista pode ser ocultada com a propriedade `Legend.Visible`, e sua posição é configurada através da propriedade `Legend.Alignment`.

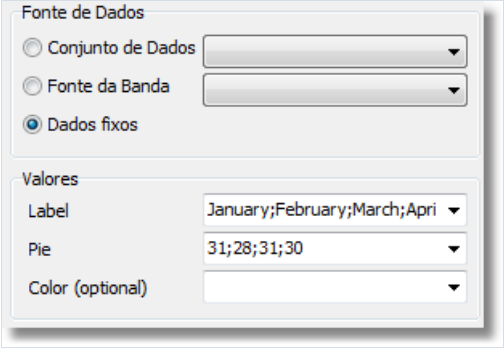
As propriedades a seguir estão disponíveis quando uma série foi selecionada:

- `ColorEachPoint`: dar uma cor diferente a cada valor
- `ExplodeBiggest`: deslocar o valor mais alto (funciona somente em gráficos de pizza)
- `Marks`: as configurações da aparência dos rótulos do gráfico
- `ValueFormat`: a linha de formatação dos valores

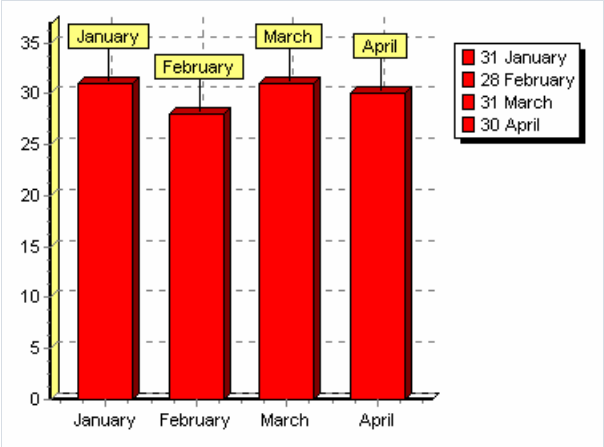
Gráfico com valores inseridos manualmente

No exemplo anterior criamos um gráfico usando dados de uma tabela de BD. Também é possível criar um gráfico com dados inseridos manualmente. Este método pode ser conveniente se o gráfico for pequeno.

Vamos demonstrar como isso funciona com um exemplo simples. Coloque um gráfico na página de design do relatório e abrea o editor do gráfico. Adicione uma série de tipo "Gráfico de barras" e configure as seguintes propriedades, usando um ponto e vírgula para separar valores individuais:



O gráfico resultante é:



Criação de gráficos com um script

Vamos ver como criar o gráfico anterior usando um script. Deixe os campos Rótulo e Y vazios no editor de gráficos. Insira o código a seguir no script do relatório:

PascalScript:

```
begin
  Chart1.SeriesData[0].XSource := 'Jan;Feb;Mar;Apr';
  Chart1.SeriesData[0].YSource := '31;28;31;30';
end.
```

C++Script:

```
{
  Chart1.SeriesData[0].XSource = "Jan;Feb;Mar;Apr";
  Chart1.SeriesData[0].YSource = "31;28;31;30";
}
```

Neste caso, SeriesData[0] nos permite configurar parâmetros para a primeira série do gráfico. Se o gráfico tiver mais de uma série, refira-se a elas com SeriesData[1], etc.

Imprimir um gráfico criado em Delphi

Se você já criou um gráfico em seu código Delphi e deseja imprimí-lo no relatório, é necessário usar o objeto "Figura" na barra de ferramentas de objetos do FastReport. Coloque o objeto no lugar requerido na página de design do relatório e crie o seguinte manipulador de eventos "TfrxReport.OnBeforePrint" no aplicativo Delphi:

```
procedure TForm1.frxReport1BeforePrint(Sender: TfrxReportComponent);
begin
  if Sender.Name = 'Picture1' then
    TfrxPictureView(Sender).Picture.Assign(
      Chart1.TeeCreateMetafile(False,
        Rect(0, 0, Round(Sender.Width), Round(Sender.Height))));
end;
```

no qual Figura1 é o nome do objeto "Figura" e Gráfico1 é seu gráfico Delphi.

Observação: Quando código é atribuído aos manipuladores de eventos do componente `TfrxReport` em um aplicativo Delphi, é necessário visualizar o relatório ao executar o aplicativo Delphi compilado. Não é possível visualizar o relatório dentro do designer de relatórios do FastReport.

Relatórios interativos

É possível tornar interativo um relatório criado no FastReport. Isso significa que ele irá reagir às ações do usuário na janela de visualização. É possível usar as interações a seguir:

· ao clicar no objeto do relatório, um tipo de operação é executada. Por exemplo, é possível executar um relatório detalhado e exibí-lo em uma janela separada;

· a janela de visualização pode exibir a estrutura de tópicos do relatório, que pode ser usada para navegar o relatório.

Hiperlink

Quase todos os objetos do relatório possuem a propriedade `Hyperlink`. Ao usar esta propriedade, é possível definir a reação de um objeto ao clique do mouse na janela de visualização.

Ao clicar em um objeto desse tipo, uma das ações a seguir pode ocorrer:

· navegar até a URL;

· enviar um e-mail;

· executar qualquer comando de sistema;

· navegar até a página do relatório que possui o número indicado;

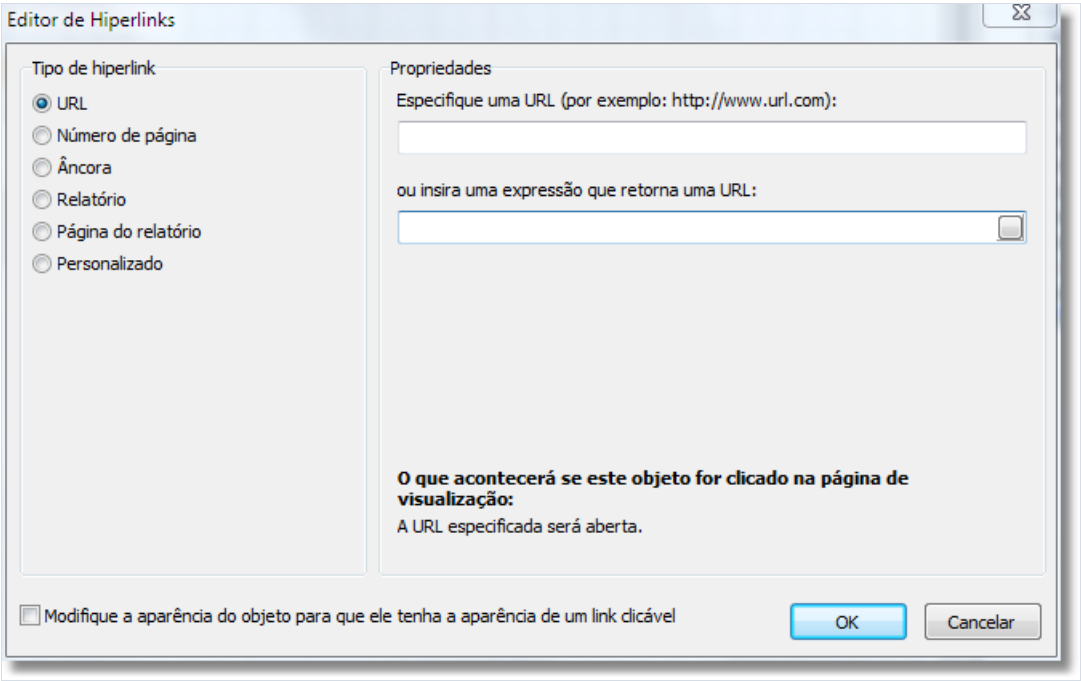
· navegar até a âncora que foi adicionada em um script;

· executar um relatório detalhado em uma janela de visualização separada;

· ação padrão, definida em um script.

Configuração de um hiperlink

Para configurar um hiperlink, selecione o objeto que deseja tornar interativo e clique nele com o botão direito do mouse. No menu de contexto, selecione o item "Hiperlink...". O editor de hiperlinks será aberto:



Escolha o tipo de hiperlink ao selecionar o botão de opção no lado esquerdo da janela. Após fazer isso, você pode clicar na caixa de seleção "Modificar a aparência do objeto..." na parte inferior da janela. A aparência do objeto será alterada da seguinte maneira:

· o texto se tornará azul e será sublinhado;

· será definido um cursor do tipo mão.

Em alguns casos o hiperlink precisa ser exibido na janela de visualização, mas não há necessidade de imprimí-lo. É fácil fazer isso ao configurar a propriedade `Visibility`. Isso pode ser fieto na janela "Inspetor de Objetos".

Link para a URL

Ao usar este tipo de link, é possível:

· navegar até o endereço da Internet fornecido;

· executar alguns comandos do sistema, por exemplo "mailto:" para enviar um e-mail.

O valor do link pode ser indicado com dois métodos:

· indique o valor diretamente, por exemplo "http://www.fast-report.com";

· indique uma expressão que retorne o valor do link. Esta expressão será calculada ao executar o relatório.

Link para o número da página

Ao usar este tipo de link, é possível organizar a navegação nas páginas de um relatório preparado. Na maioria das vezes é criado um link para a primeira página. Para fazer isso, indique o número da página (1 neste caso) como o valor do link.

É possível indicar o número da página com dois métodos:

· indicar o número diretamente, por exemplo 1;

· indicar uma expressão que retorne o número da página. Esta expressão será calculada ao executar o relatório.

Link para uma âncora

Ao usar este tipo de link, é possível navegar até uma âncora que foi adicionada em um script. Uma âncora possui um nome e uma posição específica em um relatório (número de página e posição na página). Ao saltar até uma âncora usando seu nome, você será levado até a posição indicada.

Para usar este tipo de link, primeiro é necessário definir uma âncora. Para fazer isso, use um script e o método `Engine.AddAnchor` (consulte o tópico Âncoras para maiores detalhes). Agora é possível indicar seu nome na janela de configurações do hiperlink. Isso pode ser feito por dois métodos:

· indicar o nome da âncora diretamente;

· indicar uma expressão que retorne o nome da âncora. Por exemplo, isso pode ser a coluna de dados. A expressão será calculada quando o relatório for executado.

Link para um relatório detalhado

Ao usar este tipo de link, é possível executar outro relatório e exibí-lo em uma janela de visualização separada.

É necessário configurar os seguintes parâmetros para este tipo de hiperlink:

· o nome do relatório detalhado;

· o nome da variável do relatório, que ficará com o valor do hiperlink;

· o valor do hiperlink.

Quando o link for clicado, acontecerá o seguinte:

· o relatório indicado será carregado;

· a variável do relatório será configurada com o valor do hiperlink;

· o relatório será criado e executado em uma janela de visualização separada.

O valor da variável do relatório pode ser indicado por dois métodos:

· indicar o valor diretamente;

· indicar uma expressão que retorne o valor. Esta expressão será calculada quando o relatório for executado.

Link a uma página detalhada

Este tipo de link funciona da mesma maneira, exceto que outra página no relatório atual é usada como um relatório detalhado. Para fazer isso, seu relatório deve conter pelo menos duas páginas: uma com o relatório principal e outra com o relatório detalhado.

É necessário configurar os parâmetros a seguir para este tipo de hiperlink:

· nome da página no relatório;

· nome da variável do relatório, que ficará com o valor do hiperlink;

· o valor do hiperlink.

Ao clicar no link, acontecerá o seguinte:

· a variável do relatório será configurada com o valor do hiperlink;

· a página do relatório indicada será criada e exibida em uma janela de visualização separada.

O valor da variável do relatório pode ser indicado com dois métodos:

· indicar o valor diretamente;

· indicar uma expressão que retorne o valor. A expressão será calculada quando o relatório for executado.

Ao escollher uma página do relatório, sua propriedade `Visible` é redefinida como False. Isso significa que ao criar o relatório principal, esta página é ignorada.

Link personalizado

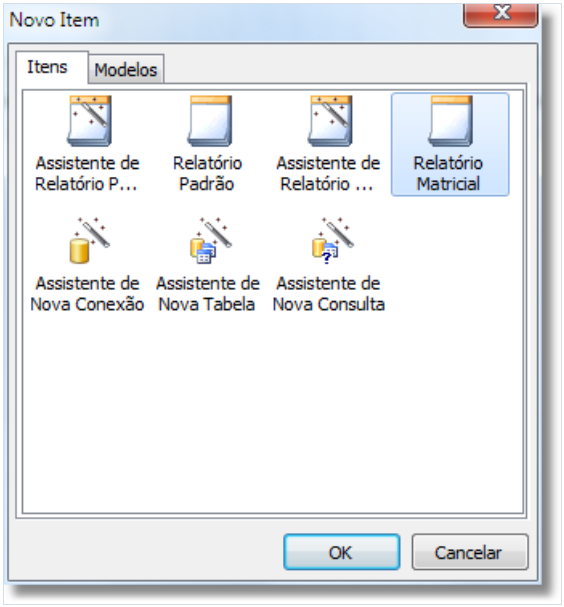
Ao usar este tipo de link, é possível definir uma reação personalizada ao clicar com o mouse. Para isso, utilize o manipulador de evento "OnClick" do objeto.

Relatórios matriciais

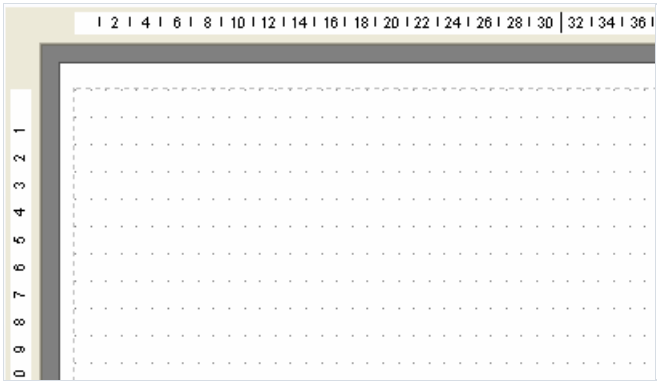
Anteriormente nós abordamos relatórios que foram projetado para serem impressos em impressoras modernas padrão (stylus, laser, etc.). Um relatório enviado a uma impressora matricial será impresso muito lentamente. FastReport permite a criação de relatórios especiais destinados a impressoras matriciais, onde somente os símbolos padrão das fontes são impressos e nenhum elemento gráfico; isso resulta em uma velocidade de impressão mais rápida.

Vamos ver como criar um relatório do tipo "Lista" projetado para a impressão em impressoras matriciais. Anteriormente nós criamos este tipo de relatório, consulte o relatório "Lista de clientes". Usaremos os mesmos dados no relatório.

Abra o designer de relatórios. Selecione "Arquivo > Novo..." para abrir a caixa de diálogo do assistente de relatórios e selecione o item "Relatório Matricial":



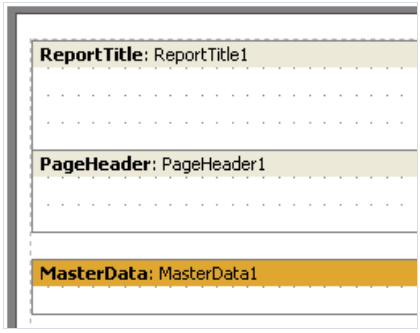
Ao clicar em OK, é exibida uma página de design vazia, marcada para uma fonte matricial:



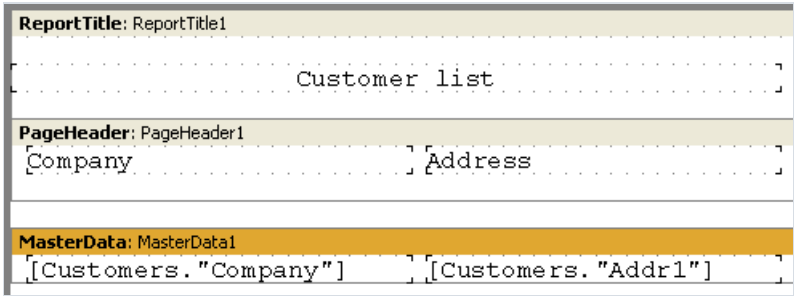
A barra de ferramentas de objetos também é alterada para exibir os objetos disponíveis para a impressão matricial, que são "Banda", "Texto", "Linha", "Comando Matricial", "Sub-relatório", "Tabela de Referência Cruzada" e "Tabela de Referência Cruzada de BD". Outros objetos não podem ser usados em uma impressora matricial.



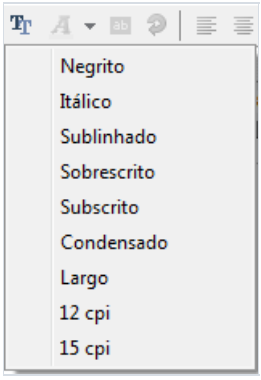
Coloque as bandas "Título do Relatório", "Cabeçalho da Página" e "Dados Mestre" na página do relatório:



Coloque objetos "Texto" nas bandas conforme é mostrado a seguir:



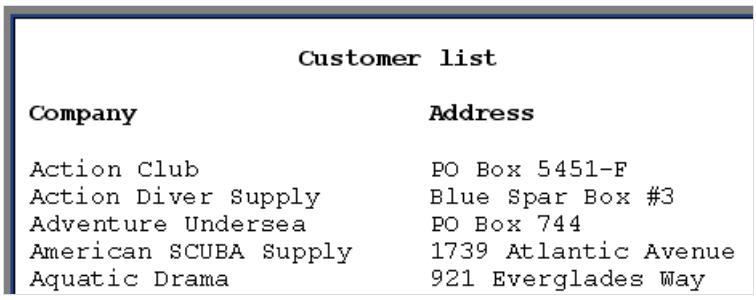
A colocação de objetos matriciais é similar à colocação em relatórios normais. No entanto, estes objetos são limitados em sua posição e aparência. Os objetos se ajustam à grade, o tamanho da fonte (altura) não pode ser alterado e eles não podem ser coloridos. Porém, alguns atributos de fonte podem ser modificados ao selecionar o objeto "Texto" e clicar no botão "Tt" na barra de ferramentas:



Estes atributos de fonte são específicos para a impressão matricial. A página do relatório e todos os objetos matriciais, exceto bandas, possuem estes atributos.

Observação: no designer e na visualização, somente os atributos "Negrito", "Itálico" e "Sublinhado" são implementados na tela. O conjunto inteiro de atributos é implementado somente na impressão.

Vamos modificar nosso relatório ao usar o estilo "Negrito" nos cabeçalhos. O relatório está concluído e está pronto para ser visualizado:



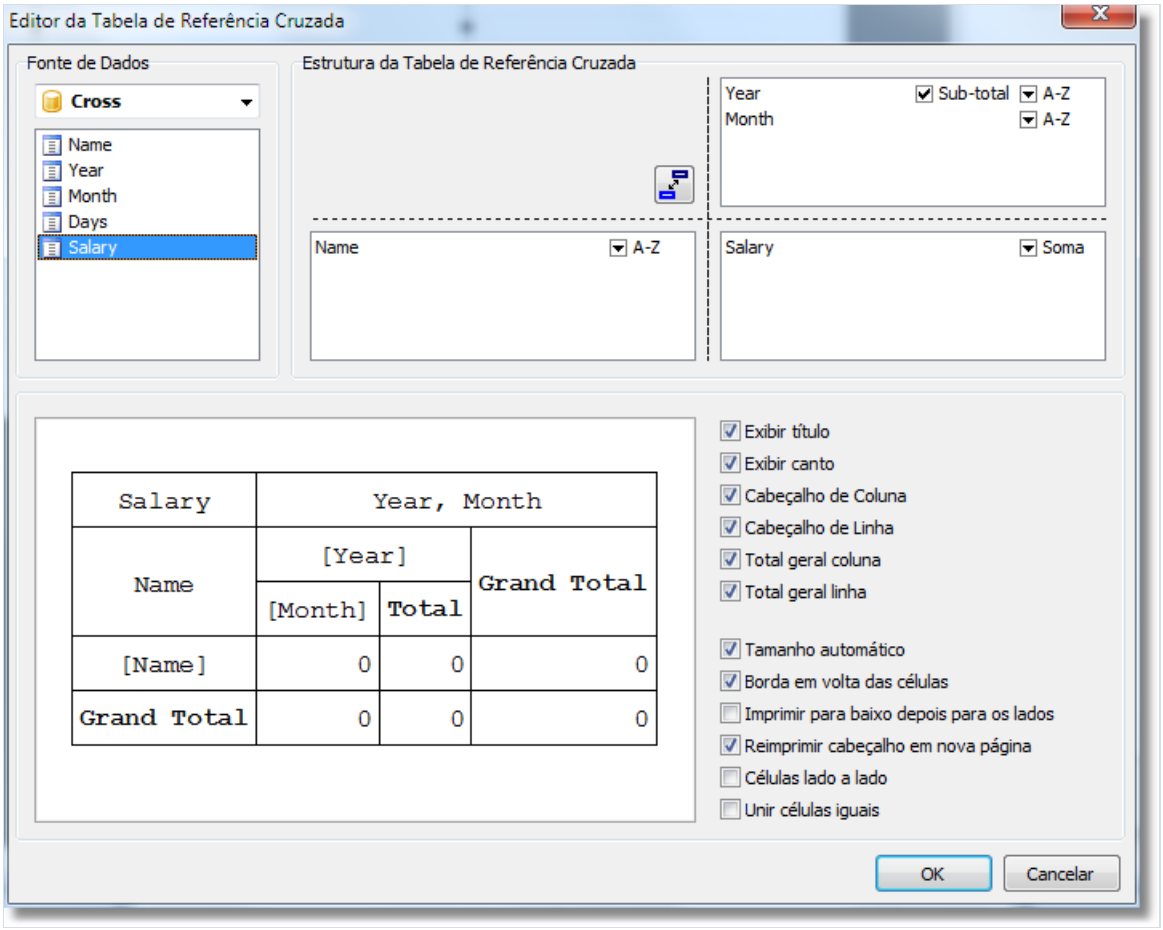
Customer list

Company	Address
Action Club	PO Box 5451-F
Action Diver Supply	Blue Spar Box #3
Adventure Undersea	PO Box 744
American SCUBA Supply	1739 Atlantic Avenue
Aquatic Drama	921 Everglades Way

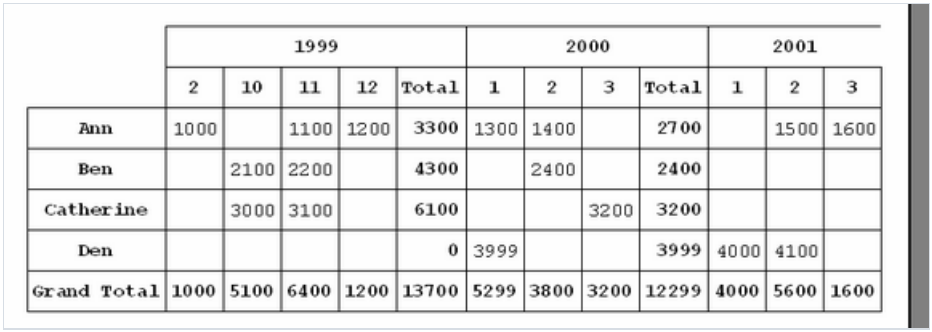
Tabelas de referência cruzada em relatórios matriciais

A gama de objetos utilizáveis em relatórios matriciais é restrita àqueles que podem ser exibidos de forma textual. Entre estes objetos se econtram os objetos "Tabela de Referência Cruzada". Vamos dar uma olhada em um relatório simples de tabela de referência cruzada, parecido com aquele que foi criado anteriormente na seção "Tabelas com cabeçalhos compostos".

