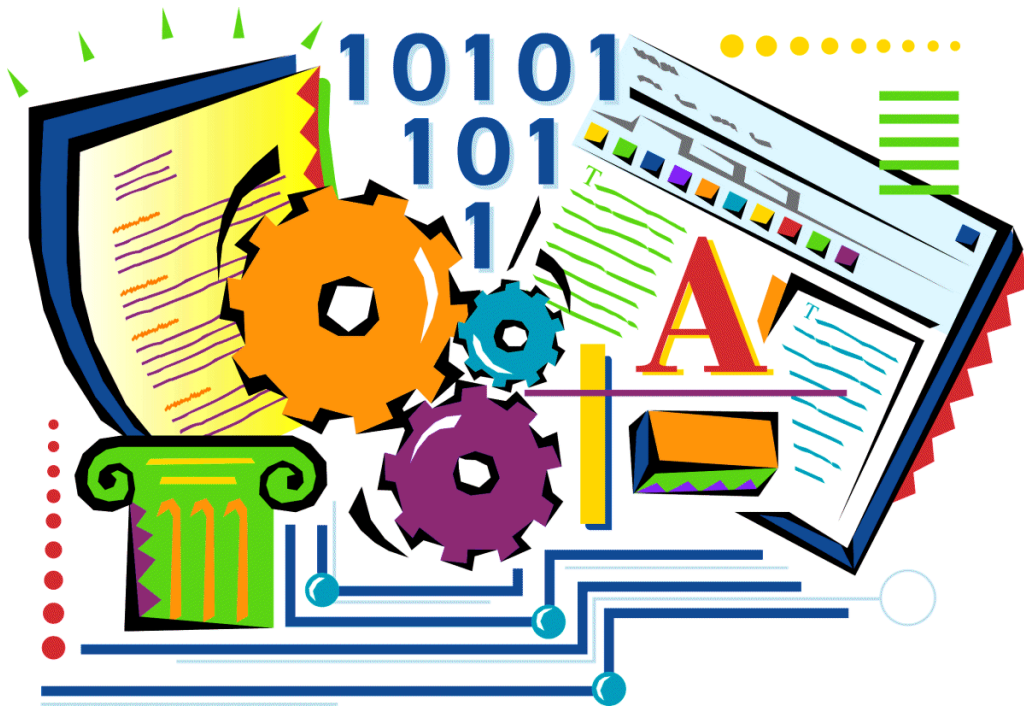


ReportBuilder



Guia do desenvolvedor
Quarta edição

ReportBuilder

Guia do desenvolvedor

Quarta edição

Copyright © 2000-2007 da Digital Metaphors Corporation

CONTENTS

eUNTRDUÇÃO

O básico

Criação de relatório	3
Selecione	3
Projeto	4
Processar	4
Geração	5

Um teste rápido

Visão geral	7
Criar uma nova aplicação	7
Criar uma tabela, fonte de dados e componente de pipeline de dados	7
Crie um relatório e conecte-o aos dados	7
Chame o Report Designer	7
Coloque um componente de etiqueta na faixa do cabeçalho	7
Coloque um componente DBText na faixa de detalhes	8
Visualize o relatório em tempo de design	8
Visualize o relatório em tempo de execução	8

A melhor maneira de aprender ReportBuilder?

Comece de forma simples e, em seguida, vá para o próximo nível fácil.	11
Use a ajuda online.	11
Use este manual.	11
Execute os exemplos; Estude os exemplos; Conheça os exemplos.	11

Elementos da interface do usuário

O Report Designer	13
A árvore de relatórios	14
O Inspetor de Objetos	15
A árvore de dados	0,15

Trabalhando com o Report Designer

Visão geral	17
Algumas dicas para ajudá-lo a obter o máximo do ReportBuilder:	17

Noções básicas de relatórios

Consultar tabelas / consultas	19
Filtrando dados	0,19
Executando Cálculos	19
Formatos de exibição	20
Bandas dinâmicas	20
Esticando memorandos e formas	20
Controlando a visibilidade do componente	20

REPORTBUILDER FSEM DANOS

MAIN

Introdução

Visão geral	25
Layout do relatório	25
Saída de relatório	25
Acesso de dados	25
Processo de Dados	0,25

Os Componentes Delphi

DBPipeline	27
BDEPipeline	0,27
TextPipeline	0,27
JITPipeline	27
Relatório	27
Visualizador	27
Leitor de arquivo	27

Componentes de relatório

Visão geral	29
Memo	29
RichText	29
Rótulo	29
SystemVariable	30
Variável	30
Imagem	30
Forma	30
TeeChart	30
BarCode	30
2D BarCode	30
CheckBox	30
DBText	30
DBMemo	30
DBRichText	31
DBCalc	31
DBImage	31
DBBarCode	31
2D DBBarCode	31
DBTeeChart	31
DBCheckBox	31
Região	31
Sub-relatório	31
CrossTab	31
Quebra de página	31

Layouts inteligentes

Visão geral	33
Âncoras	33
StretchWithParent	33
ShiftRelativeTo	33
StopPosition (para sub-relatórios)	33
BottomOffset	0,33
OverflowOffset	33
ReprintOnOverflow	33
Um memorando na faixa de detalhes	34
Um Memorando com um Fundo de Forma	34
Um memorando com etiqueta abaixo	35
Dois memorandos empilhados na faixa de detalhes	35
Dois memorandos lado a lado com rótulos abaixo	36
Sub-relatórios filhos em posições fixas	36
Um memorando com dois memorandos lado a lado abaixo	37

SubReports

Visão geral	39
Conjunto de dados único	39
Conjunto de dados mestre com conjunto de dados de detalhe único	40
Conjunto de dados mestre com conjuntos de dados de detalhes aninhados	40
Conjunto de dados mestre com vários conjuntos de dados independentes	41
Conjuntos de dados independentes	41

Emulação de Formulário

Visão geral	43
Formulários de página única	43
Formulários de várias páginas	45

DATA

Introdução

Visão geral	49
BDEPipeline	49
DBPipeline	49
TextPipeline	49
JITPipeline	49
Fornecendo Dados	49
Controlando Traversal de Dados	50

Suporte de banco de dados

Visão geral	53
Acesso de dados	53

Alternativas de banco de dados

Visão geral	55
ALVOROÇO	55
Incluído no Delphi. Usado para conectar ao MS SQL Server e MS Access	55
IBExpress	55
Incluído no Delphi. Usado para conectar ao Interbase e Firebird.	55
DBExpress	55
Acesso direto ao Oracle (DOA)	55
Usado para se conectar ao Oracle.	55
mySQLDAC	55
Vantagem	55
DBISAM e ElevateDB	55
NexuxDB	55

Arquivos de Texto

O componente TextPipeline	57
O Editor de Campo	57

Objetos Delphi

Visão geral	59
JITPipeline	59

Acesso nativo a dados proprietários

Visão geral	61
-------------------	----

CTRIBUTO

O modelo de evento Delphi

Visão geral	65
Significância	65
Tempo	65
Eventos ReportBuilder	65
Report.OnIntializeParameters	65
Report.BeforeOpenDataPipelines	65
Band.BeforePrint	66
Variable.OnCalc	66

Configuração Dinâmica

Configurar relatórios durante a geração	67
Cor da fonte	67
Concatenação	68
Address Squeeze	69
Grupo continuado	70
Regiões	71
SubReports	72

Executando Cálculos

Visão geral	73
Componente DBCalc	73
Componente variável	73
Contar	74
Contagem Mestre / Detalhe	75
Total do grupo	76
Total geral	77
Soma cumulativa	78
Total condicional do grupo	79
Total geral condicional	80
Look Ahead Total	81

Criação de relatórios em código

Codificando um Relatório	83
--------------------------------	----

DESIGN

O Report Designer

Guia Design	89
Guia de visualização	89

Diálogos

Imprimir diálogo	91
Diálogo de configuração de página	92
Diálogo de Grupos	93
Imprimir para caixa de diálogo de configuração de arquivo	93
Diálogo de dados	94
Opções de grade	94
Configurações de esboço	94
Encontrar configurações de texto	94

Barras de Ferramentas

Visão geral	95
A árvore do relatório	95
A árvore de dados	95
Paleta de componentes padrão	96
Paleta de componentes de dados	97
Paleta de componentes avançados	98
Barra de ferramentas padrão	98
Editar barra de ferramentas	99
Barra de ferramentas de formatação	100
Alinhar ou Barra de Ferramentas de Espaço	101
Barra de ferramentas de tamanho	102
Barra de ferramentas Nudge	102
Barra de Ferramentas Desenhar	103