Crie um relatório matricial como na seção anterior, escolhendo um "Relatório Matricial". Coloque um objeto "Tabela de Referência Cruzada de BD" na página do relatório e abra seu editor:



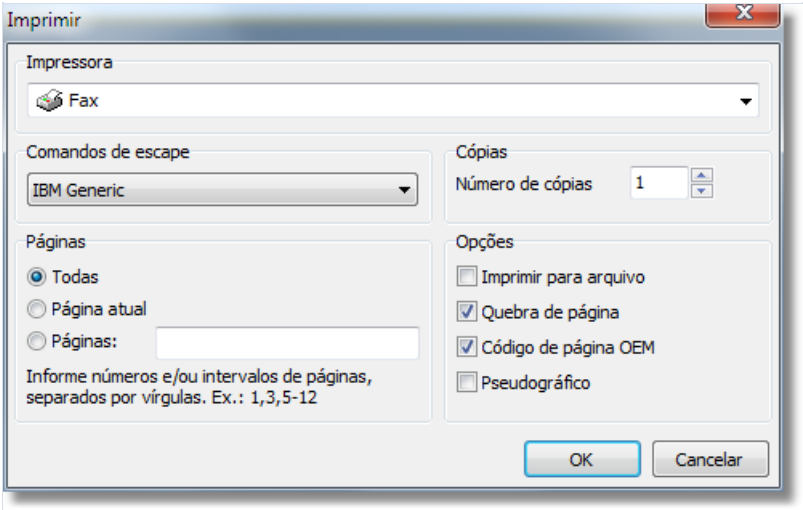
O editor da tabela de referência cruzada exibe a estrutura da tabela de saída no modo matricial. Os atributos da fonte da célula podem ser configurados usando o botão "Tt" na barra de ferramentas. Quanto aos demais aspectos, os objetos tabela de referência cruzada se comportam como foi descrito anteriormente. O relatório visualizado têm a seguinte aparência:



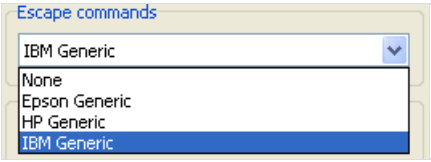
	1999					2000				2001		
	2	10	11	12	Total	1	2	3	Total	1	2	3
Ann	1000		1100	1200	3300	1300	1400		2700		1500	1600
Ben		2100	2200		4300		2400		2400			
Catherine		3000	3100		6100			3200	3200			
Den					0	3999			3999	4000	4100	
Grand Total	1000	5100	6400	1200	13700	5299	3800	3200	12299	4000	5600	1600

Impressão de um relatório matricial

A seguinte caixa de diálogo de impressão é usada ao imprimir um relatório matricial:



A caixa de diálogo de impressão é parecida com a normal, mas possui elementos específicos para impressoras matriciais. O sistema de comandos da impressora deve ser escolhido na lista a seguir antes da impressão (os comandos ESC):

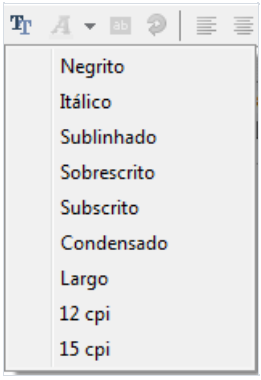


Também há outras opção para a impressão matricial:

- Imprimir para arquivo: controla se o fluxo de impressão é enviado a um arquivo no disco rígido. Se for habilitada, a caixa de diálogo "Salvar Como..." é exibida
- Quebra de página: controla se um comando "Quebra de página" é enviado ao alcançar o final da página. Se for desabilitada, a impressão não é interrompida em folhas contínuas
- Página de código OEM: controla a conversão de símbolos
- Pseudográfico: especifica como desenhar linhas verticais e horizontais. Se for desabilitada as linhas são desenhadas usando os símbolos -, | e +

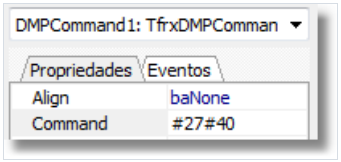
Objeto "Comando"

Conforme foi descrito antes, é possível configurar os seguintes atributos em objetos de relatórios matriciais:



Este é um conjunto padrão que é aceito por todos os modelos de impressoras matriciais. Alguns modelos de impressoras podem suportar outros comandos que não estão presentes no conjunto padrão, por exemplo a impressão na resolução de 20 caracteres por polegada. Para poder enviar estes atributos adicionais à impressora, use o objeto "Comando Matricial" no relatório.

Coloque o objeto "Comando Matricial" no local correto na página do relatório, antes de objetos que usem atributos fora do padrão (ou seja, no canto superior esquerdo da página). Para configurar um comando, edite a propriedade Command do objeto no inspetor de objetos:

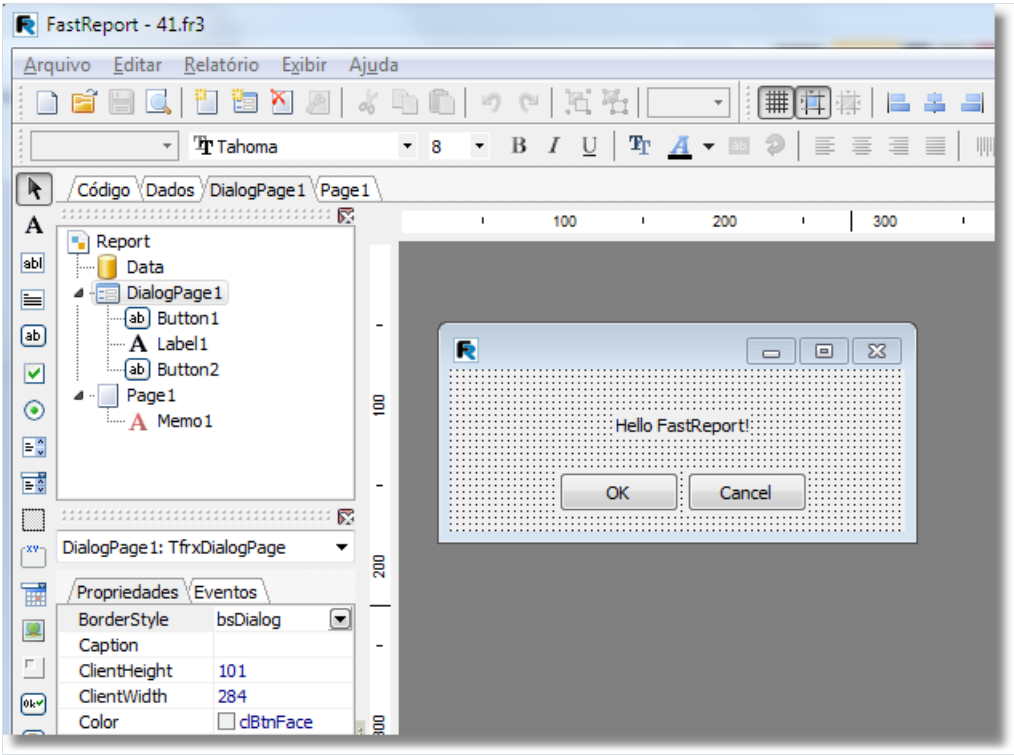


A propriedade pode ser configurada com um dos seguintes formatos:

decimal (por exemplo, `#27#40`) ou hexadecimal (por exemplo, `1B28`).

Formulários de diálogo

Além das páginas de design do relatório normais, é possível usar formulários de diálogo em um relatório. Formulários de diálogo são criados no designer de relatório normal com o botão na barra de ferramentas do designer. O botão adiciona uma nova guia de página de diálogo ao relatório. Ao abrir a guia da página de diálogo, o espaço de trabalho do designer é alterado para exibir o formulário, e a barra de ferramentas de objetos é alterada para mostrar os objetos de controle que podem ser colocados no formulário:



Controles

Elemento	Nome	Descrição
	TfrxLabelControl	controle para exibir textos fixos no formulário de diálogo
	TfrxEditControl	controle para inserir texto usando o teclado
	TfrxMemoControl	controle para inserir várias linhas de texto com o teclado
	TfrxButtonControl	controle que representa um botão
	TfrxCheckBoxControl	controle que representa um sinalizador, que possui dois estados – habilitado ou desabilitado, além de um rótulo adjacente
	TfrxRadioButtonControl	controle que representa um botão de opção. Deve haver mais de um controle de botão de opção no formulário
	TfrxListBoxControl	controle que representa uma lista de itens, da qual um item pode ser selecionado
	TfrxComboBoxControl	controle que representa uma lista suspensa de itens, na qual é possível selecionar um item
	TfrxDateEditControl	controle que representa um campo com um calendário suspenso para a entrada de datas
	TfrxGroupBoxControl	controle que representa uma caixa com rótulo que pode conter outros objetos
	TfrxPanelControl	controle que representa um painel que pode conter outros objetos
	TfrxBitBtnControl	controle que representa um botão que contém uma figura
	TfrxSpeedButtonControl	controle que representa um botão de velocidade que contém uma figura
	TfrxMaskEditControl	controle para inserir texto usando o teclado e usando um modelo
	TfrxCheckListBoxControl	controle que representa uma lista de itens que são selecionados por uma caixa de seleção
	TfrxBevelControl	controle usado para exibir linhas e bordas no formulário
	TfrxImageControl	controle que representa uma figura no formato "BMP", "ICO", "WMF" ou "EMF"

Relatório "Olá, Mundo!"

Neste exemplo criaremos um relatório que exibe uma janela de saudação usando o formulário de diálogo, antes de criar a visualização. Crie um relatório novo e adicione um formulário de diálogo ao relatório. Coloque os objetos `TfrxLabelControl` e `TfrxButtonControl` no formulário:



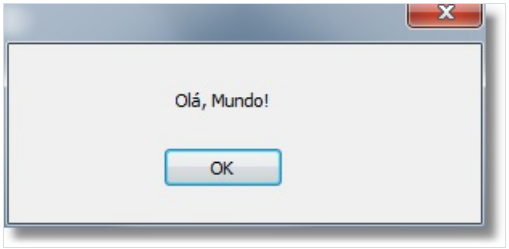
Configure as propriedades dos objetos:

```
TfrxLabelControl:
Caption = 'Olá, Mundo!'

TfrxButtonControl:
Caption = 'OK'
Default = True
ModalResult = mrOk
```

Configure a propriedade `BorderStyle` = bsDialog para o formulário.

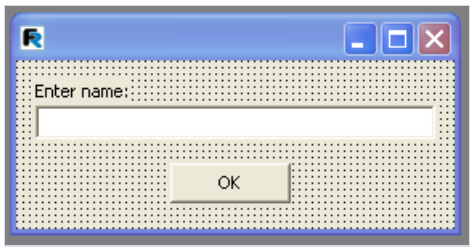
Quando o design do formulário de diálogo estiver concluído, volte para a guia Página1 do design do relatório e coloque um objeto "Texto" que contenha a saudação na página. Visualize o relatório e você verá o formulário de diálogo:



Após clicar no botão OK, a caixa de diálogo é fechada e o relatório é construído e exibido. Se a caixa de diálogo for fechada ao clicar no botão "X" do menu do sistema, o relatório não será criado. O FastReport funciona deste modo: se houver formulários de diálogo em um relatório, o relatório somente é construído depois que todas as caixas de diálogo terem sido fechadas com `ModalResult` – mrOk, ou seja, neste exemplo ao clicar no botão OK. É por isso que a propriedade `ModalResult` do botão tem que ser configurada como "mrOk".

Inserir parâmetros e transferi-los para um relatório

Vamos tornar este exemplo mais complicado para demonstrar como transferir valores inseridos na caixa de diálogo do relatório. Modifique a caixa de diálogo conforme é mostrado aqui:



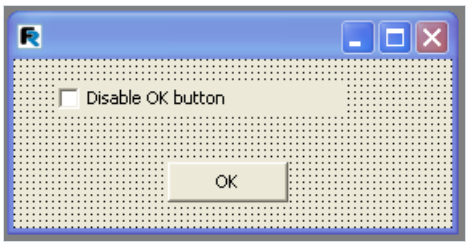
Coloque um objeto "Texto" que contenha este texto na Página1 do relatório, e configure `AutoWidth` como True:

```
Você inseriu: [Edit1.Text]
```

Visualize o relatório e certifique-se de que o texto digitado nele é exibido com êxito no relatório. É possível acessar outros objetos na caixa de diálogo de maneira parecida. Já que cada objeto possui um nome único em todo o relatório, ele pode ser usado em qualquer lugar dentro do mesmo relatório.

Interação com controles

Ao usar um script, é possível incorporar lógica complexa na operação de uma caixa de diálogo. Vamos ilustrar isso com um exemplo simples. Modifique o formulário desta maneira:



Clique duas vezes no objeto "Caixa de Seleção" para criar um manipulador de evento "OnClick", e insira o script a seguir:

PascalScript:

```
procedure CheckBox1OnClick(Sender: TfrxComponent);
begin
  Button1.Enabled := not CheckBox1.Checked;
end;
```

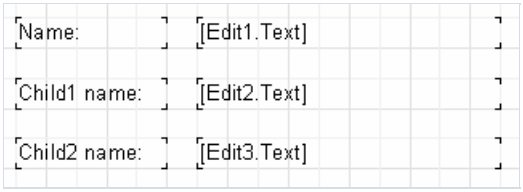
C++ Script:

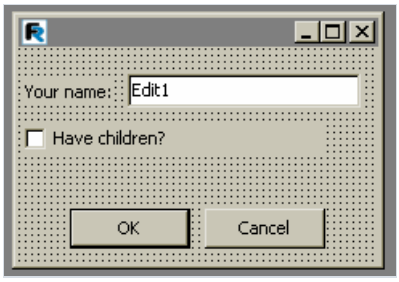
```
void CheckBox1OnClick(TfrxComponent Sender)
{
  Button1.Enabled = !CheckBox1.Checked;
}
```

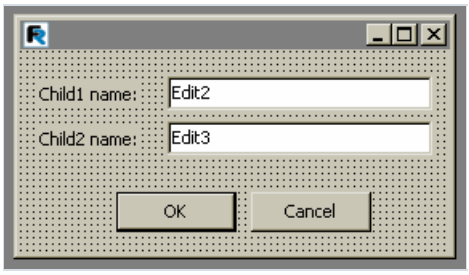
Ao executar o relatório, o botão responde ao estado da caixa de seleção.

Diversos formulários de diálogo

Vamos ver como funciona um relatório com duas caixas de diálogo. Crie um relatório com duas caixas de diálogo e uma página de design:







Configure a propriedade `ModalResult` dos botões OK e Cancelar como mrOk e mrCancel, respectivamente. Agora execute o relatório. Primeiro o usuário deverá responder perguntas da primeira caixa de diálogo (nome, se tem filhos) e após clicar em OK, ele deverá responder as perguntas da segunda caixa de diálogo (nomes dos filhos). Após clicar em OK na segunda caixa de diálogo, o relatório será criado. O FastReport lida com várias caixas de diálogo da seguinte maneira: as caixas de diálogo são exibidas na ordem de criação; cada caixa de diálogo é exibida somente após fechar a anterior com `ModalResult` = mrOk (neste exemplo, ao clicar no botão OK). Se alguma caixa de diálogo for fechada com o botão Cancelar ou o botão "X" do menu do sistema, o relatório não é criado.

Gerenciar formulários de diálogo

No exemplo anterior, a segunda caixa de diálogo é exibida independentemente do fato se "Tem filhos" foi marcada. Vamos mostrar como ocultar a segunda caixa de diálogo se esta caixa de seleção não for marcada. Crie um manipulador de evento "OnClick" para o botão OK na primeira caixa de diálogo (clique duas vezes no botão para criar o manipulador):

PascalScript:

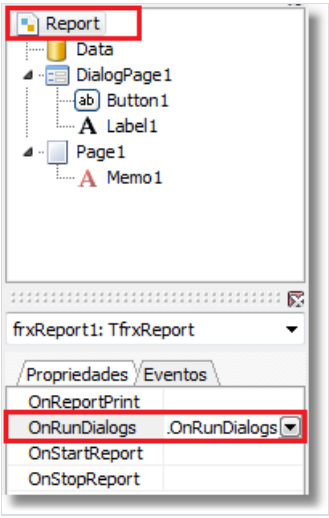
```
procedure Button1OnClick(Sender: TfrxComponent);
begin
  DialogPage2.Visible := CheckBox1.Checked;
end;
```

C++Script:

```
void Button1OnClick(TfrxComponent Sender)
{
  DialogPage2.Visible = CheckBox1.Checked;
}
```

Este código oculta a segunda caixa de diálogo (DialogPage2) se a caixa não for marcada. Visualize o relatório para ver se ele funciona corretamente.

Outra maneira de gerenciar formulários e usar o evento do relatório "OnRunDialogs". Para criar este manipulador de evento, selecione o objeto Report na árvore do relatório ou no inspetor de objetos e abra a guia "Eventos" no inspetor de objetos. Clique duas vezes no evento "OnRunDialogs" para criar um manipulador:



Escreva o código a seguir no manipulador:

PascalScript:

```
procedure frxReport1OnRunDialogs(var Result: Boolean);
begin
  Result := DialogPage1.ShowModal = mrOk;
  if Result then
  begin
    if CheckBox1.Checked then
      Result := DialogPage2.ShowModal = mrOk;
  end;
end;
```

C++Script:

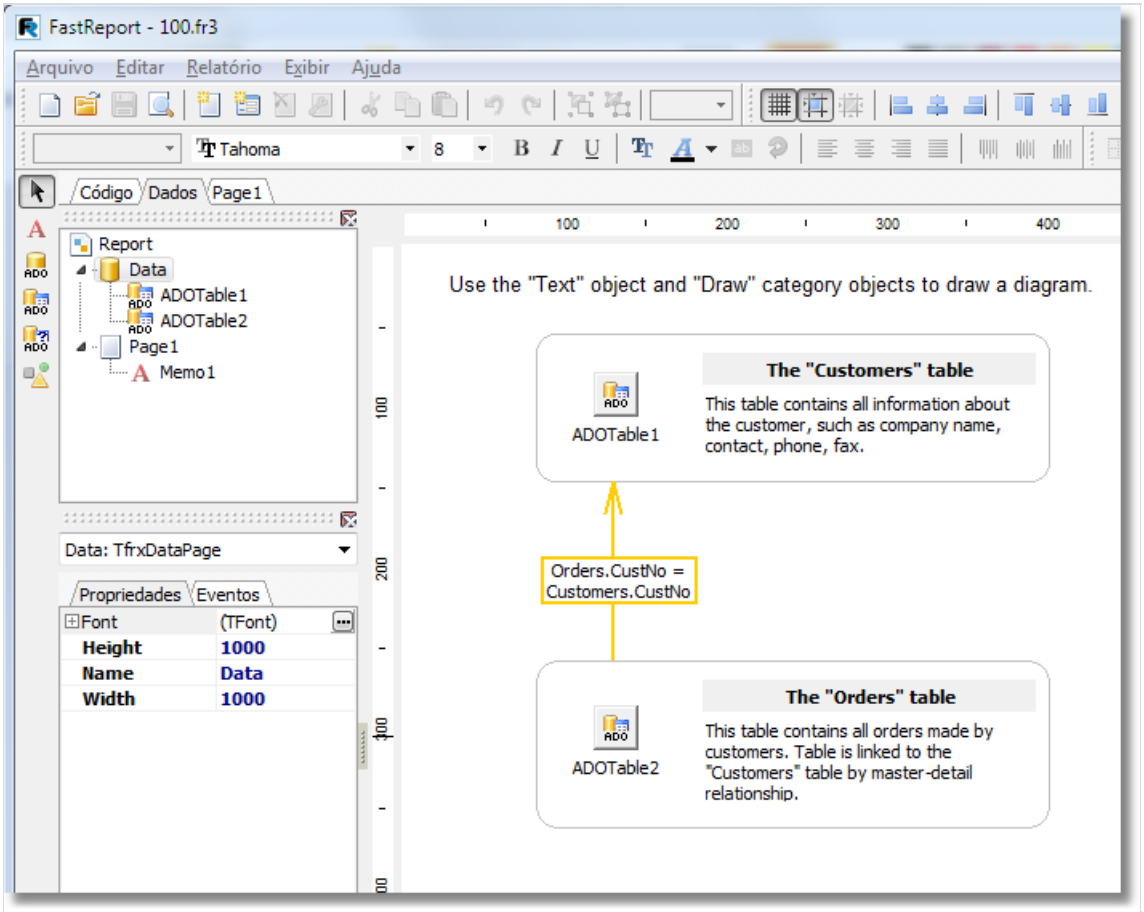
```
void frxReport1OnRunDialogs(bool &Result);
{
  Result = DialogPage1.ShowModal == mrOk;
  if (Result)
  {
    if (CheckBox1.Checked)
      Result = DialogPage2.ShowModal == mrOk;
  }
}
```

O manipulador funciona deste modo: a primeira caixa de diálogo é exibida; se ela for fechada com o botão OK, o estado de CheckBox1 é avaliado; se o estador for Checked, então a segunda caixa de diálogo é exibida; se a segunda caixa de diálogo for fechada com o botão OK, Result é configurado como True. Se o manipulador retornar Result = True, então a visualização é criada; se Result = False, a execução do relatório é parada e nenhuma visualização é criada.

Componentes de acesso a dados

Normalmente, a maioria dos relatórios são baseados em dados que têm origem em um BD. Delphi fornece uma variedade de componentes para vincular dados de BD, e FastReport utiliza estas conexões. Aqui iremos discutir a utilização dos componentes `TTable` e `TQuery` como fontes de dados para relatórios, mas em geral qualquer descendente de `TDataSet` pode ser utilizado.

Além dos componentes de acesso a dados de projetos Delphi, como `TfrxDBDataset` que usamos em nossos exemplos até agora, há diversos componentes de BD específicos no FastReport que podem ser ussados em relatórios, e seu uso depende das escolhas feitas durante a instalação. Os princípios do acesso aos dados no FastReport são praticamente os mesmos do ambiente Delphi. Assim como em Delphi, componentes são colocados em um formulário de diálogo, e suas propriedades são configuradas no inspetor de objetos. O design de componentes é bastante flexível e é fácil criar componentes novos para suportar mecanismos de banco de dados diferentes (consulte o Manual do Desenvolvedor). Com o suporte do componente `TfrxDesigner`, eles permitem que os usuários de um aplicativo criem o design de um relatório no tempo de execução.



Descrições de componentes

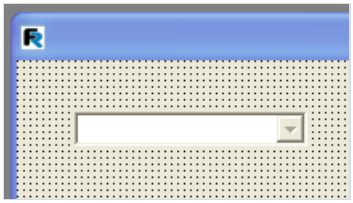
Vamos ver como os componentes são usados para o acesso de dados através de ADO. Eles são disponibilizados no FastReport ao adicionar o componente `TfrxADOComponents`  da paleta Delphi do FastReport ao formulário do projeto. Os seguintes objetos serão então exibidos na barra de ferramentas de objetos ao clicar na guia "Dados" no designer de relatórios: `TfrxADOTable` , `TfrxADOQuery` e `TfrxADODataBase` . Estes componentes são similares aos componentes Delphi correspondentes (`TADOTable` , `TADOQuery` e `TADOConnection`) em relação à sua funcionalidade. Há também um controle `TfrxDBLookupComboBox` que pode ser usado em formulários de diálogo.

Ícone	Nome	Descrição	Usado em
	TfrxDBLookupComboBox	controle usado para selecionar um valor de um diretório	formulários de diálogo
	TfrxADOTable	controle usado para acessar uma tabela de BD	guia "Dados"
	TfrxADOQuery	controle usado para executar uma consulta SQL	guia "Dados"
	TfrxADODataBase	controle usado para conectar a um BD	guia "Dados"

Vamos dar uma olhada em cada componente.

TfrxDBLookupComboBox

Este componente é usado para selecionar um valor de um conjunto de dados de consulta.



O componente tem as seguintes propriedades:

Propriedade	Descrição
DataSet	fonte de dados à qual o controle está conectado
ListField	nome do campo de BD exibido no controle
KeyField	nome do campo-chave do BD que identifica o registro selecionado
KeyValue	valor do campo-chave do BD que é retornado pela seleção da lista
Text	valor do campo da lista do BD exibido na lista
AutoOpenDataSet	se for True, a fonte de dados conectada é aberta automaticamente após o evento OnActivate da caixa de diálogo

Para conectar o controle ao conjunto de dados de consulta, insira valores para estas três propriedades: `DataSet`, `ListField` e `KeyField`.

Observe que o valor retornado é disponibilizado através das propriedades `Text` ou `KeyValue`, que não são exibidas no inspetor de objetos. Elas somente podem ser acessadas através do código. A posição inicial do cursor no conjunto de dados de consulta pode ser configurado por código usando `KeyValue`.

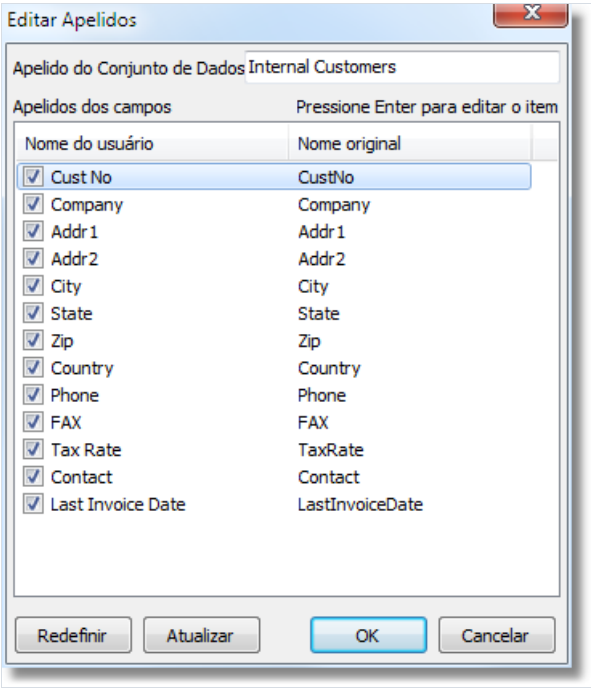
TfrxADOTable

Este componente é usado para acessar uma tabela de BD usando ADO. O componente possui as seguintes propriedades:

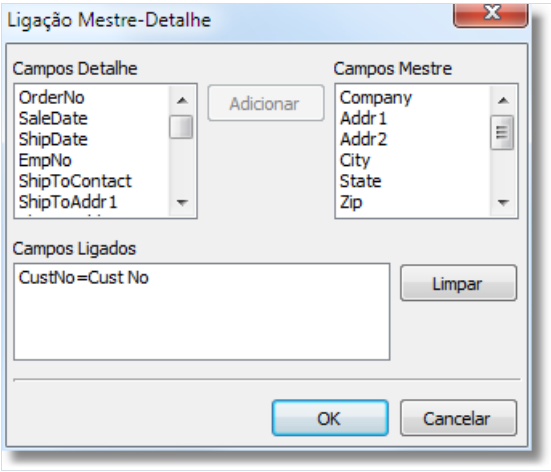
Propriedade	Descrição
DatabaseName	nome da conexão (nome do componente TfrxADODatabase)
FieldAliases	permite que aliases sejam configurados para os campos do conjunto de dados
Filter	expressão para filtrar registros
Filtered	se filtrado ou não
IndexFieldNames	nomes dos campos de índice (para classificação)
IndexName	nome do índice secundário
MasterFields	campos conectados ao conjunto de dados mestre
Master	conjunto de dados mestre
TableName	nome da tabela de BD
UserName	alias do conjunto de dados, usado no código

Estas propriedades são parecidas àquelas do componente `TADOTable` de Delphi. Para conectar o componente a uma tabela de BD, configura as propriedades `DatabaseName` e `TableName`. A tabela é aberta ao configurar a propriedade `Active` como True ou ao chamar o método `Open`.

O editor da propriedade `FieldAliases` é aberto no inspetor de objetos ao clicar duas vezes no componente. O editor permite a seleção dos campos que ficarão disponíveis para serem usados no código e a configuração dos aliases destes campos e do conjunto de dados como um todo.



O editor da propriedade `MasterFields` é usado para criar conexões mestre-detalhe entre duas tabelas. Para conectar duas tabelas com a relação mestre-detalhe, é necessário configurar uma segunda tabela na propriedade `Master` da tabela dependente e abrir o editor da propriedade `MasterFields`. Se a tabela possuir índices secundários que devem ser usados, configure antes a propriedade `IndexName`.



O editor vincula visualmente os campos "mestre" e "detalhe" dos conjuntos de dados. Quando dois conjuntos de dados são conectados em um relacionamento "Mestre-Detalhe", a movimentação no conjunto de dados mestre filtra o conjunto de dados detalhe automaticamente, e somente os registros que pertencem ao registro atual do conjunto de dados mestre são exibidos.

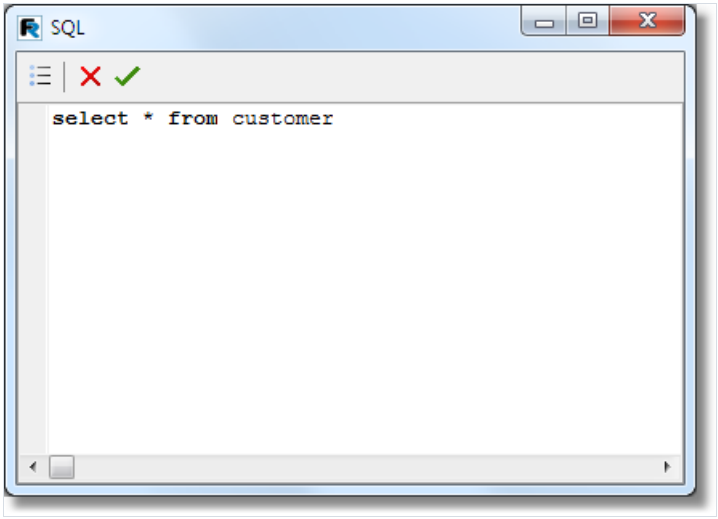
Para conectar campos nos dois conjuntos de dados, selecione um campo na lista à esquerda (o conjunto de dados detalhe), depois um campo da lista à direita (o conjunto de dados mestre) e clique no botão Adicionar. O vínculo entre os dois campos é exibido na lista inferior. Para limpar a lista inferior use o botão Limpar. Os campos vinculados devem ser indexados e de tipos compatíveis.

TfrxADOQuery

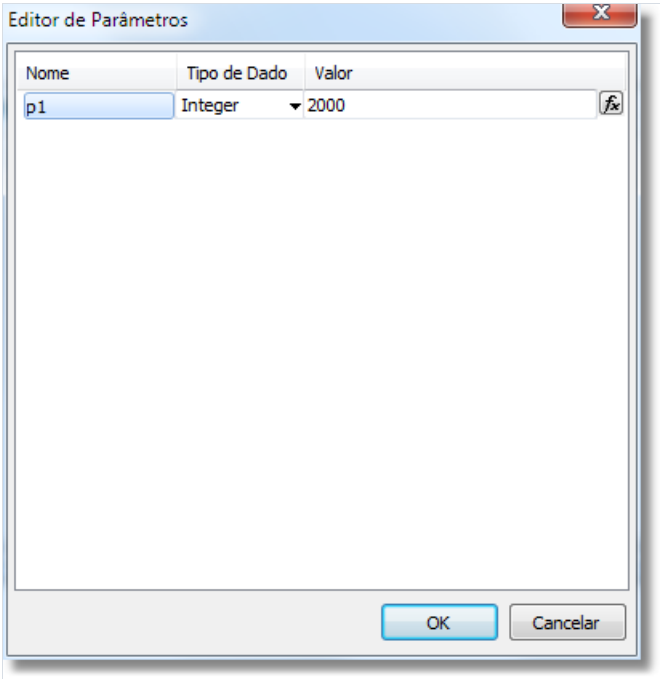
Este componente executa consultas SQL em um BD. Ele tem as seguintes propriedades:

Propriedade	Descrição
DatabaseName	nome da conexão (nome do componente `TfrxADODatabase`)
FieldAliases	permite que aliases sejam definidos para os campos do conjunto de dados
Filter	expressão para filtrar registros
Filtered	se filtrado ou não
Master	conjunto de dados mestre
Params	lista dos parâmetros da consulta
SQL	texto da consulta
UserName	alias do conjunto de dados, usado no código
IgnoreDupParams	se for True, permite usar parâmetros com nomes duplicados. O nome dos parâmetros da consulta não é editado no editor de parâmetros

As propriedades `Active`, `DatabaseName`, `FieldAliases`, `Filter`, `Filtered` e `Master` são parecidas às do componente `TfrxADOTable` descrito acima. A propriedade `SQL` possui seu próprio editor para inserir a consulta SQL.



A propriedade `Params` também possui seu próprio editor. Ela é habilitada quando o texto de uma consulta tiver parâmetros.



Um parâmetro pode ter dois tipos: seu valor é derivado da fonte mestre ou ele é configurado como um valor discreto (seja como um valor absoluto, como foi mostrado acima, ou vinculado a uma variável ou à propriedade de um objeto).

Quando um parâmetro é derivado do conjunto de dados mestre, é necessário configurar a propriedade `TfrxADOQuery.Master`. Este conjunto de dados deve conter um campo com o mesmo nome do parâmetro. Não é necessário especificar o tipo de parâmetro ou seu valor.

TfrxADODataBase

Este componente é usado para a conexão a um banco de dados. Sua função é parecida à do componente Delphi `TADOConnection`. O componente possui as seguintes propriedades:

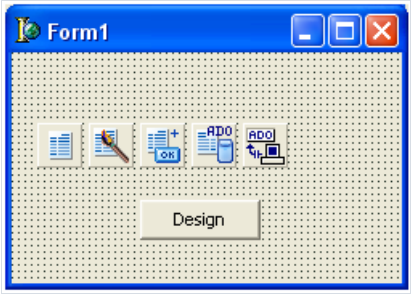
Propriedade	Descrição
Connected	se for True, a conexão é ativada
DatabaseName	a cadeia de conexão ADO
LoginPrompt	se a senha deve ser solicitada ao conectar ao BD

A propriedade `LoginPrompt` define se a senha deve ser solicitada ao efetuar uma conexão ao BD. Se `LoginPrompt` for False, é necessário incluir um nome de usuário e uma senha devem ser incluídos na cadeia de conexão ADO.

Designer de relatórios

Vamos dar uma olhada no design de um relatório simples usando componentes de acesso a dados durante o tempo de execução. Ele usará o banco de dados Demo.mdb, que é fornecido com o FastReport, como a fonte de dados - "FR\Demos\Main\demo.mdb".

Crie um novo projeto Delphi e adicione um componente de cada ao formulário: `TfrxReport`, `TfrxDesigner`, `TfrxDialogControls`, `TfrxADOComponents`, `TADOConnection` e `TButton`.



Configure a conexão ao banco de dados clicando duas vezes em `TADOConnection` e escolhendo "Criar cadeia de conexão", o fornecedor ("Microsoft Jet 4.0 OLE DB Provider") e o banco de dados ("FR\Demos\Main\demo.mdb"). Feche a caixa de diálogo de conexão ao clicar em OK e configure as propriedades do componente aqui:

```
ADOConnection1:
LoginPrompt = False

frxADOComponents1:
DefaultDatabase = ADOConnection1
```

Defina o seguinte manipulador para o botão "Design":

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  frxReport1.DesignReport;
end;
```

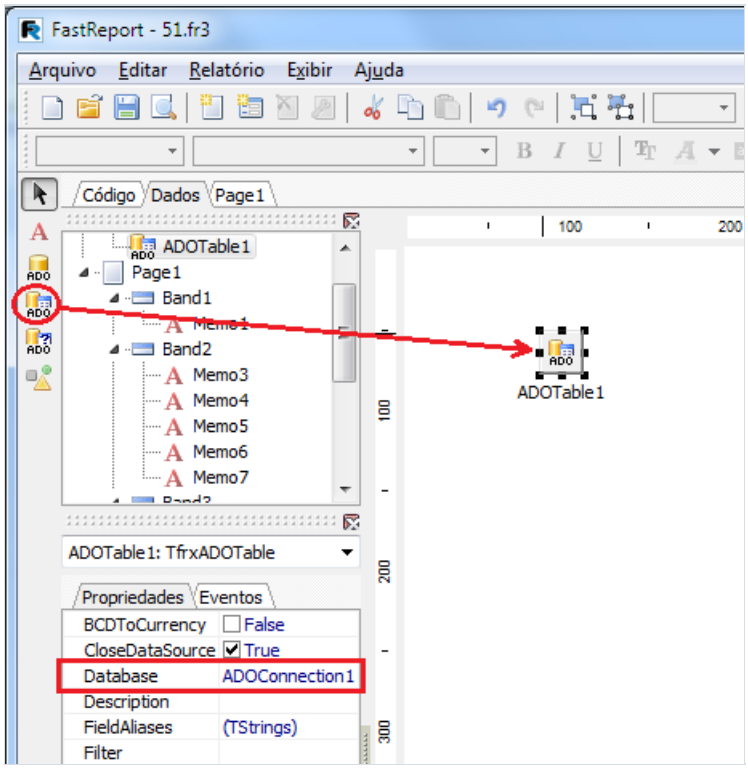
Depois disso, compile e execute o projeto. Isso é tudo que você precisa fazer para criar o designer de relatórios de tempo de execução para o usuário final.

Ao clicar no botão "Design", o desiger de relatórios do FR é aberto, contendo um relatório em branco. Vamos dar uma olhada no design de um relatório simples neste ambiente.

Relatório simples do tipo "Lista"

Este relatório irá conter dados de uma única tabela de BD. Para criar o relatório, siga as etapas a seguir:

Clique no botão "Novo Relatório" na barra de ferramentas do designer, e o FastReport criará um relatório vazio contendo as guias "Código", "Dados" e "Página1". Clique na guia "Dados" e coloque um componente "Tabela ADO" na página:

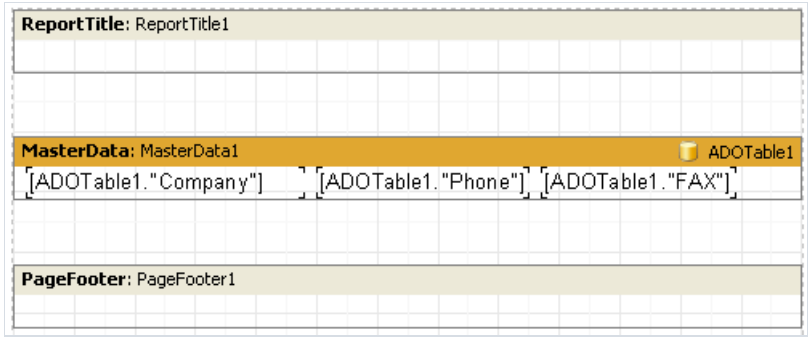


Observe que a propriedade `Database` já está conectada a nosso banco de dados, porque isso foi especificado na propriedade `TfrxADOComponents.DefaultDatabase`. O nome da tabela, porém, deve ser configurado agora:

```
TableName = 'Customer'
```

Vá para a guia Página1. Conecte a banda "DadosMestre" à tabela ao clicar duas vezes nela e selecionar "ADOTable1" na caixa de diálogo.

Arraste os campos exibidos abaixo da janela "Árvore de dados" para a página do relatório, que ficará parecido com isto:



Visualize o relatório com o botão "Visualizar" na barra de ferramentas.

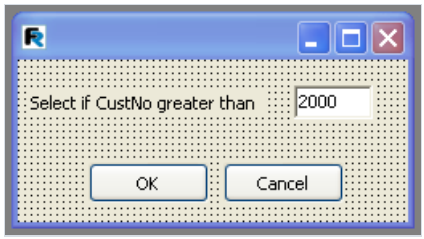
Relatório com consulta paramétrica

Vamos fazer o design de um relatório mais complicado, no qual os parâmetros da consulta são solicitados em uma caixa de diálogo antes de criar o relatório. Use o mesmo projeto da seção anterior e clique no botão Novo Relatório no designer de relatórios para limpar o antigo.

Abra a guia "Dados" e coloque um componente "Consulta ADO" na página. Clique duas vezes nele para abrir seu editor e insira o seguinte texto SQL:

```
select * from Customer where CustNo > :p1
```

Adicione um formulário de diálogo ao relatório e coloque os componentes "Label", "Edit" e dois componentes "Button" no formulário:



Configure as propriedades do componente:

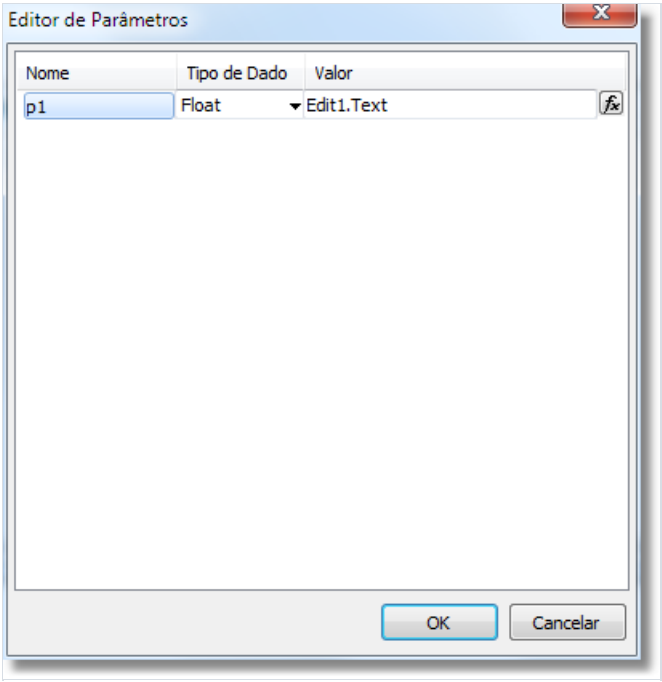
```
Label1:
Caption = 'Select if CustNo greater than'

Edit1:
Text = '2000'

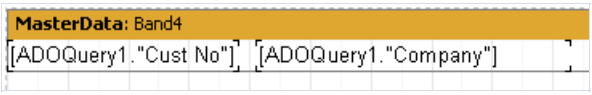
Button1:
Caption = 'OK'
ModalResult = mrOk

Button2:
Caption = 'Cancel'
ModalResult = mrCancel
```

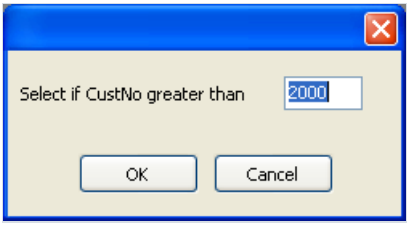
Abra o editor da propriedade `Params` do componente "Query" e configure o parâmetro:



Depois disso vá para a Página1 do design do relatório e crie o relatório conforme foi mostrado no exemplo anterior:

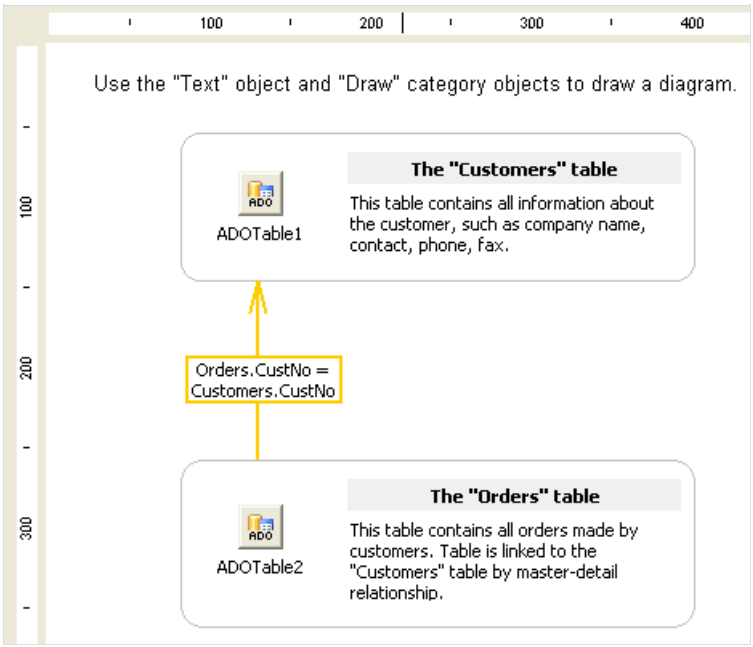


Quando o relatório é visualizado, é exibida uma caixa de diálogo solicitando o número do cliente. Após inserir um número e fechar a caixa de diálogo com o botão OK, o relatório é criado. Todos os clientes com números CustNo maiores que o inserido são exibidos.



Outros recursos úteis

Os elementos "Texto" e "Desenhar" podem ser colocados na página "Dados". Com o uso destes elementos, é possível desenhar diagramas básicos de explicação, conforme é mostrado abaixo:



Herança do relatório

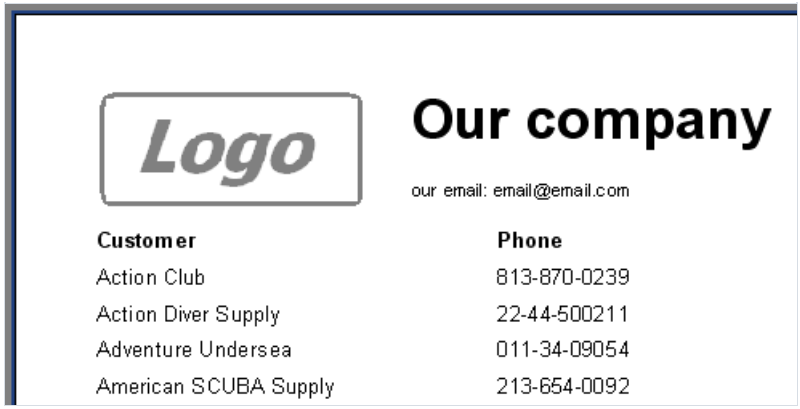
Muitas vezes um grupo de relatórios compartilha dados comuns – por exemplo, o cabeçalho/rodapé com o logo da empresa ou outros dados, como o e-mail ou o endereço, etc. Se houver a necessidade de modificar os dados da empresa, por exemplo o endereço de e-mail, então isso teria que ser feito em cada relatório! Para evitar essa tarefa entediante, é possível usar a herança do relatório. O que é a herança do relatório?

Como exemplo, relatórios normalmente possuem elementos (logo, nome da empresa, e-mail, etc.) colocados no título do relatório ou no cabeçalho da página. É possível criar um relatório base que contém somente estes elementos comuns. Todos os outros relatórios podem então usar o relatório base e assim conter todos os elementos comuns, assim como outros elementos que forem adicionados especificamente a cada relatório.

Se algo precisar ser alterado (como o logo ou e-mail), o relatório base é aberto e as alterações necessárias serão feitas ali. Todos os relatórios com herança do relatório base serão então alterados automaticamente. Na verdade, quando um relatório baseado em herança é aberto, o relatório base é aberto primeiro, seguido do relatório derivado.

Criar um relatório base

Vamos criar um relatório simples que utilize a herança. Nosso relatório final deve ter esta aparência:

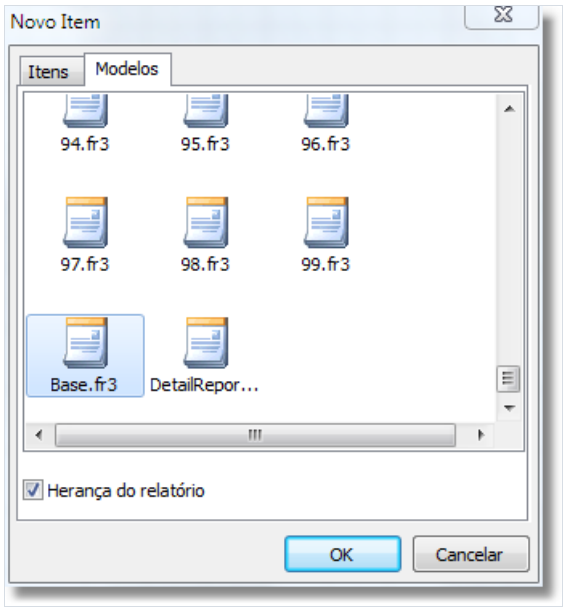


Primeiro é necessário criar o relatório base. Que elementos ele deve conter? Os elementos são o bitmap do logo, o título "Nossa empresa" e o endereço de e-mail. Crie um relatório novo e coloque os objetos comuns em "TítuloDoRelatório":



Salve o relatório como "base.fr3". Em que pasta? Isso depende de como você configurou o componente `TfrxDesigner`. Por padrão, o FastReport procura os relatórios de base na pasta que contém o arquivo .exe do aplicativo. De forma alternativa, é possível especificar uma pasta para modelos na propriedade `TfrxDesigner.TemplateDir`.

Agora crie o relatório derivado. Para fazer isso, use "Arquivo > Novo...", selecione a guia "Modelos" na caixa de diálogo e procure o relatório base ("base.fr3"). Clique em "Herança do relatório" e pressione OK:

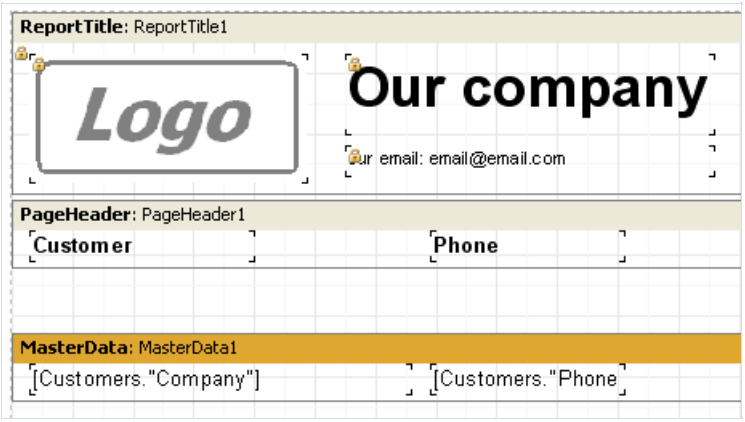


O FastReport criará um relatório que contém todos os objetos do relatório base. Eles são marcados com o símbolo de "cadeado":



O que significa o símbolo de "cadeado"? Ele significa que estes objetos não podem ser renomeados ou excluídos, e também não podem ser movidos para outra banda. É possível efetuar alterações em qualquer outra propriedades (como texto, cor e borda). Observe que se você alterar alguma propriedade de um objeto bloqueado (por exemplo a cor), esta alteração será armazenada no relatório derivado. Se você depois alterar a cor deste objeto no relatório base, a alteração será ignorada pelo relatório derivado. Por exemplo: abra o relatório derivado, altere a cor de "Nossa empresa" para vermelho e salve o relatório. Agora abra o relatório base e configure a cor de "Nossa empresa" como verde. Quando o relatório derivado for aberto novamente, a cor de "Nossa empresa" ainda será vermelho. Por isso, é preferível alterar as propriedades de objetos que possuem o símbolo de "cadeado" no relatório base, e não no relatório derivado.

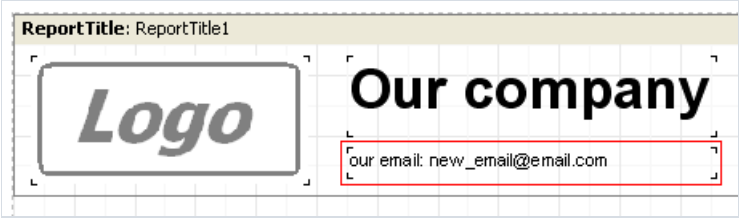
Vamos concluir nosso relatório. Tudo que é necessário é adicionar as bandas cabeçalho de página e dados mestre:



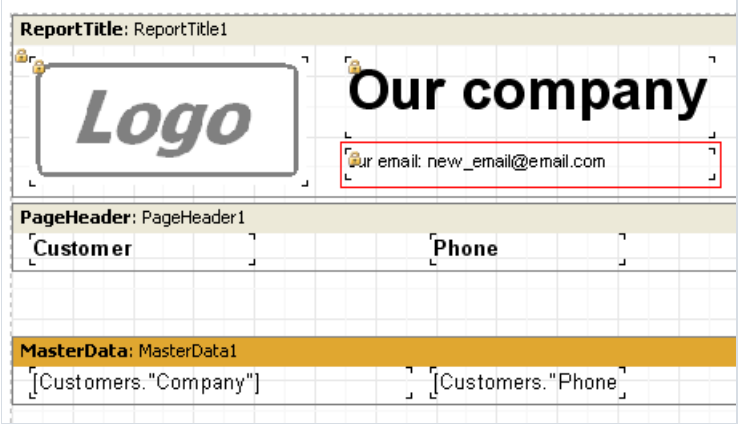
E o relatório está concluído.

Alterar um relatório base

Vamos ver como alterar um relatório base. Abra o relatório base ("base.fr3" em nosso exemplo) e altere alguns campos. Vamos alterar o endereço de e-mail:



Salve o relatório. Agora abra o relatório derivado e veja que o endereço de e-mail foi alterado nele também:

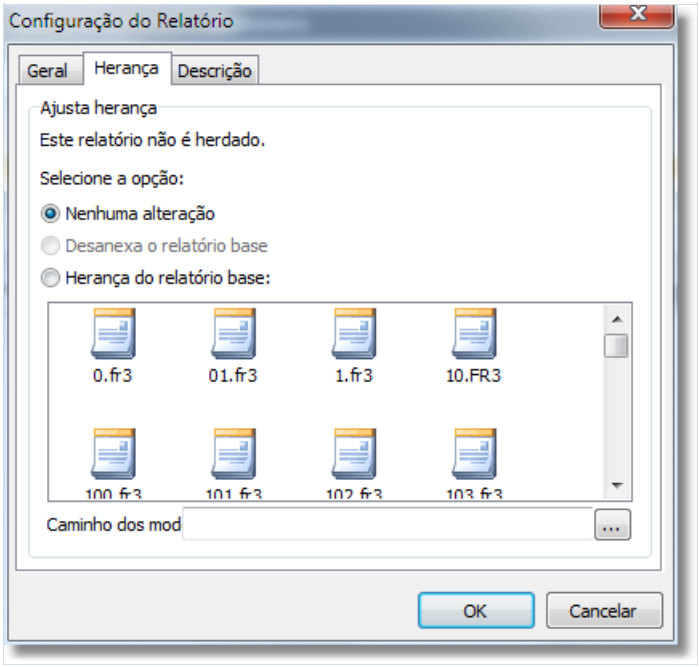