Suporte para arrastar e soltar

Visão geral	105
-------------------	-----

O Assistente de Relatório

Visão geral	107
Assistente de Relatório: Criar um Relatório Simples	107
Assistente de relatório: Criar um relatório baseado em grupo	110

PRINT

Introdução

Geração	115
---------------	-----

Visualizando

Visualizar impressão	117
Impressão	117
E-mail	117
Encontrar texto	117
Controles de zoom	117
Controles de navegação	117

Configurações de impressão personalizadas

Visão geral	119
BinName	119
Collation	119
Cópias	119
Nome do Documento	119
Duplex	119
Margens	119
Orientação	119
PaperHeight & PaperWidth	119
PaperName	0,19
Nome da impressora	119

PDF

Imprimir em PDF	121
Imprimindo diretamente para arquivo PDF	121
Permitir que o usuário final imprima em PDF	121
Imprimindo diretamente para fluxo de PDF	121

Arquivamento de relatórios

Visão geral	123
Arquivando um Relatório	123
Usando o ArchiveReader	123

Imprimir em texto ASCII

Visão geral	125
Imprimir diretamente para o arquivo de texto ASCII	125
Permitir que o usuário final imprima no arquivo de texto ASCII	125

Arquivo de texto de emulação de relatório

Visão geral	127
Imprimir diretamente no arquivo de texto de emulação de relatório	127
Permitir que o usuário final imprima em um arquivo de texto de emulação de relatório	127

RTF, HTML e outros formatos

Visão geral	129
TExtraDevices	129
PsRBExportDevices	129
IIPDFLib	129
Gnostice eDoc Engine	129

Relatórios de e-mail

Visão geral	131
Habilitar opção de e-mail para o visualizador	131
Abra o cliente de email com PDF anexado	131
Enviar e-mail sem interação do usuário	131
Envie e-mail diretamente usando Indy	131

DEPLOY**Introdução**

Visão geral	135
Implementação de aplicativos	135
Implementação de relatório	135

Modelos de relatório

Visão geral	137
Modelos baseados em arquivo	137
Modelos de banco de dados	138
Carregando e salvando relatórios	138

Como um EXE

Visão geral	139
-------------------	-----

Como pacotes

Visão geral	141
-------------------	-----

Suporte de idioma internacional

Visão geral	143
O idioma padrão	144
Traduções personalizadas	145

REPORTBUILDER ENTERPRISE EDIÇÃO

FSEM DANOS

MAIN

Introdução

Visão geral	151
Dados	151
Calc	152
Projeto	153
Pré-visualizar	153

Os Componentes Delphi

Visão geral	155
DBPipeline	155
BDEPipeline	155
TextPipeline	155
JITPipeline	155
Relatório	155
Visualizador	155
ArchiveReader	155
Dicionário de dados	156
Designer	156
ReportExplorer	156

Componente de Designer

Visão geral	157
AllowDataSettingsChange	157
AllowSaveToFile	157
DataSettings	157
Ícone	157
InitStorageType & IniStorageName	157
RAPInterface	157
RAPOptions	157
Relatório	157
ShowComponents	157
TabsVisible	0,157
Uma solução simples de relatório de usuário final	158

Report Explorer

Visão geral	161
Designer	161
FolderFieldNames	161
FolderPipeline	161
ItemFieldNames	161
ItemPipeline	161
ModalForm	161

DATA

Dicionário de dados

Visão geral	163
FieldFieldNames	163
FieldPipeline	163
TableFieldNames	163
TablePipeline	163

Juntando tudo

Resumo	165
--------------	-----

Ajuda online

Visão geral	167
-------------------	-----

Introdução

Visão geral	171
Exibição de dados	171

Assistente de Consulta

Visão geral	173
Assistente de Consulta: Criar uma Visão de Dados Baseada em Consulta Simples	173

Designer de Consulta

Visão geral	177
Designer de Consulta: Adicionando Critérios de Pesquisa	177
Criar uma soma de grupo	179
Concatenar Campos	182
Editar SQL	184

Configurando DADE

Visão geral	185
SessionType	185
AllowEditSQL	185
Nome do banco de dados	185
Dicionário de dados	185
UseDataDictionary	185
SQLType	185
DatabaseType	185
CollationType	185
GuidCollationType	185
IsCaseSensitive	185
Configurações de dados	186