O que acontece se for necessário adicionar alguns objetos ao relatório base? Há uma regra simples: os relatórios de base e os relatórios derivados não podem conter nenhum objeto com o mesmo nome. Ao alterar o relatório base, pode não ser possível saber quantos relatórios utilizam o relatório base nem quais nomes de objetos foram usados nos relatórios derivados. Neste caso, siga uma estratégia simples: ao adicionar objetos a relatórios de base, nomeie os objetos usando um modelo, como 'NomeRelatório_NomeObjeto'. Em nosso exemplo, adicione um objeto "Texto" a nosso relatório e configure seu nome como 'BaseMemo3'.

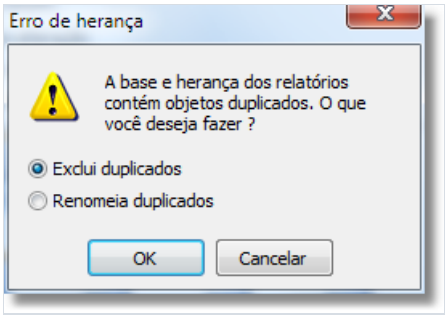
Não há nenhuma restrição em excluir objetos de um relatório base ou movê-los.

Controle de herança

Seguimos a criação de um relatório derivado do zero. O que acontece se um relatório existente deve ser modificado para ter a herança de um relatório diferente? Para fazer isso, abra o relatório e em "Relatório > Opções..." escolha a guia "Herança":



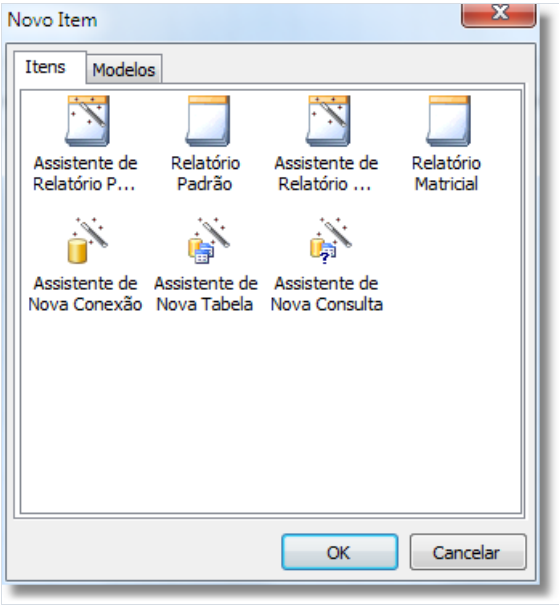
Selecione a opção "Herança do relatório base" e escolha o relatório base na lista. Se necessário, altere o caminho do modelo para ver um conjunto de modelos diferente. Aperte OK e FastReport irá combinar os dois relatórios. A seguinte mensagem de erro pode aparecer:



Isso acontece se os dois relatórios tiverem objetos com o mesmo nome. Os objetos com nomes duplicados podem ser excluídos ou renomeados no relatório derivado.

Wizards

O FastReport fornece assistentes que simplificam o processo de criação do relatório. Os assistentes podem ser encontrados em "Arquivo > Novo...":



Assistente de relatório novo

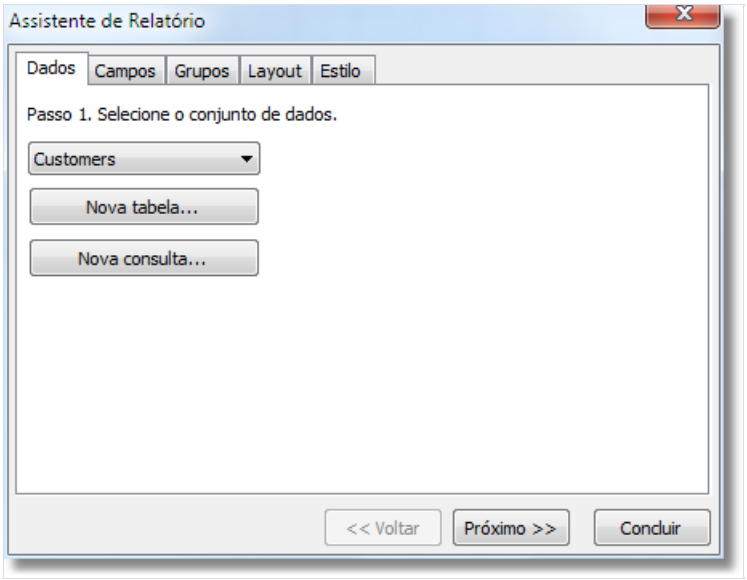
Há dois ícones de assistentes para criar relatórios novos, e dois ícones para a criação direta de novos relatórios:

- Assistente de relatório padrão
- Assistente de relatório matricial
- Relatório padrão
- Relatório matricial

Os ícones do tipo "Relatório padrão" e "Reltório matricial" criam relatórios padrão e matriciais vazios, respectivamente (veja mais informações sobre relatórios matriciais nas seções anteriores). Os relatórios novos contém um página vazia.

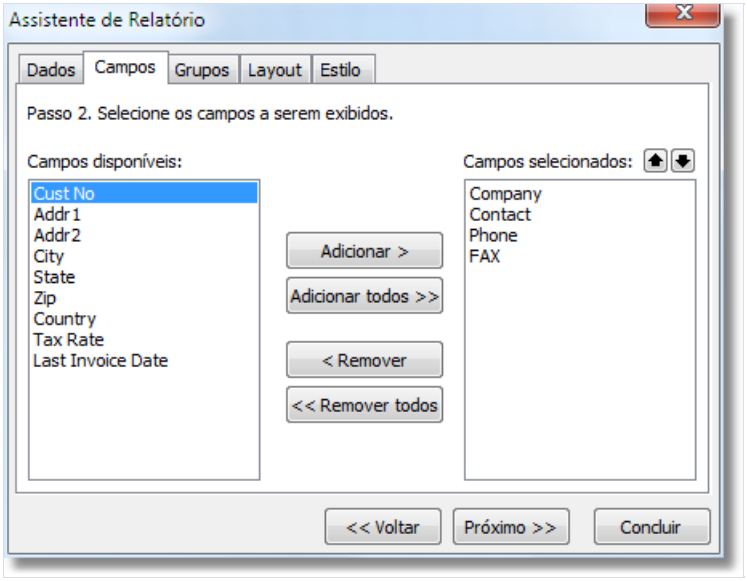
Os ícones do tipo "Assistente de Relatório Padrão" e "Assistente de Relatório Matricial" guiam o processo de escolher o conjunto de dados e os campos necessários do relatório, a criação opcional de grupos e a seleção do layout de dados. Vamos dar uma olhada em um relatório criado com a ajuda do "Assistente de Relatório Padrão".

Escolha "Arquivo > Novo..." e clique no ícone "Assistente de Relatório Padrão". A caixa de diálogo do assistente de relatório é exibida:



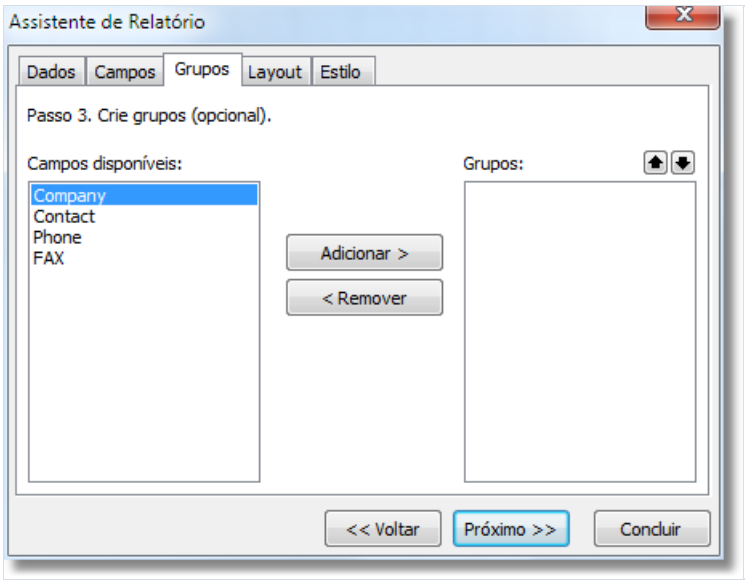
A caixa de diálogo possui diversas guias. Na primeira guia, escolha a fonte de dados do relatório. Escolha a tabela customer e aperte o botão "Próximo >>".

Na próxima guia selecione os campos a serem exibidos no relatório:



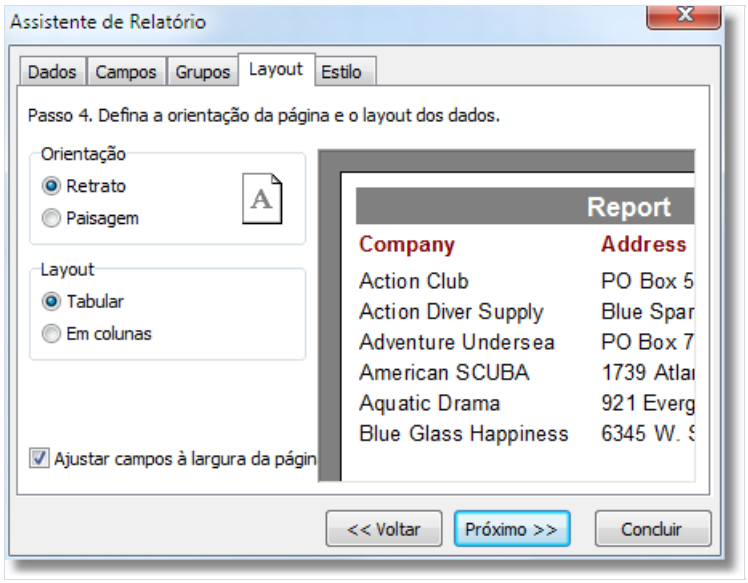
No lado esquerdo há uma lista de campos disponíveis; no campo direito há uma lista de campos que foram selecionados para serem exibidos no relatório. Use os botões "Adicionar >", "Adicionar todos >>", "< Remover" e "<< Remover todos" para mover os campos de uma lista para a outra. Use os botões para mover os campos selecionados para cima ou para baixo na lista. Adicione os campos "Company", "Contact", "Phone" e "FAX" à lista de campos selecionados e aperte o botão "Próximo >>".

Na próxima guia é possível criar um ou mais grupos. O FastReport adicionará o cabeçalho do grupo e as bandas de rodapé do grupo ao relatório.



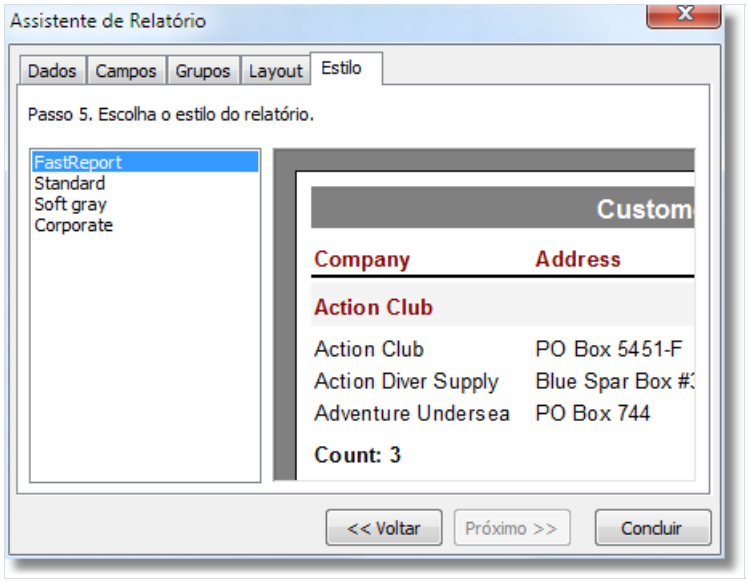
A criação de grupos é opcional. Ignore-a ao pressionar o botão "Próximo >>".

A próxima guia define a orientação da página e a escolha de um entre dois layouts de página – tabular e em colunas:

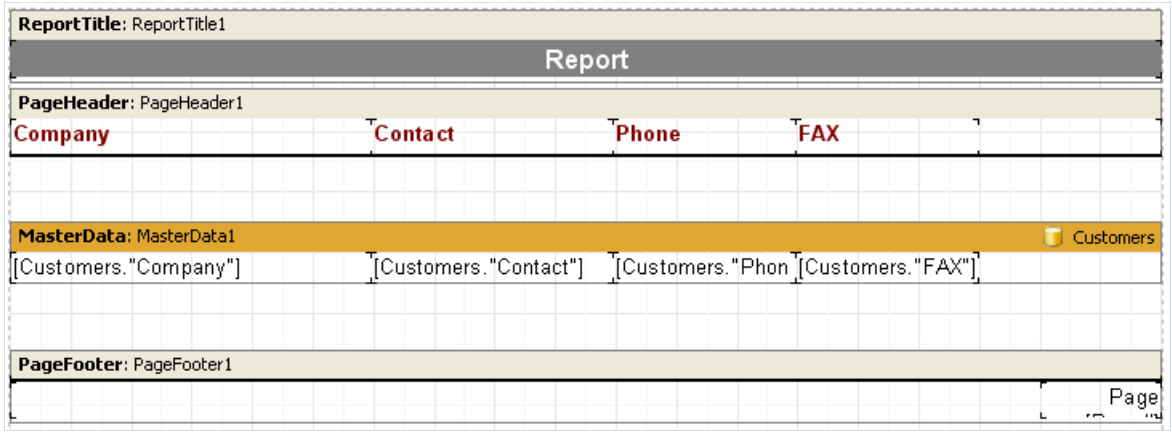


O layout escolhido é ilustrado na parte direita da caixa de diálogo.

Finalmente, a última guia lista os esquemas de cores disponíveis para seu relatório, que novamente são ilustrados no lado direito da caixa de diálogo.



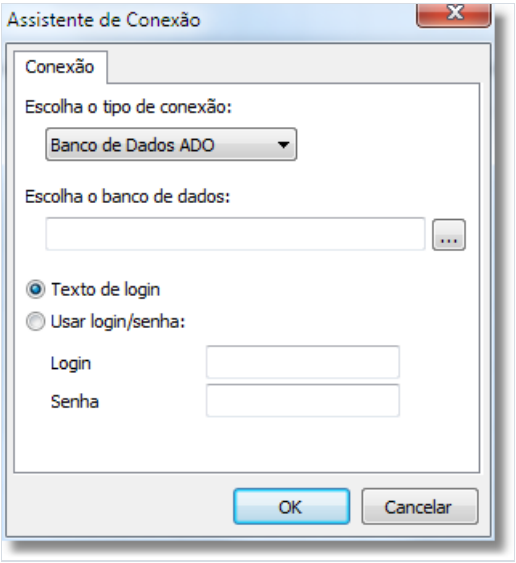
Ao pressionar o botão "Concluir", o assistente criará o seguinte relatório:



Agora é possível visualizar o relatório.

Assistente de nova conexão

Este assistente adiciona uma nova conexão de banco de dados a um relatório existente. Podem ser necessárias duas ou mais conexões se for necessário exibir dados de dois ou mais bancos de dados no relatório. O assistente adiciona o componente de banco de dados (por exemplo `TfrxADODatabase`) ao relatório.

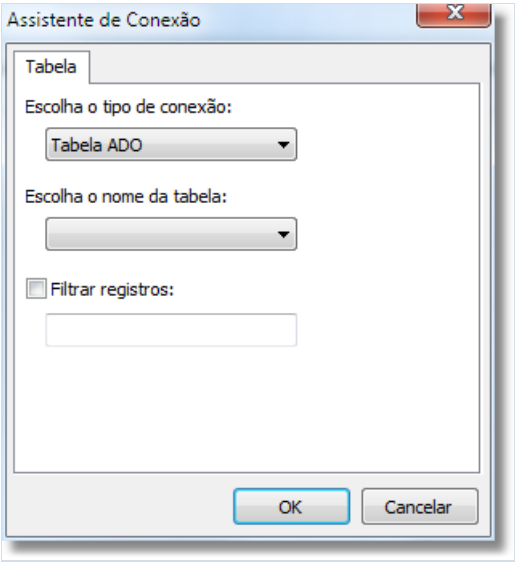


É necessário criar a cadeia de conexão com o botão ..., que abre a caixa de diálogo de conexão padrão do Windows para configurar os parâmetros do banco de dados e da conexão. Depois disso é possível configurar o nome de usuário e a senha, se forem necessários.

Observação: é possível criar uma nova conexão manualmente – apenas coloque um componente `TfrxADODatabase` na guia Dados do relatório.

Assistente de nova tabela

Este assistente adiciona uma nova tabela de banco de dados a um relatório existente.



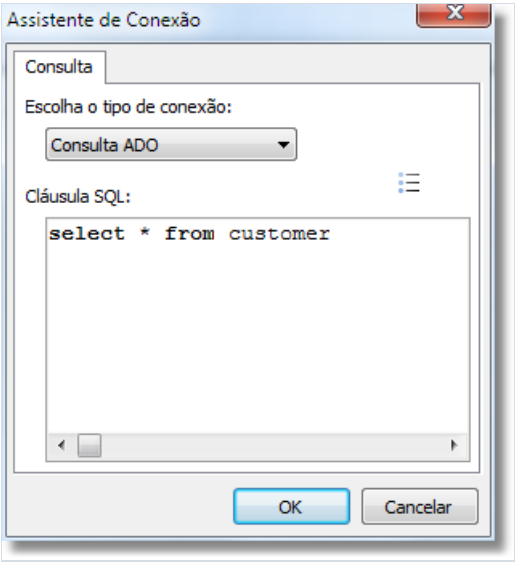
Selecione o nome da tabela. Se necessário, também é possível definir um filtro, por exemplo:

```
(CustNo > 2000) and (CustNo < 3000)
```

Observação: é possível criar uma nova tabela manualmente ao colocar o componente `TfrxADOTable` na guia Dados do relatório.

Assistente de nova consulta

Este assistente adiciona uma nova consulta SQL a um relatório existente.

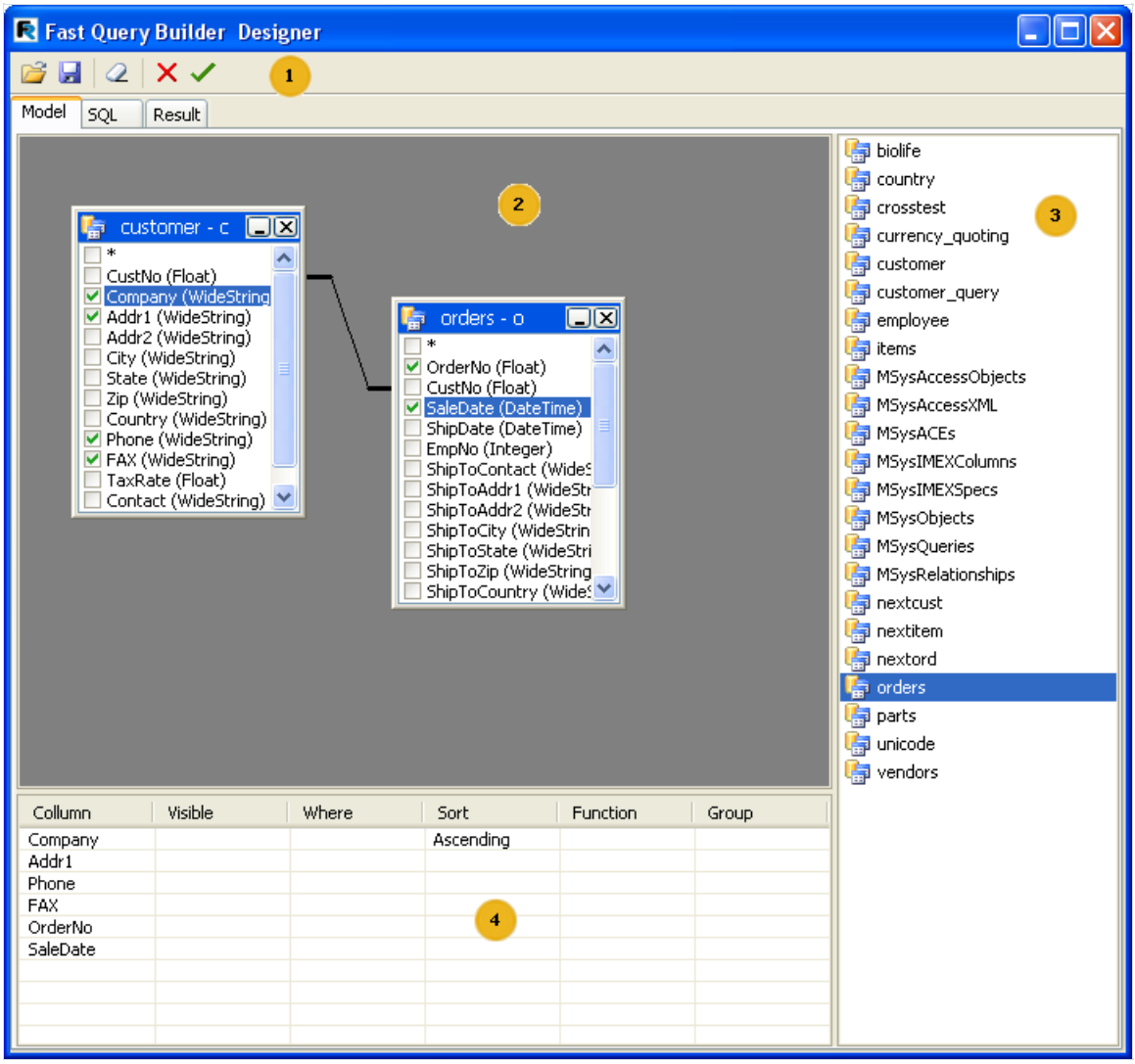


A consulta SQL deve ser criada aqui. É possível usar o construtor visual de consultas para fazer isso - clique no botão . O construtor de consultas é descrito mais adiante neste capítulo.

Observação: é possível criar uma nova consulta manualmente ao colocar o componente `TfrxADOQuery` na guia de Dados do relatório.

Construção da consulta

O FastQueryBuilder pode ser usado para criar consultas visualmente. FastQueryBuilder é incluído nas edições FastReport Professional e Enterprise, e também pode ser comprado como um produto independente. O construtor de consultas cria consultas visualmente na linguagem SQL. O construtor é ilustrado abaixo:



1 – barra de ferramentas

2 – espaço de trabalho do construtor

3 – lista de tabelas disponíveis

4 – propriedades de campo da tabela selecionada

Barra de ferramentas:

- abrir arquivo SQL

- salvar consulta em um arquivo (o diagrama da consulta também é salvo no arquivo)

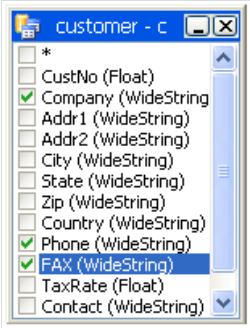
- limpar o espaço de trabalho do construtor

- botão OK: salvar e fechar o construtor

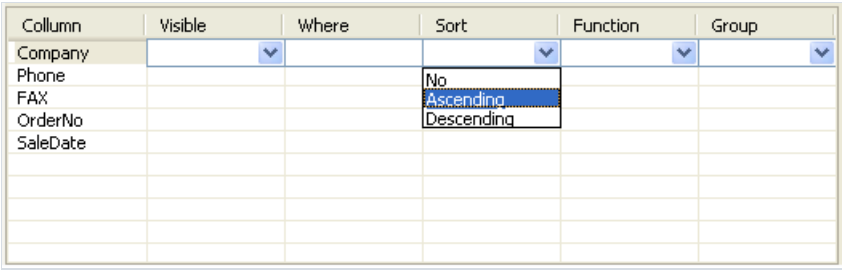
- botão Cancelar: fechar o construtor sem salvar

O espaço de trabalho e a lista de tabelas disponíveis suportam arrastar e soltar, ou seja, tabelas podem ser arrastadas para o espaço de trabalho com o mouse. Também é possível clicar duas vezes em uma tabela na lista de tabelas disponíveis.

Para incluir um campo da tabela na consulta, selecione-o na lista:

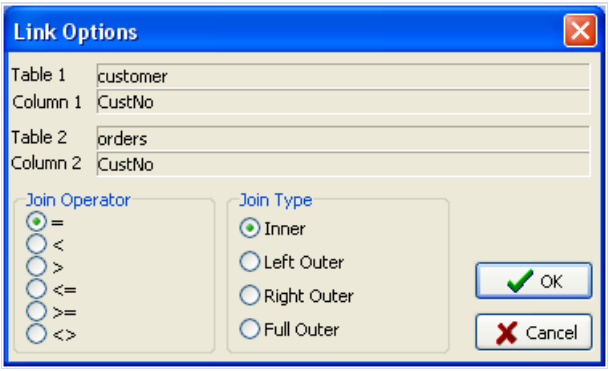


Os campos selecionados são exibidos no espaço de propriedades do campo (4):



- Visível: controla se o campo é exibido na saída
- Onde: condição de seleção do campo, por exemplo '> 5'
- Classificação: direção de classificação do campo
- Função: função aplicada ao campo
- Grupo: agrupamento no campo

Ao "arrastar" campos entre as tabelas no espaço de trabalho (2), "linhas de união" são exibidas. Quando campos são unidos, a compatibilidade dos tipos dos campos unidos é verificada. O construtor não permite a união de campos de tipos não compatíveis. Para alterar os parâmetros da união, mantenha o cursor sobre a "linha de união", clique com o botão direito e selecione o item Opções. A caixa de diálogo Opções de Link é exibida, na qual é possível configurar a união, como é mostrado abaixo:

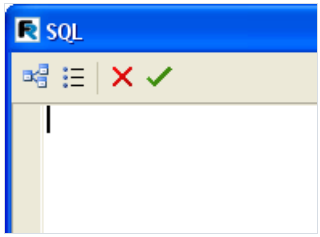


Utilização do construtor de consultas

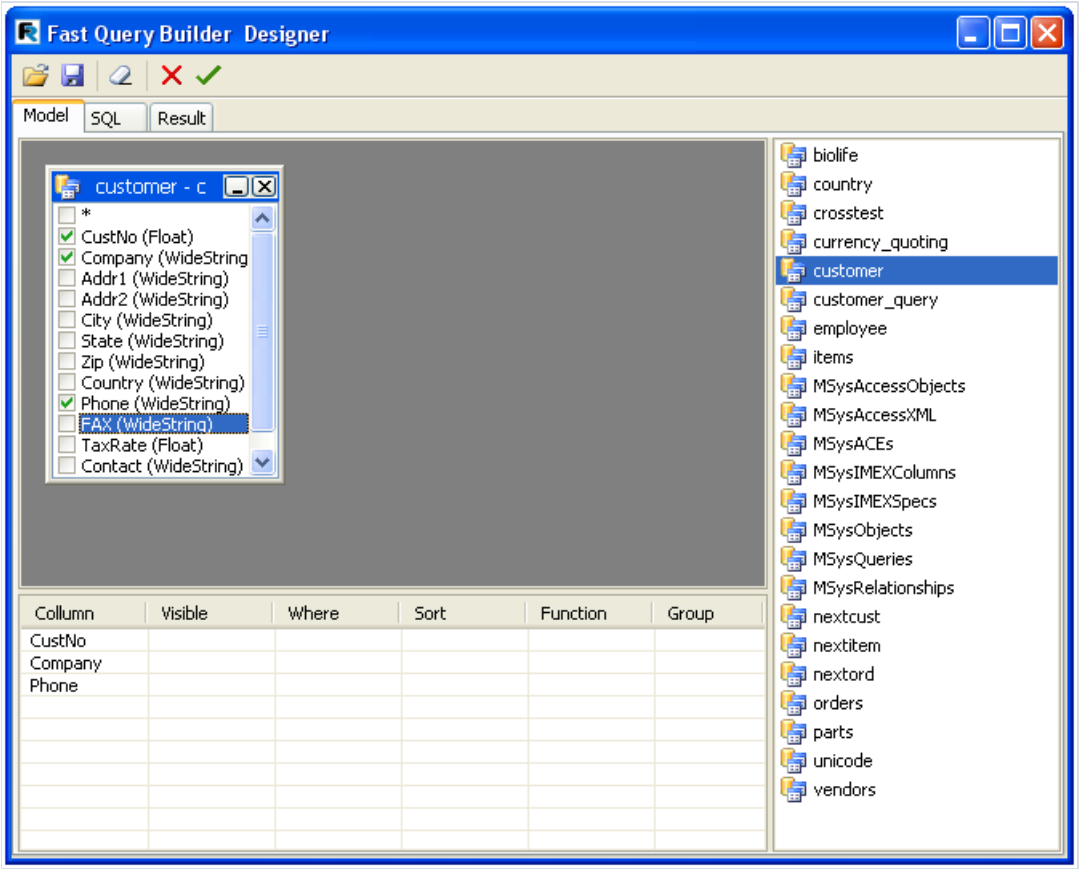
Vamos criar um relatório simples usando o construtor de consultas.

Clique em "Novo Relatório" na barra de ferramentas do designer para criar uma página de relatório com as bandas "Cabeçalho do Relatório", "Dados de 1o. Nível" e "Rodapé da Página".

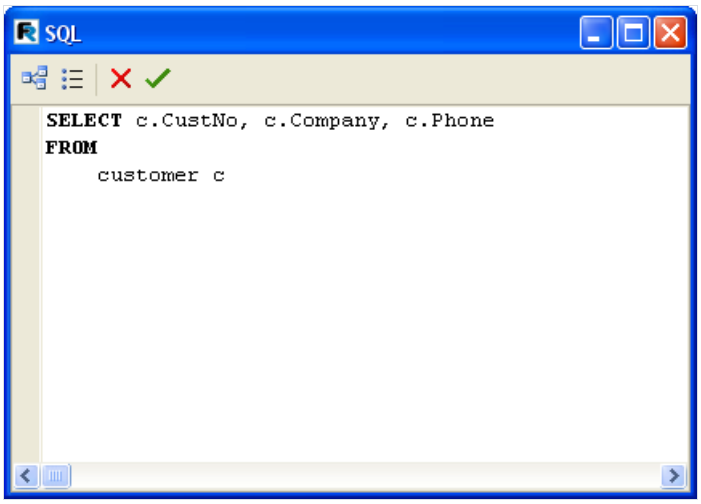
Coloque um componente "Consulta ADO" na guia "Dados". Clique duas vezes no componente para abrir o editor de consultas.



Clique no botão no editor de consultas para abrir a janela do criador de consultas. Selecione a tabela Customer na lista de tabelas (3) e arraste-a para o espaço de trabalho (ou clique duas vezes na tabela). Selecione os campos CustNo, Company e Phone:



Isso é tudo que é necessário fazer para criar uma consulta. O texto da consulta é exibido na guia SQL e a guia Resultado exibe os dados retornados pela consulta. Clique em para fechar o construtor e retornar ao editor de consultas, onde o texto da consulta é exibido:



Atenção! Se o texto da consulta for alterado no editor de consultas, o diagrama da consulta de tabelas e uniões será perdido. Não altere o texto da consulta manualmente, sempre abra o construtor de consultas e modifique o diagrama visualmente.

Ao clicar em no editor de consultas, você volta para o designer de relatórios. Tudo que resta fazer é conectar a banda "DadosMestre" à fonte de dados e colocar os campos necessários na banda "DadosMestre".

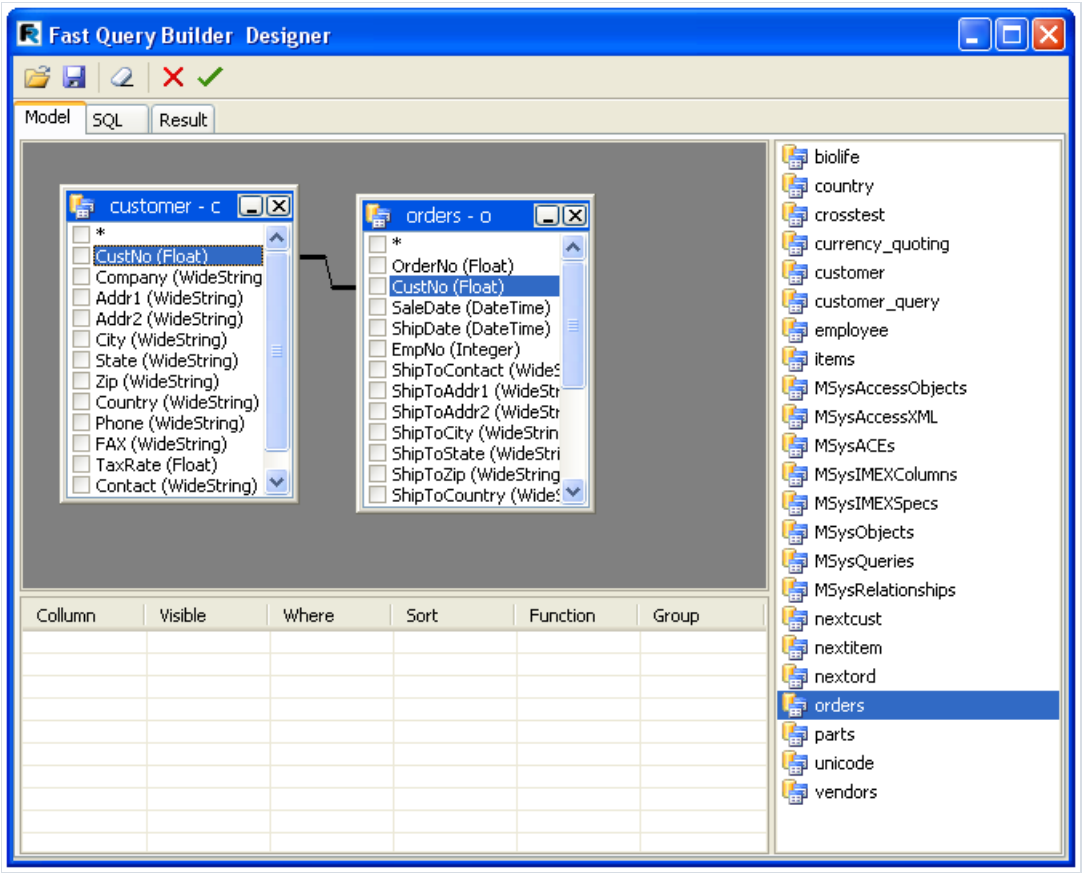
Construir consultas complexas

No último exemplo criamos um relatório baseado em uma tabela. Agora vamos ver como criar uma consulta na qual os dados são fornecidos por duas tabelas.

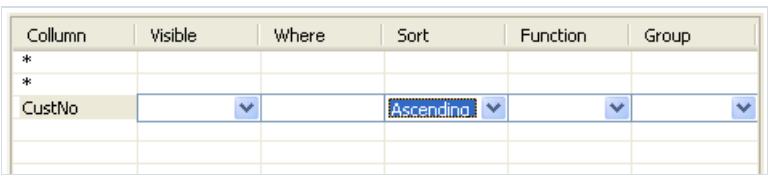
Anteriormente abordamos um relatório que utiliza grupos. Vamos criar uma consulta para este relatório com a utilização do construtor de consultas. Precisamos criar uma consulta em SQL que retornará dados de ambas as tabelas, e os dados devem ser agrupados com condições específicas. Em nosso exemplo a condição será os campos CustNo nas duas tabelas.

Assim como no exemplo anterior, crie um relatório novo e coloque um componente "ADOQuery" na página. Abra o editor de consultas e depois o construtor de consultas.

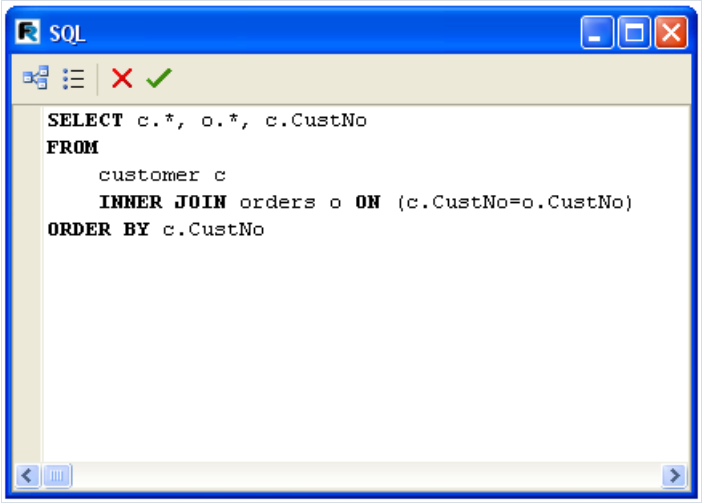
Arraste duas tabelas para a área de trabalho – Customers e Orders. Ambas possuem o campo CustNo, que será usado para unir as tabelas. Arraste o campo CustNo de uma tabela para a outra para unir as duas tabelas:



Agora é necessário configurar os campos que devem ser exibidos e o campo de classificação. Marque o campo "*" nas duas tabelas e marque o campo CustNo na tabela Customer. Os campos selecionados são exibidos na lista de parâmetros de campo. Selecione a ordem de classificação do campo CustNo:

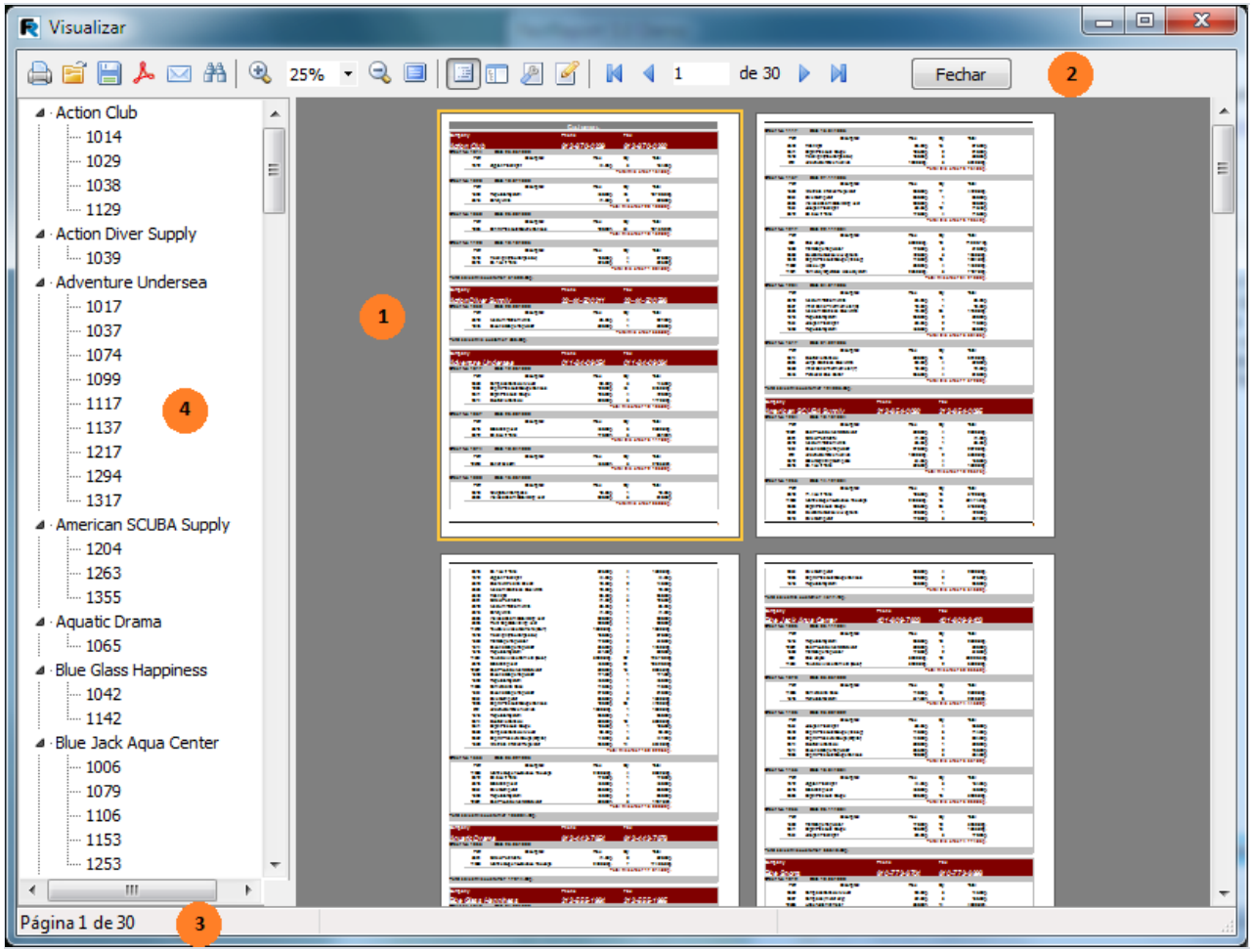


Isso é tudo que é necessário fazer para concluir a consulta. Esta é a aparência do código SQL:


SQL
SELECT c.*, o.*, c.CustNo
FROM
customer c
INNER JOIN orders o ON (c.CustNo=o.CustNo)
ORDER BY c.CustNo

Visualização, impressão e exportação de um relatório

Um relatório construído pode ser exibido e impresso ou exportado para um dos formatos com suporte. Tudo isso pode ser feito na janela de visualização.



Legenda:

1 – páginas concluídas do relatório

2 – barra de ferramentas

3 – barra de status

4 – espaço da estrutura de tópicos para a árvore da estrutura de tópicos (exibida acima) ou para miniaturas

A barra de ferramentas possui os seguintes botões:



Ícone	Nome	Descrição
	Imprimir relatório	imprime o relatório – atalho: Ctrl+P

Ícone	Nome	Descrição
	Abrir relatório	abre o arquivo que contém o relatório concluído (*.fp3)
	Salvar relatório	salva o relatório em um arquivo (*.fp3) ou exporta o relatório para um dos formatos suportados
	Exportar para PDF	exporta o relatório para um arquivo Adobe Acrobat (*.pdf)
	Enviar por e-mail	exporta o relatório para um dos formatossuportados e o envia por e-mail como anexo
	Localizar texto	pesquisa o texto no relatório – atalho: Ctrl+F
	Aumentar Zoom	aumenta o zoom na visualização
	Zoom	seleciona uma escala arbitrária para o zoom
	Diminuir Zoom	diminui o zoom da visualização
	Tela Cheia	exibe o relatório em tela cheiaretorne ao tamanho normal clicando duas vezes no relatório
	Estrutura de tópicos do relatório	exibe ou oculta a estrutura de tópicos do relatório
	Miniaturas	exibe ou oculta a exibição das miniaturas
	Propriedades da Página	abre a caixa de diálogo com as configurações da página
	Editar Página	edita a página atual
	Primeira Página	salta para a primeira página do relatório
	Página Anterior	salta para a página anterior do relatório
	Número da Página	salta para o número da página do relatóriodigite o número e aperte Enter
	Próxima Página	salta para a próxima página do relatório
	Última Página	salta para a última página do relatório
	Fechar janela	fecha a visualização

Teclas de controle

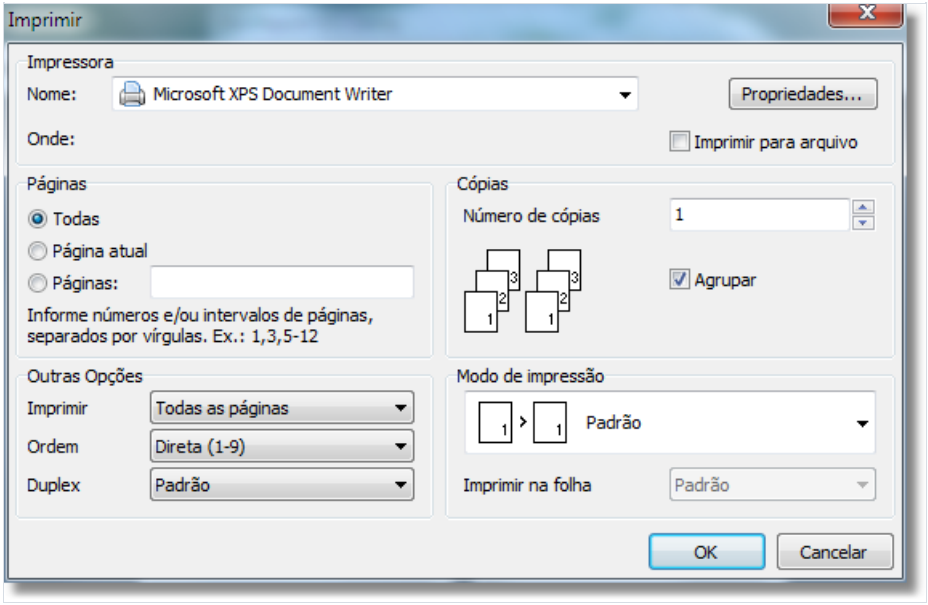
Teclas	Descrição
Ctrl+S	salvar relatório em um arquivo *.fp3
Ctrl+P	imprimir relatório
Ctrl+F	pesquisa de texto
F3	pesquisar novamente
Arrows	rolagem suave do relatório
PageUp, PageDown	rolagem para cima/para baixo do relatório
Ctrl+PageUp, PageDown	rolagem para a próxima página/página anterior do relatório
Home	início do relatório
End	final do relatório

Controle do mouse

Ação	Descrição
Botão esquerdo	clique no objeto selecionado (no relatório interativo)rolagem do relatório no modo "mão" (mova o mouse com o botão apertado)aumentar o zoom no modo "lupa"
Botão direito	menu de contextodiminuir o zoom no modo "lupa"
Dois cliques	retornar ao tamanho normal quando estiver no modo tela cheia
Roda do mouse	rolagem do relatório

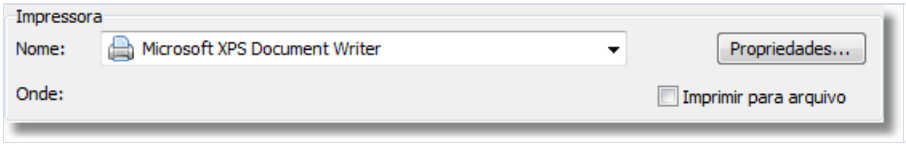
Impressão do relatório

Para imprimir um relatório, clique no botão (ou no atalho Ctrl+P). A caixa de diálogo padrão é aberta.

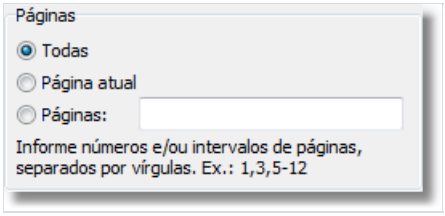


Vamos dar uma olhada nas opções disponíves nesta caixa de diálogo.

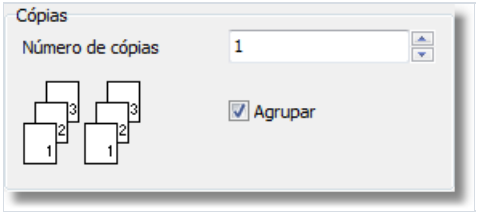
Painel "Impressora": selecione a impressora na qual será impresso o relatório; configure as propriedades da impressora, por exemplo a qualidade de impressão; escolha a impressão para um arquivo.



Painel "Páginas": selecione quais páginas serão impressas (todas, a página atual ou um intervalo selecionado).



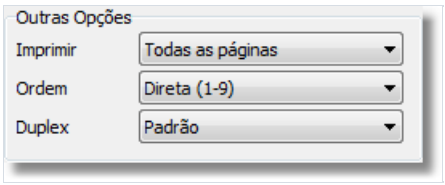
Painel "Cópias": configure o número de cópias a serem impressas. Ao imprimir mais de uma cópia, se Agrupar estiver marcado, a primeira cópia é impressa completamente, então a segunda é impressa completamente, etc. Se Agrupar não estiver marcado, então todas as cópias da primeira página são impressas, seguidas de todas as cópias da segunda página, etc.



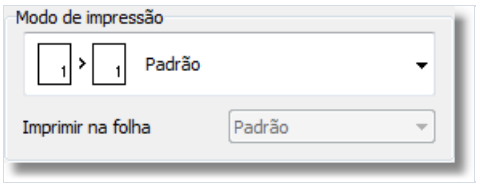
Painel "Outras Opções":

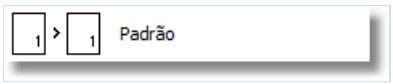
- Imprimir: seleciona quais páginas serão impressas (Todas as páginas, Páginas pares, Páginas ímpares)

- Ordem: imprimir páginas na ordem direta ou inversa (primeira página até a última, última página até a primeira)
- Duplex: gerenciar duplex por padrão (as configurações do relatório são usadas) ou escolha uma das opções de duplex: vertical, horizonal, simplex

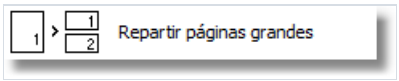


Painel "Modo de impressão": selecione o modo de impressão.

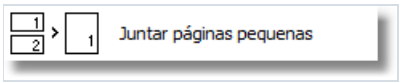




- Modo padrão: imprime na folha definida no relatório. Uma página de relatório é impressa em cada folha



- Repartir páginas grandes: este modo é útil ao imprimir um relatório A3 em uma folha A4. Uma página do relatório é impressa em mais de uma folha. Quando este modo é escolhido, o tamanho da folha ("Imprimir na folha") também deve ser especificado.



- Juntar páginas pequenas: este modo é útil ao imprimir um relatório A4 em uma folha A3. Duas ou mais páginas de visualização são impressas em uma folha. Quando este modo é escolhido, o tamanho da folha ("Imprimir na folha") também deve ser especificado.

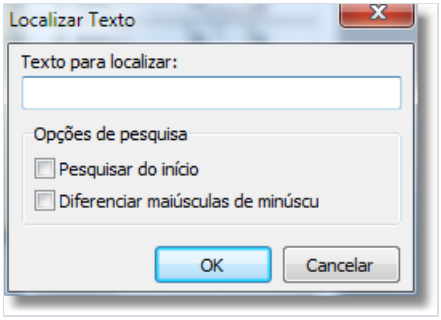


- Modo escala: o relatório é impresso em uma folha de tamanho especificado. Toda a saída do relatório é dimensionada. Uma página da visualização é impressa por folha. Quando este modo é escolhido, o tamanho da folha ("Imprimir na folha") também deve ser especificado.

Ao clicar em OK, a impressão do relatório é iniciada. Se "Imprimir para arquivo" for marcado, a caixa de diálogo padrão "Salvar Como..." é aberta. O relatório é salvo em um arquivo de extensão *.prn. O arquivo contém uma cópia das informações enviadas à impressora.

Pesquisa de texto em relatórios

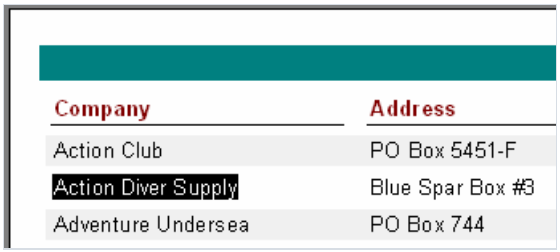