Arquitetura DADE

Visão geral	187
Classes de base	187
DADE Plug-in Classes	188

Estendendo DADE

Visão geral	189
Modelos de Dataview	189
Modelo de Dataview: A Visão do Usuário Final	189
Modelo de Dataview: A Implementação	191
Suporte para produtos de banco de dados	191
Aulas	192

CTRIBUTO

Introdução

Visão geral	195
Sobre RAP	195

O Calc Workspace

Visão geral	197
O Code Explorer	198
O Editor de Código	200
Janela de mensagem	201
A caixa de ferramentas de código	201

Escrevendo código RAP

Programas em RAP	205
Codificando um manipulador de eventos	205
Compilando manipuladores de eventos	206
Declarando Variáveis Locais	206
Declarando constantes locais	206
Declarando Variáveis Globais	206
Declarando constantes globais	207
Declarando procedimentos e funções globais	207
Parâmetros de procedimento e função	208
Procedimentos e funções de chamada	208

Configurando RAP

Configurações de usuário final	209
--------------------------------------	-----

Estendendo RAP

Funções de passagem RAP	213
Adicionando funções à caixa de ferramentas de código	213
Adicionando classes e propriedades via RTTI	214

Depuração de código RAP

Suporte CodeSite	215
Usando as funções CodeSite	216
Compilando condicionalmente o suporte CodeSite	217

DESIGN

O Report Explorer

Barra de ferramentas do Report Explorer	221
---	-----

PRINT

Opções do usuário final

Visão geral 225

Aplicativos de usuário final 225

DEPLOY

Resumo

Visão geral 229

Implementação de aplicativos 229

Implementação de relatório 229

REPORT TUTORIAIS

Criação de um relatório por meio da árvore de dados

Visão geral	233
Criar uma pasta de tutorial	233
Criar uma nova aplicação	233
Criar uma tabela, fonte de dados e componente de DataPipeline	234
Crie um relatório e conecte-o aos dados	234
Chame o Report Designer e defina a orientação do papel	234
Defina a altura da banda do cabeçalho para 1 polegada	234
Criar rótulos na faixa de cabeçalho	235
Use a árvore de dados para organizar o	236
Informação ao Cliente	236
Ajuste o layout	237
Adicionar um TimeStamp	238
Adicionar números de página	238
Visualize o relatório em tempo de execução	239

Criação de um relatório por meio do assistente de relatório

Visão geral	241
Criar uma nova aplicação	241
Criar uma tabela, fonte de dados e componente de DataPipeline	242
Crie um relatório e conecte-o aos dados	242
Chame o Report Designer e acesse o Report Wizard	242
Use o Assistente de Relatório para Organizar um Relatório de Estilo Tabular	242
Visualize o relatório	243
Defina o relatório para o modo de duas passagens	243
Salvar o layout do relatório em um arquivo de modelo	244
Use o Assistente de Relatório para Organizar um Relatório de Estilo Vertical	244
Modificar o layout do relatório para conter colunas	245
Salvar o layout do relatório em um arquivo de modelo	245
Adicionar a interface de tempo de execução	246
Componentes	246
Codifique os manipuladores de eventos	247
Compilar e executar o aplicativo	248

Um relatório simples da maneira mais difícil

Visão geral	251
Criar uma nova aplicação	251
Criar uma tabela, fonte de dados e componente de DataPipeline	252
Crie um relatório e conecte-o aos dados	252
Chame o Report Designer e defina a orientação do papel	252
Esquematize a etiqueta do título da lista de clientes na faixa do cabeçalho	252
Layout da etiqueta de título da empresa na faixa de cabeçalho	253
Criar etiquetas para a faixa de cabeçalho	254
Complete o Layout da Faixa do Cabeçalho	254
Esquematize a faixa de detalhes	255
Alinhe as etiquetas da faixa do cabeçalho verticalmente	256
Alinhe as etiquetas da faixa do cabeçalho com os componentes DBText da faixa de detalhe	256
Set PrintDateTime	256
Definir números de página	257
Visualize o relatório em tempo de execução	257