O FastReport pode pesquisar uma frase específica no texto de um relatório visualizado. Pesquise usando o botão na barra de ferramentas (ou o atalho Ctrl+F). A caixa de diálogo de pesquisa é aberta:



Insira a frase que deve se pesquisada e seleciona as opções se necessário:

- Pesquisar do início: pesquisar o relatório desde o início; se não for marcado, a pesquisa é efetuada a partir da página atual
- Diferenciar maiúsculas de minúsculas: as maiúsculas e minúsculas da frase de pesquisa devem corresponder com o resultado da pesquisa

Clique em OK e a pesquisa é iniciada. O primeiro resultado (se houver) é realçado:

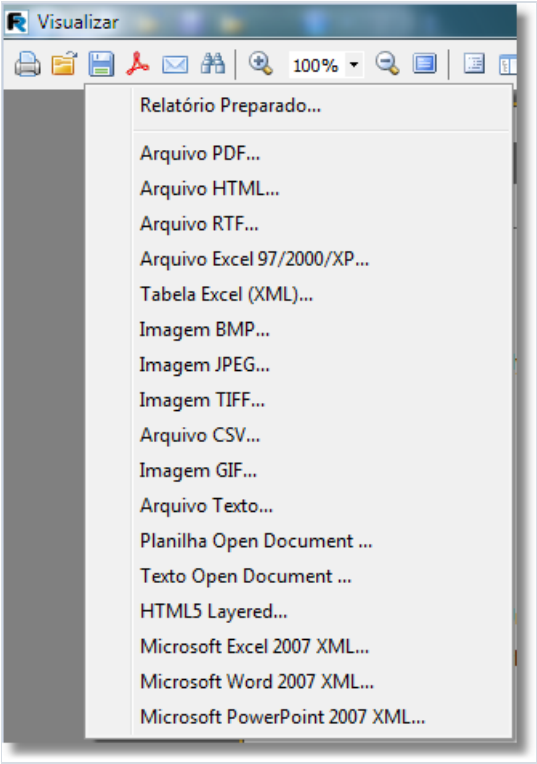


Para continuar a pesquisa, clique em F3. O próximo resultado (se houver) será realçado.

Exportar o relatório

O FastReport pode exportar um relatório visualizado para diversos formatos para permitir a edição adicional, arquivamento ou o envio por e-mail, etc. Para habilitar a exportação, os componentes de exportação do FR apropriados devem ser adicionados ao formulário Delphi.

O FastReport pode exportar para os seguintes formatos: PDF, Planilha Open Document, Texto Open Document, Excel 97/2000/XP, Excel XML, Excel 2007, RTF, Word 2007, PowerPoint 2007, HTML, texto, CSV, BMP, Jpeg, Tiff e Gif. Relatórios também podem ser enviados por e-mail em qualquer um dos formatos listados.



O FastReport usa um dos seguintes três métodos para exportar relatórios:

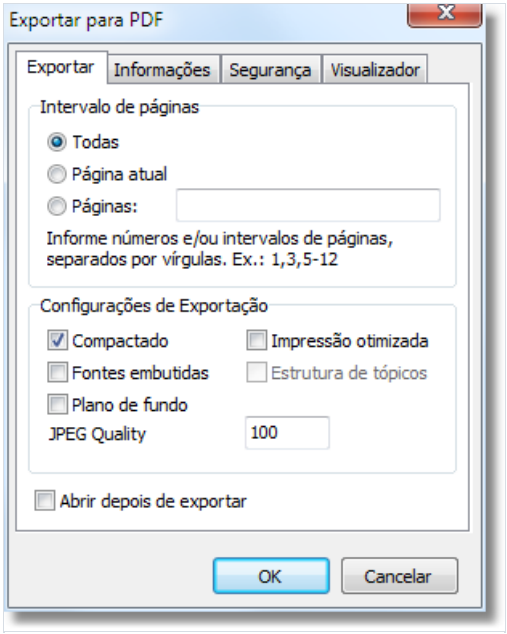
- 'Camada': cada objeto do relatório é exportado para uma camada separada. A saída exportada é parecida com a visualização original.
- 'Tabela': a exportação para o arquivo de saída é feita através da criação de uma grade de transição na memória, seguido da saída dessa grade. A saída exportada corresponde à visualização original, contanto que os princípios do bom design de um relatório tenham sido seguidos (consulte o capítulo "Considerações do design do relatório").
- 'Desenho': objetos exportados são capturados da imagem da página. A saída exportada é uma cópia direta da visualização. este método é usado ao exportar para formatos gráficos.

Exportar para o formato PDF

PDF (Portable Document Format): um formato independente de plataforma para documentos eletrônicos criado por Adobe Systems. O pacote gratuito Adobe Reader é usado para a visualização. Este formato é flexível - ele permite a inclusão das fontes necessárias e imagens de vetor e bitmap; é uma maneira de distribuir e armazenar documentos que foram criados para a visualização ou impressão.

O método de exportação é 'camada por camada'.

Ao exportar para o formato PDF, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

- Compactado: o arquivo de saída é compactado; o tamanho do arquivo é reduzido, porém o tempo de exportação aumenta
- Fontes embutidas: todas as fontes usadas no relatório estão contidas no arquivo PDF de saída, o que permite a renderização precisa em computadores nos quais as fontes estão ausentes; o tamanho do arquivo de saída aumenta consideravelmente
- Plano de fundo: a imagem gráfica configurada como plano de fundo é exportada ao arquivo PDF; o tamanho do arquivo de saída aumenta consideravelmente
- Impressão otimizada: imagens gráficas são exportadas em alta resolução para obter uma impressão precisa; esta opção somente é necessária quando um documento possuir gráficos e precisar ser impresso; o tamanho do arquivo de saída aumenta consideravelmente
- Estrutura de tópicos: esta opção somente é habilitada se o relatório conter uma estrutura de tópicos; a estrutura de tópicos é exportada ao arquivo PDF
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação usando o visualizador de PDF padrão instalado no computador (por exemplo o Adobe Reader).

Recurso da exportação: objetos RichText são exportados como gráficos.

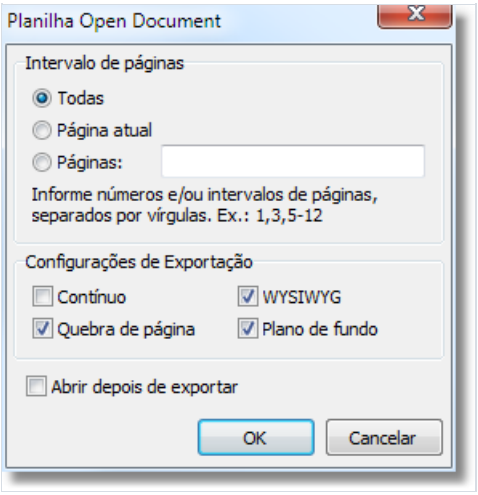
Exportar para Open Document

O formato Open Document (ODF, OASIS Open Document Format for Office Application) foi projetado pela OASIS e é baseado no formato XML usado no OpenOffice.

O FastReport suporta a exportação para arquivos de tabela (.ods) e texto (.odt). Estes arquivos podem ser abertos no OpenOffice.

O método de exportação é 'tabela'.

Ao exportar para o formato ODF, é aberta uma caixa de diálogo que solicita as configurações do arquivo de saída.



Configurações de exportação:

- Contínuo: exporta como um documento contínuo, sem quebras de página e sem cabeçalhos/rodapés de página
- Quebra de página: habilita quebras de página no documento
- WYSIWYG: uma representação precisa do relatório visualizado; se esta opção for desabilitada, a otimização é habilitada, o que reduz o número de linhas e colunas no arquivo de exportação
- Plano de fundo: a imagem gráfica configurada como o plano de fundo da página é exportada ao arquivo ODF; o tamanho do arquivo de saída aumenta consideravelmente
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação.

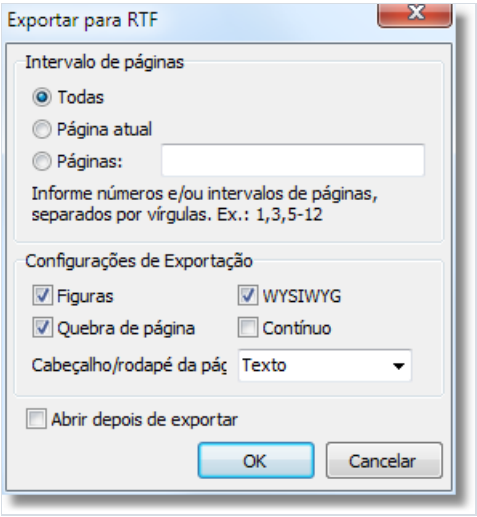
Recurso da exportação: objetos RichText são exportados como texto simples, e a exportação de imagens gráficas é suportada.

Exportar ao formato RTF

RTF (Rich Text Format) foi desenvolvido pela Microsoft como o padrão para o intercâmbio de documentos de texto. Documentos RTF são suportados por muitos editores de texto e sistemas operacionais modernos.

O método de exportação é 'tabela'.

Ao exportar para o formato RTF, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

- Figuras: habilita a exportação de imagens gráficas ao arquivo
- Quebra de página: habilita quebras de página no documento
- WYSIWYG: uma representação precisa do relatório visualizado; se esta opção for desabilitada, a otimização é habilitada, o que reduz o número de linhas e colunas no arquivo de exportação
- Contínuo: exporta como um documento contínuo, sem quebras de página e sem cabeçalhos/rodapés de página
- Cabeçalho/rodapé da página: modo de exportação do cabeçalho/rodapé; os modos são: Texto (c/r exportados como texto normal), Cabeçalho/rodapé (c/r são exportados) e Nenhum (c/r não são exportados).
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação usando o visualizador de RTF padrão instalado no computador (por exemplo Microsoft WordPad).

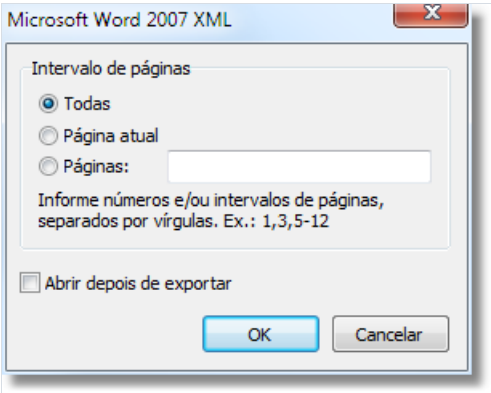
Recurso da exportação: objetos RichText são completamente integrados no formato RTF; a precisão da renderização e do tamanho do arquivo depende do design do relatório. Consulte o capítulo "Considerações do design do relatório".

Exportar para Word 2007

Word 2007 é um aplicativo para trabalhar com documentos de texto. Ele está incluído no Microsoft Office 2007.

O método de exportação é 'tabela'.

Ao exportar para o formato Word 2007, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

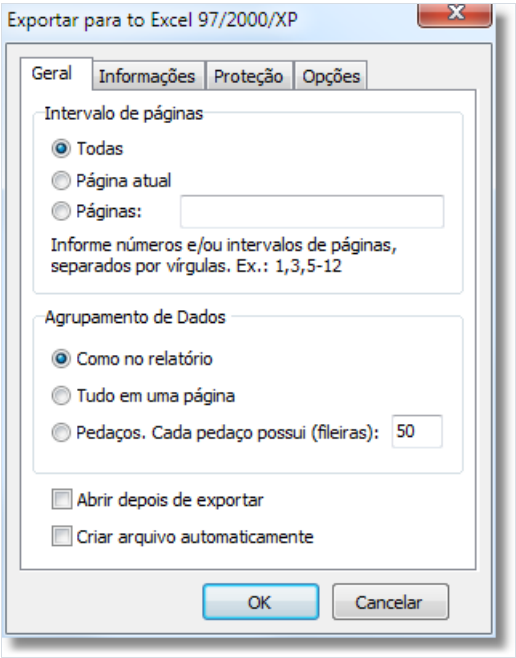
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação.

Exportar para Excel 97/2000/XP

Excel – um aplicativo para trabalhar com planilhas eletrônicas. É incluído no Microsoft Office.

O método de exportação é 'tabela'.

Ao exportar para o formato Excel, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Agrupamento de Dados:

- Como no relatório: cada página do relatório é exportada em uma planilha Excel separada;
- Tudo em uma página: gera um documento contínuo sem quebras de página e cabeçalhos/rodapés de página;
- Pedaços. Cada pedaço possui (fileiras): cada pedaço é exportado em uma planilha Excel separada.

Configurações de exportação:

- WYSIWYG: uma representação precisa do relatório visualizado; se esta opção for desabilitada, a otimização é habilitada, o que reduz o número de linhas e colunas no arquivo de exportação
- Figuras: inclui imagens gráficas exportadas na tabela de saída;
- Linhas da grade: ativa/desativa linhas de grade no Excel;
- Ajustar tamanho da página: ajusta o tamanho da célula ao conteúdo;
- Excluir fileiras vazias: exclui fileiras vazias da tabela de saída;
- Exportar fórmulas: se o texto de uma célula começar com o símbolo "=", ele é exportado como uma fórmula Excel;
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação.

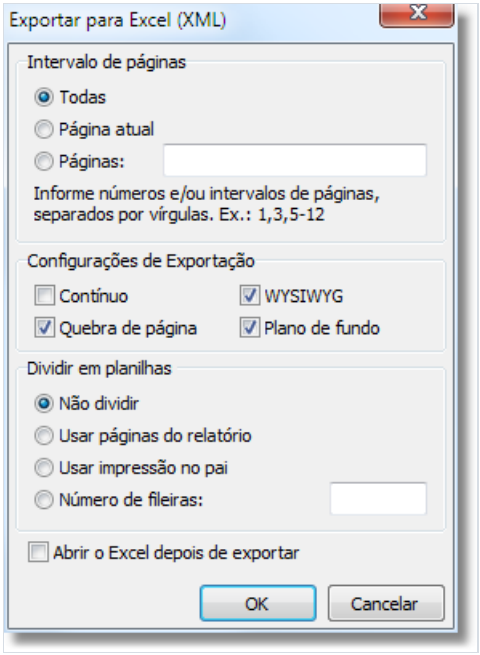
Recursos da exportação: objetos RichText são transferidos como texto simples, e a transferência de imagens gráficas é suportada.

Exportar para o formato Excel XML

XML (Extensible Markup Language) é uma linguagem de marcação extensível. XML foi projetada para o armazenamento estruturado de dados e também para o intercâmbio de dados entre programas diferentes. O FastReport usa o formato XML para a transferência de dados para o Excel versão 2003 e posterior.

O método de exportação é 'tabela'.

Ao exportar para o formato XML, é exibida uma caixa de diálogo para a configuração dos parâmetros do arquivo de saída.



Parâmetros de exportação:

- Contínuo: gera um documento contínuo, sem quebras de página e sem cabeçalhos/rodapés de página
- Quebra de página: habilita quebras de página no documento
- WYSIWYG: uma representação precisa do relatório visualizado; se esta opção for desabilitada, a otimização é habilitada, o que reduz o número de linhas e colunas no arquivo de exportação
- Plano de fundo: a cor de plano de fundo da página do relatório é exportada para a planilha
- Abrir o Excel depois de exportar: o arquivo exportado é aberto imediatamente após a exportação.

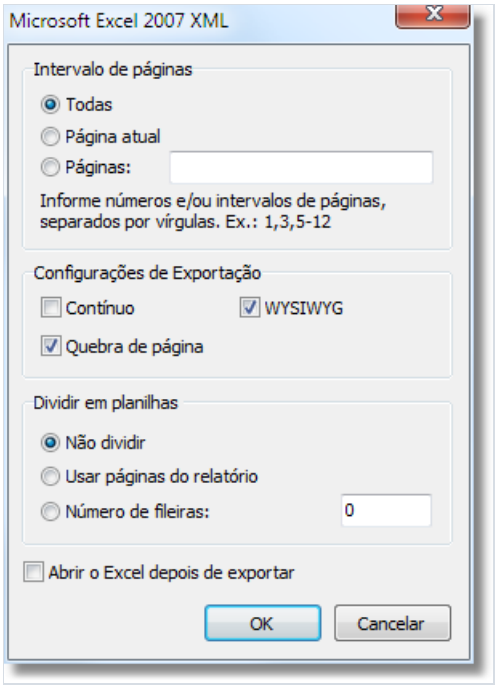
Recursos da exportação: objetos RichText são exportados como texto simples; imagens gráficas não são suportadas.

Exportar para Excel 2007

Excel 2007 é um aplicativo para trabalhar com planilhas eletrônicas. É incluído no Microsoft Office 2007.

O método de exportação é 'tabela'.

Ao exportar para o formato Excel, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Dividir em planilhas:

- Não dividir: todas as páginas do relatório são exportadas para uma única planilha Excel
- Usar páginas do relatório: cada página do relatório é exportada para uma planilha Excel separada
- Número de fileiras: cada conjunto de fileiras é exportado para uma planilha Excel separada

Parâmetros de exportação:

- Contínuo: gera um documento contínuo, sem quebras de página e sem cabeçalhos/rodapés de página
- Quebra de página: inclui quebras de página no documento resultante
- WYSIWYG: uma representação precisa do relatório visualizado; se esta opção for desabilitada, a otimização é habilitada, o que reduz o número de linhas e colunas no arquivo de exportação
- Abrir o Excel depois de exportar: o arquivo exportado é aberto imediatamente após a exportação

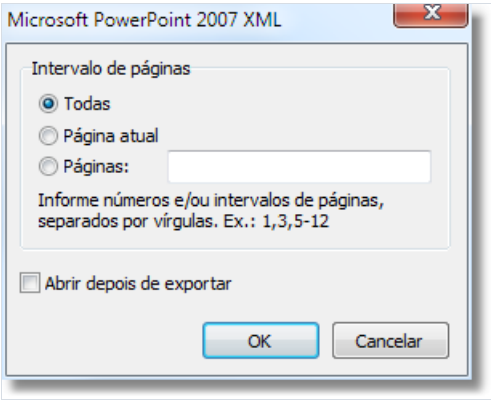
Recursos da exportação: objetos RichText são exportados como texto simples; imagens gráficas são suportadas.

Exportar para PowerPoint 2007

PowerPoint 2007 é um aplicativo para trabalhar com apresentações eletrônicas. Ele é incluído no Microsoft Office 2007.

O método de exportação é em camadas.

Ao exportar para o formato PowerPoint, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

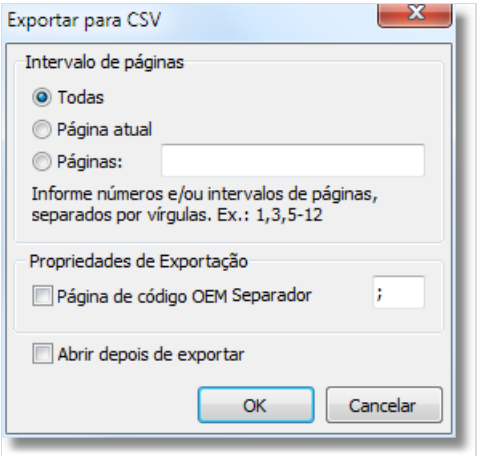
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação.

Exportar para o formato CSV

Arquivos CSV contêm dados em tabelas (números e texto) no formato texto sem formatação. Um separador especificado é inserido entre valores de colunas, e as fileiras começam em uma linha nova. Este formato pode ser aberto por diversos editores de tabelas/diagramas.

O método de exportação é 'tabela'.

Ao exportar para o formato CSV, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