Grupos, cálculos e a faixa de resumo

Visão geral	259
Criar uma nova aplicação	259
Criar um componente Query, DataSource e DataPipeline	260
Crie um relatório e conecte-o aos dados	260
Chame o Report Designer	260
Criar rótulos na faixa de cabeçalho	260
Crie um grupo no campo 'rcmndation'	261
Esquema a faixa do cabeçalho do grupo	261
Codifique o Evento BeforeGenerate da Banda do Cabeçalho do Grupo	262
Adicionar dados gerais à faixa de detalhes	263
Adicione os dados de estatísticas vitais à faixa de detalhes	264
Adicione os dados de preços à faixa de detalhes	265
Alinhar os componentes de dados de preços	265
Adicione os dados de recomendação à faixa de detalhes	266
Alinhar os componentes de dados de recomendação	0,266
Adicione o símbolo de ações e os dados da empresa à faixa de detalhes	267
Esquematize a faixa de rodapé	268
Esquema a Faixa de Resumo	269
Ajuste as etiquetas de banda de resumo	270
Criar e ajustar componentes variáveis	270
Atribuir manipuladores de eventos aos eventos OnCalc dos componentes variáveis	271
Criar o total geral	272
Pré-visualizar em tempo de execução	273

Usando regiões para agrupar componentes dinâmicos de maneira lógica

Visão geral	275
Criar uma nova aplicação	275
Criar uma tabela, fonte de dados e componente de DataPipeline	276
Crie um relatório e conecte-o aos dados	276
Chame o Report Designer	276
Configure as bandas de cabeçalho e rodapé	276
Crie um grupo no campo 'Categoria'	277
Criar uma região de imagem	277
Ajustar os componentes da região da imagem	278
Criar uma região de memorando	279
Criar uma região de campos	279
Ajuste os componentes DBText	280
Adicionar um manipulador de eventos de codificação de cores	281
Visualize o relatório em tempo de execução	282

Emulação de formulários com uma imagem WMF

Visão geral	285
Criar uma nova aplicação	285
Criar uma tabela, fonte de dados e componente de DataPipeline	286
Crie um relatório e conecte-o aos dados	286
Criar campos calculados	286
Configure o tamanho da página e bandas	287
Criar um componente de imagem	287
Criar os componentes Wages DBText	288
Criar o DBText de retenção	289
Componentes	289
Crie as informações de endereço	289
Criar componente duplicado	290
Em formação	290
Escreva a rotina de 'aperto de endereço'	291
Visualize o relatório em tempo de execução	292

Mestre - Relatório Detalhado

Visão geral	293
Criar uma nova aplicação	293
Crie a tabela, fonte de dados e linha de dados para a tabela mestre	294
Crie a tabela, fonte de dados e linha de dados para a tabela de detalhes	294
Crie a tabela, fonte de dados e linha de dados para a tabela de pesquisa	294
Definir o relacionamento entre o cliente e a tabela de pedidos	294
Definir a relação entre o pedido e a tabela de funcionários	295
Crie um relatório e conecte-o aos dados	295
Crie as etiquetas de banda de cabeçalho	296
Crie a forma da faixa do cabeçalho	297
Use a árvore de dados para completar o layout da faixa do cabeçalho	297
Criar um grupo no campo CustNo	298
Comece o Layout de Informações do Pedido	299
Preencher o Layout de Informações do Pedido	300
Organizar os componentes de informações de envio	0,301
Layout dos componentes de informações de pagamento	0,301
Complete a faixa de detalhes	302
Esquema a Faixa de Resumo	303
Esquematize a faixa de rodapé	304
Visualize o relatório em tempo de execução	304

Mestre - Detalhe - Relatório Detalhado

Visão geral	307
Criar uma nova aplicação	307
Atualizar o nome do relatório	308
Adicione a tabela, fonte de dados e linha de dados para a tabela de detalhes de itens	308
Definir a relação entre o pedido e a tabela de itens	308
Criar a Tabela, DataSource e DataPipeline para a Tabela de Consulta de Peças	309
Definir a Relação entre o Item e a Tabela de Peças	309
Atualizar o título do relatório	309
Crie o SubReport2 e conecte-o aos dados	309
Remova o título e a faixa de resumo	310
Crie um grupo no campo 'OrderNo'	311
Esquematize a faixa do cabeçalho do grupo para o sub-relatório	311
Comece a organizar os componentes de informações de itens para o sub-relatório	312
Preencher o Layout de Informações de Itens	313
Complete o Layout da Faixa de Detalhe para o Sub-relatório	314
Esquema a faixa de rodapé do grupo para o sub-relatório	314
Codifique os cálculos para os totais	315
Preencha o Layout para SubReport1	316
Converter a faixa de título em uma faixa de cabeçalho de grupo	316
Visualize o relatório em tempo de execução	317

Visualização interativa com sub-relatórios de detalhamento

Visão geral	319
Criar uma nova aplicação	319
Chame o Report Designer e configure o Drill-Down	320

Conectando relatórios com sub-relatórios de estilo de seção

Visão geral	323
Criar uma nova aplicação	323
Transfira o Relatório de Lista de Clientes para o Formulário	324
Transfira o Relatório de Resumo de Estoque para o Formulário	324
Criar e configurar o relatório principal	325
Crie o sub-relatório da lista de clientes	325
Crie o Sub-relatório de resumo de estoque	326
Visualize o relatório em tempo de execução	327
Copie os manipuladores de eventos do relatório de resumo de estoque	327
Reanexe os manipuladores de eventos ao sub-relatório de resumo de estoque	328
Adicionar uma página de título ao relatório	329
Comece a organizar a faixa de título	329
Complete o Layout da Faixa de Título	330

Usando colunas para criar etiquetas de endereçamento

Visão geral	333
Criar uma nova aplicação	333
Chame o Report Designer e configure o layout da página	334
Configure o relatório	334
Criar etiquetas de endereçamento por meio do assistente de modelo de etiqueta	335

Impressão para um arquivo de texto

Visão geral	337
Criar uma nova aplicação	337
Complete o Tutorial do Assistente de Relatório	337
Carregar o modelo de relatório de estilo tabular	337
Atribuir nomes de usuário aos componentes DBText	338
Especifique o nome do arquivo e formato	338
Configure o relatório para imprimir em arquivo	338
Codifique o nome do modelo	338
Codifique o Nome do Arquivo de Texto	339
Imprimir o relatório para arquivo	339

Impressão de um arquivo de texto

Visão geral	341
Criar uma nova aplicação	341
Crie o componente Text DataPipeline	341
Defina os campos de dados para o pipeline de texto	341
Crie um relatório e conecte-o aos dados	342
Teste a conexão de dados	342
Carregar o modelo de relatório de estilo vertical	342
Codifique o nome do arquivo de texto	343
Visualize o relatório em tempo de execução	343

Usando o JITPipeline para imprimir de um StringGrid

Visão geral	345
Criar uma nova aplicação	345
Renomear TextPipeline	346
Crie um StringGrid	346
Adicionar código para carregar o StringGrid	346
Crie o componente JIT DataPipeline	347
Defina os campos de dados para o JITPipeline	347
Adicionar uma função para retornar os valores do campo da grade	347
Adicionar manipuladores de eventos JITPipeline para retornar os valores de campo	348

Usando o componente Rich Text para mala direta / mesclagem

Visão geral	349
Criar uma nova aplicação	349
Criar uma tabela, fonte de dados e componente de DataPipeline	349
Crie um relatório e conecte-o aos dados	350
Chame o Report Designer e defina o layout da página	350
Modificar as bandas	350
Carregar MMLetter no componente RichText	351
Adicionar campos à carta	351
Visualize o relatório em tempo de execução	352

Criando uma Crosstab

Visão geral 353

Criar uma nova aplicação 353

Criar um componente Query, DataSource e DataPipeline 353

Criar um relatório 354

Projetar a tabela de referência cruzada 355

Adicionar valores adicionais à tabela de referência cruzada 355

Defina o formato dos valores 355

Calcular totais por estado 356

Dispor a faixa de cabeçalho 356

Definir paginação 356

Use legendas repetidas 356

RAP TUTORIAIS

Campos de Concatenação

Visão geral	359
Crie um DataView	359
Layout do relatório	360
Navegue pelo Calc Workspace	360
Adicione o código de concatenação	361
Compilar e visualizar	362
Salvar o relatório	362
Carregar o relatório	362

Componentes de codificação de cores

Visão geral	363
Abra o Projeto Demo	363
Habilitar DetailBeforePrint	363
Adicionar Variáveis Locais	364
Visualizar e salvar	365

Duplex Dinâmico

Visão geral	367
Habilitar RAP	0,367
Crie um novo DataView	367
Criar Layout de Relatório	368
Compilar e visualizar	369

Adicionando funções ao RAP

Visão geral	371
Criar uma nova unidade	371
Editar cláusula de uso	371
Adicionar TraSystemFunction Descendent para Categoria	372
Adicionar nova classe de função	373
Implementar ExecuteFunction	373
Implementar GetSignature	373
Implementar HasParams	374
Adicionar inicialização	374
Adicionar myRapFuncs a um projeto	375
Veja ApplicationFilename na Code Toolbox	375
Adicionar ApplicationFilename ao código	376