- Página de código OEM: a codificação OEM do arquivo exportado
- Separador: o separador usado entre as colunas
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação usando o visualizador CSV padrão instalado no computador.

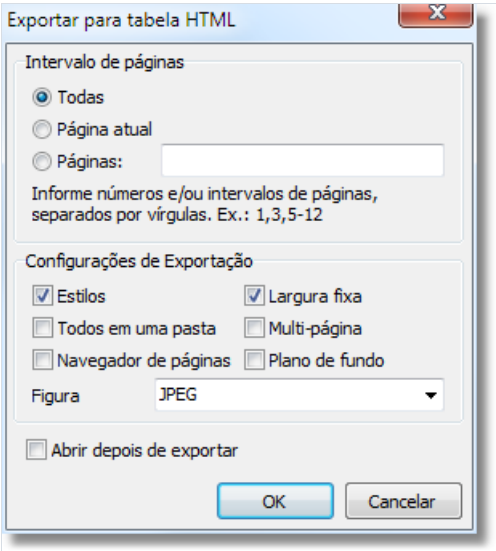
Recursos de exportação: nenhuma informação de layout (o design do relatório) é incluída no arquivo de exportação; imagens gráficas não são suportadas.

Exportar para o formato HTML

HTML (Hypertext Markup Language) é considerada a linguagem padrão para documentos na Internet. Ela é usada para criar documentos relativamente simples porém com bom design. HTML suporta links de hipertexto, assim como um layout de documento simples.

O método de exportação é 'tabela'.

Ao exportar para o formato HTML, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

- Estilos: estilos de design dos objetos texto exportados; quando desabilitados, a velocidade de exportação é maior, mas a aparência não é igual a uma planilha
- Todos em uma pasta: arquivos adicionais são salvos na mesma pasta do arquivo principal
- Navegador de páginas: um navegador especial para criar saltos rápidos entre páginas
- Largura fixa: bloqueia o ajuste de largura automático de tabelas/diagramas ao alterar o tamanho da janela de exibição
- Multi-página: cada página é exportada como um arquivo separado
- Plano de fundo: a imagem gráfica configurada como o plano de fundo da página é exportada para o arquivo HTML
- Figuras: imagens gráficas são exportadas para o arquivo HTML
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação usando o visualizador HTML padrão instalado no computador.

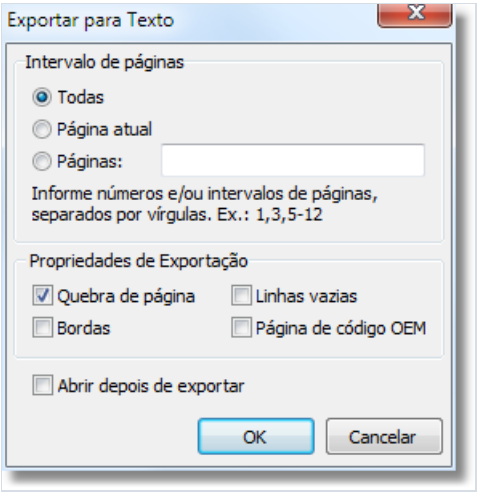
Recursos da exportação: a exportação pode ser composta de diversos arquivos; cada imagem gráfica é exportada para seu próprio arquivo; objetos RichText são exportados como texto simples; a precisão da renderização e do tamanho do arquivo dependem do design do relatório, consulte o capítulo "Considerações do design do relatório".

Exportar para o formato texto

Um arquivo de texto sem formatação sem nenhum gráfico.

O método de exportação é 'tabela'.

Ao exportar para o formato texto, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

- Quebra de página: exporta as quebras de página ao arquivo resultante
- Linhas vazias: habilita quebras de página no documento
- Bordas: habilita a exportação de bordas do objeto texto
- Página de código OEM: codificação OEM do arquivo exportado
- Abrir depois de exportar: o arquivo exportado é aberto imediatamente após a exportação usando o visualizador de texto padrão instalado no computador.

Recursos da exportação: nenhuma informação de layout (o design do relatório) é incluída no arquivo de exportação; imagens gráficas não são suportadas; a largura da página é configurada automaticamente em dependência ao tipo de objetos texto na página do relatório.

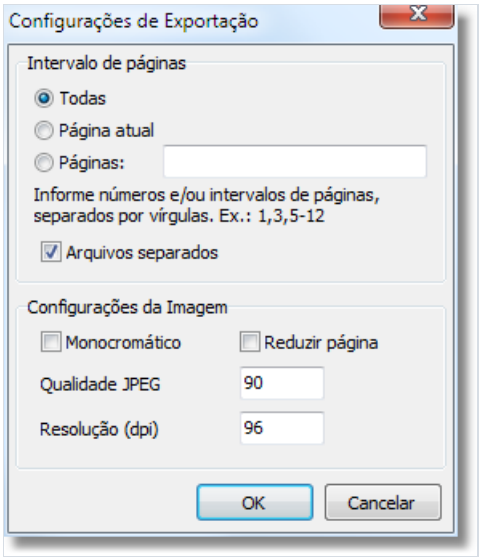
Exportar para os formatos gráficos Jpeg, BMP, Gif, Tiff

O FastReport exporta para formatos gráficos.

- JPEG (Joint Photographic Experts Group): um formato compactado baseado em um algoritmo que registra as diferenças entre os pixels. Ele é caracterizado pela compactação alta em detrimento à precisão gráfica.
- BMP (Windows Device Independent Bitmap): usado para armazenar imagens bitmap usadas no Windows. Um formato de arquivo padrão em computadores que utilizam o Windows.
- GIF (Graphics Interchange Format): um formato independente de hardware que foi desenvolvido para a transmissão de imagens bitmap através de redes. É apropriado para compactar conteúdo homogêneo (logos, inscrições, esquemas).
- TIFF, TIF (Target Image File Format): um formato independente de hardware. Hoje em dia é um dos formatos mais difundidos e confiáveis na transmissão de poligrafia e facsimile.

O método de exportação é 'envolver'.

Ao exportar para os formatos exibidos acima, uma caixa de diálogo é aberta que solicita as configurações do arquivo de saída.



Configurações de exportação:

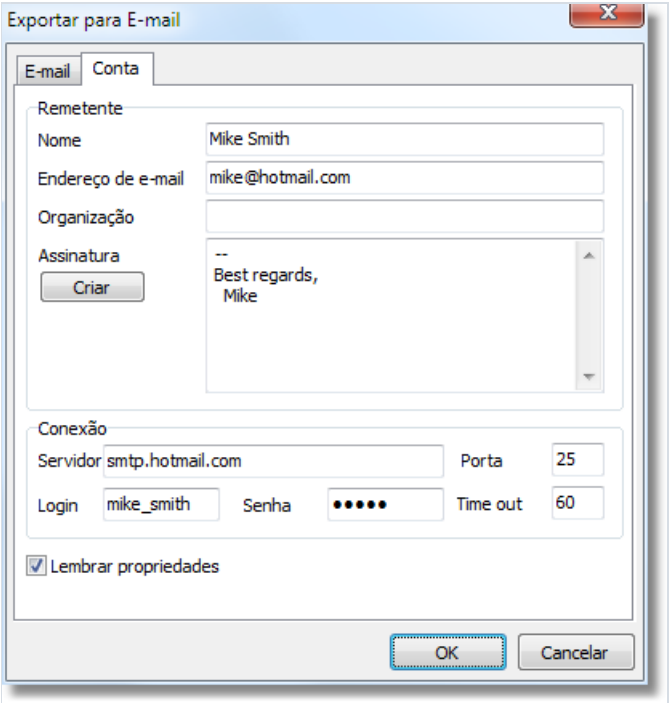
- Arquivos separados: quando habilitado, cada página do relatório é exportada para um arquivo separado; os nomes de arquivo são derivados do nome de arquivo especificado com um sufixo de um caractere de sublinhado + o número da página
- Monocromático: exporta como uma imagem monocromática
- Reduzir página: o espaço em branco nas bordas da página são cortados
- Qualidade JPEG: a taxa de compactação JPEG; somente é habilitada ao exportar para o formato JPEG
- Resolução: a resolução gráfica da imagem exportada

Recursos da exportação: se Arquivos separados for desabilitado, é criado um arquivo muito grande.

Enviar um relatório por e-mail

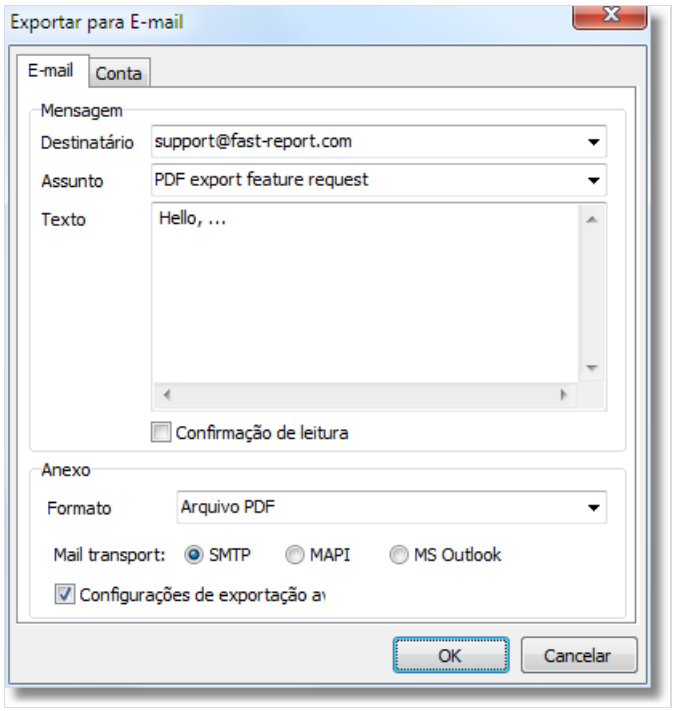
O FastReport pode enviar um relatório concluído por e-mail em qualquer formato necessário, sem precisar de um aplicativo de e-mail separado.

Ao exportar por e-mail, a caixa de diálogo de configuração de e-mail é aberta. Antes de exportar, é necessário configurar os detalhes da conta de e-mail do remetente na guia "Conta":



- Nome: o nome do remetente
- Endereço de e-mail: o endereço de e-mail do remetente
- Organização: a organização do remetente
- Assinatura: a assinatura do e-mail; ela pode ser gerada ao clicar no botão "Criar" depois de preencher os campos anteriores
- Host: servidor SMTP
- Porta: porta do servidor SMTP
- Login: nome de acesso para a autorização no servidor SMTP, se requerido pelo servidor SMTP especificado
- Senha: senha para a autorização, se requerido pelo servidor SMTP especificado
- Lembrar propriedades: lembrar de todos os campos para uso posterior

Após preencher os campos necessários na guia "Conta", é necessário preencher os campos de mensagem na guia "E-mail":



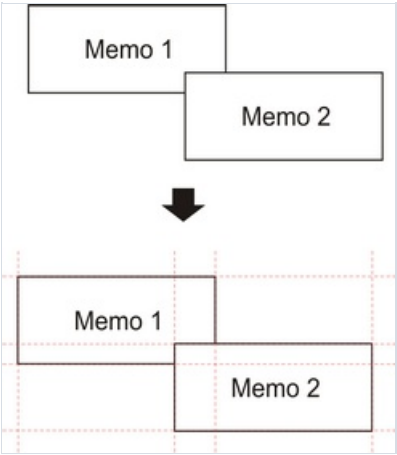
- Destinatário: endereço de e-mail do destinatário; é possível selecionar endereços usados anteriormente na lista suspensa
- Assunto: o assunto da mensagem; é possível selecionar assuntos usados anteriormente na lista suspensa
- Texto: o texto da mensagem
- Formato: o formato do relatório anexado ao e-mail; selecione um dos formatos disponíveis; o formato FastReport (FR3) também pode ser selecionado
- Configurações de exportação avançadas: quando habilitadas, o botão OK abre a caixa de diálogo das configurações do formato de exportação apropriada; quando desabilitadas, as configurações de exportação padrão são usadas

Recursos da exportação por e-mail: somente a autenticação simples nos servidores SMTP é suportada. Se a autenticação não for requerida, não é necessário inserir o "Login" e "Senha".

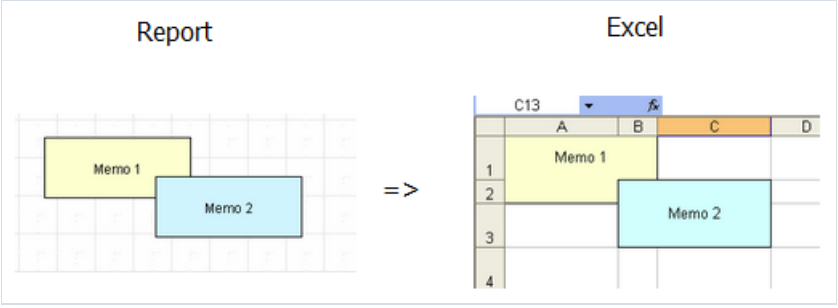
Considerações do design do relatório

A qualidade da saída do relatório em qualquer formato depende da qualidade do design do relatório original. O FastReport pode manipular objetos de muitas maneiras durante a criação do relatório. Isso fornece a vantagem do desenvolvimento rápido de relatórios e sua impressão posterior. Documentos impressos têm a aparência exata da visualização. Este é o objetivo principal do mecanismo do FastReport. A desvantagem dessa liberdade de desenvolvimento é a complexidade da exportação de documentos FastReport para outros formatos de dados, dos quais cada um possui seus próprios requerimentos e limitações, que às vezes podem ser complexos. Neste capítulo discutiremos os requerimentos de design especiais de relatórios que precisam ser exportados.

Muitos formatos usam uma apresentação de dados em tabela, como HTML, XLS, XML, RTF e CSV. Ao contrário da liberdade permitida no design da página no FastReport, ao exportar para estes formatos, as células de saída criadas para estas tabelas não podem ter interseções ou serem arranjadas em camadas. Os filtros de exportação normalmente levam esses requerimentos em conta quando os objetos são exportados pelo FastReport, através da utilização de algoritmos especiais que lidam com interseções e o melhor posicionamento de células. Nas interseções de objetos são criadas novas linhas e colunas na tabela de saída. Isso é necessário para permitir que o FastReport posicione os objetos de maneira exata e para obter a maior correspondência com a página de visualização original. Um grande número de objetos com interseção no design de um relatório resulta em um grande número de colunas e fileiras adicionais na tabela de saída. Isso, por sua vez, pode levar à necessidade de editar o arquivo exportado em seu próprio editor antes que ele possa ser utilizado para outros fins.

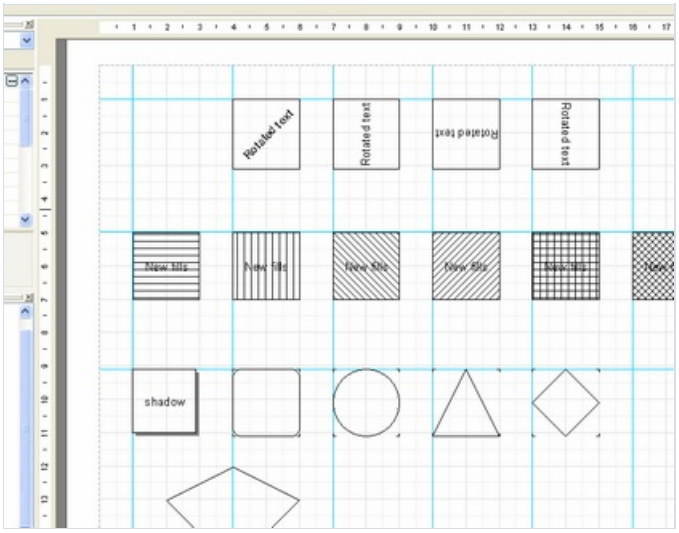


Por exemplo, veja um relatório onde o design possui uma sobreposição pequena entre dois objetos na mesma banda, e o número de registros no relatório é 150. Ao exportar para o formato RTF, 450 linhas serão criadas (150 fileiras para cada objeto e 150 fileiras para a interseção). Se removermos a sobreposição, haveria apenas 300 fileiras. Em relatórios grandes com um grande número de objetos, a diferença seria muito maior. Isso, é claro, afeta o tamanho do arquivo de saída.



Tenha isso em mente ao criar relatórios que devem ser exportados para qualquer um dos formatos que usa o método de saída de 'tabela'.

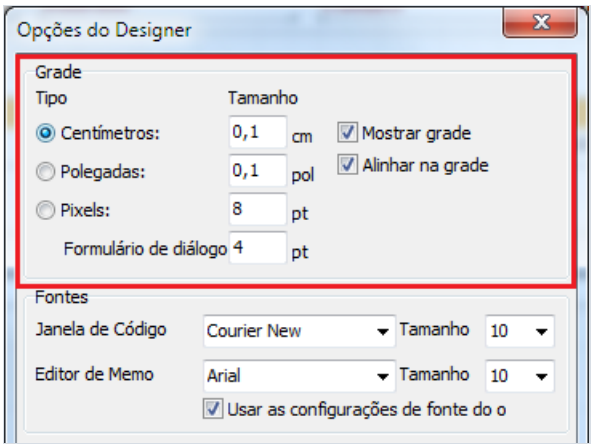
Ao fazer o design de tabelas nos relatórios, fique ciente das bordas de células adjacentes. É importante que células não se sobreponham e não sejam arranjadas em camadas. O algoritmo de exportação pode lidar com as células de maneira inesperada e fornecer um resultado que não é o desejado. É melhor arranjar objetos de maneira que sejam colocados em linha tanto verticalmente como horizontalmente. As linhas guia podem ajudar com isso.



Usando linhas guia no designer

Para usar linhas guia no designer do FastReport, clique na régua horizontal ou vertical no topo ou no lado esquerdo da página do relatório e arraste a régua até a posição desejada na página, onde a linha guia será exibida. Então você poderã posicionar objetos que estejam alinhados com essas linhas guia horizontais e verticais.

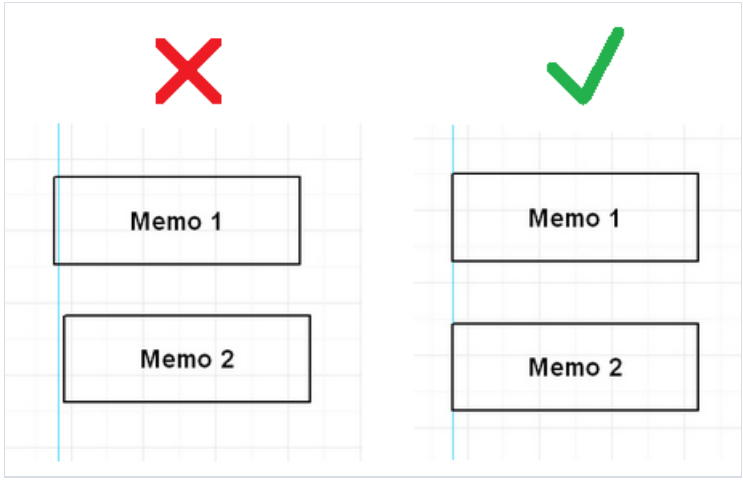
O alinhamento da grade também pode ajudar no posicionamento de objetos "Texto", para evitar sua sobreposição. A grade é habilitada nas Opções do designer, onde também é possível ajustar a densidade: "Exibir > Opções... > Grade".



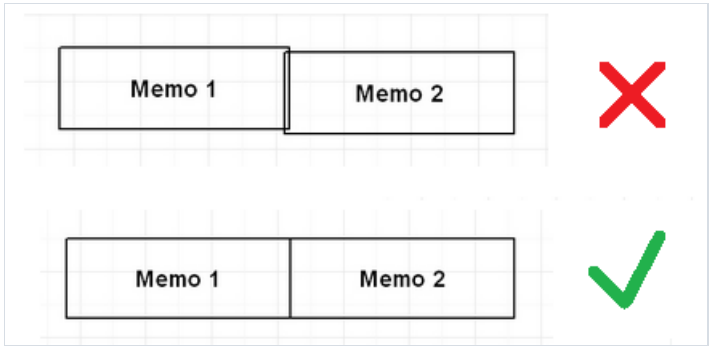
Ao usar bordas em objetos "Texto", é melhor usar as propriedades de borda do objeto em vez de adicionar objetos gráficos como linhas e retângulos ao redor do texto. Também tente não usar objetos no plano de fundo, embaixo de objetos de texto transparentes.

Manter essas regras simples em mente ajudará você a criar um relatório que terá uma aparência perfeita após ser exportado para um formato que utilize o método de saída baseado em tabela.

Abaixo seguem alguns exemplos de um arranjo bom e ruim de objetos.



Os objetos são deslocados horizontalmente – eles não estão alinhados horizontalmente com a linha guia vertical.



Os objetos estão sobrepostos – ao exportar para um formato de tabela/diagrama, serão criadas fileiras e colunas adicionais e desnecessárias, além de três células adicionais na área de sobreposição.

Recomendamos o estudo dos relatórios de demonstração incluídos na instalação do FastReport para ajudar você a dominar os princípios básicos do bom design de relatórios.