Estendendo o RAP RTTI

Visão geral	377
Declare TmyTDataBaseRTTI	378
Implementar GetPropList	378
Implementar RefClass	379
Implementar GetPropRec	379
Implementar CallMethod	380
Implementar GetParams	380
Implementar GetPropValue	381
Implementar SetPropValue	381
Registrar a aula	381
Criar Função de Acesso	382
Adicionar myRapClass a um projeto	384
Escrever código RAP	385
Visualize o relatório	386

Imprimindo uma descrição dos critérios de pesquisa automática

Visão geral	387
Criar critérios de pesquisa	388
Escreva o código	388
Compilar e visualizar	389

Exibindo Formulários Delphi do RAP

Visão geral	391
Adicionar componentes	391
Criar uma nova unidade	393
Adicionar unidades ao projeto	397
Crie um DataView	397
Layout do relatório	397
Declare Global Variables	397
Implementar Global OnCreate	398
Evento Antes de Impressão de Cabeçalho de Grupo	399
Compilar e visualizar	401

UMAPPLICATION TUTORIAIS

Construindo um aplicativo de relatório

Visão geral	405
Criar uma nova aplicação	405
Crie um ListBox e um botão de visualização no formulário principal	405
Adicionar um formulário ao projeto	406
Criar uma classe de formulário ancestral	406
Tornar o formulário de relatório do tutorial um descendente de TrbReportForm	407
Preencher a caixa de listagem no evento OnCreate do formulário principal	408
Codifique o procedimento LaunchReport no formulário principal	408
Conecte o procedimento LaunchReport ao ListBox e ao botão Visualizar	409
Criar um formulário de visualização de impressão personalizado	410
Adicione o formulário de visualização de impressão personalizado ao seu projeto	410

Criação de um aplicativo de relatório de usuário final

Visão geral	413
Adicionar tabelas ReportExplorer ao banco de dados	413
Criar uma nova aplicação	414
Crie componentes de acesso a dados para a tabela de pastas	414
Criar Componentes de Acesso a Dados para a Tabela de Itens	415
Crie os componentes ReportBuilder	415
Configure o Componente Designer	416
Compilar e executar o aplicativo	416
Criar novas pastas	417

Adicionando suporte de dicionário de dados ao aplicativo do usuário final

Visão geral	419
Adicionar tabelas DataDictionary ao banco de dados	419
Abra o aplicativo de usuário final	420
Criar componentes de acesso a dados para as tabelas	420
Criar componentes de acesso a dados para a tabela de campo	420
Criar Componentes de Acesso a Dados para a Tabela de Junção	421
Crie o componente de dicionário de dados	421
Preencher as tabelas	421
Configure o Componente Designer	422
Compilar e executar o aplicativo	422
Crie uma consulta simples	422
Crie um relatório simples	423

Personalizando o formulário do Report Explorer

Visão geral	425
Abra o aplicativo de usuário final	425
Criar um novo formulário do Report Explorer	425

Construindo um aplicativo de relatório usando o InterBase

Visão geral	427
Revise o script SQL Criar tabelas para o Report Explorer	427
Execute o Script SQL Create Tables	429
Criar uma nova aplicação	430
Criar componentes de acesso a dados para a tabela de pastas	430
Criar Componentes de Acesso a Dados para a Tabela de Itens	431
Crie os componentes ReportBuilder	431
Execute o aplicativo	432
Criar novas pastas	433

Para onde eu vou daqui?

The Digital Metaphors Website	435
Learning ReportBuilder	435
Ajuda do ReportBuilder	435
Ajuda RAP	435
Grupos de notícias ReportBuilder	435
Suporte adicional	436

INTRODUÇÃO

O básico 3

Um teste rápido 7

A melhor maneira de aprender ReportBuilder 11

Elementos da interface do usuário 13 Trabalhando

com o Report Designer 15 Noções básicas de

relatórios 19

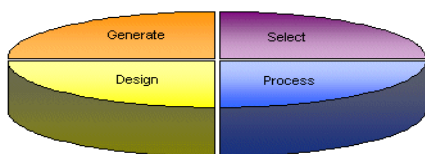
O básico

Bem-vindo ao Guia do desenvolvedor do ReportBuilder. Este guia foi escrito por engenheiros do ReportBuilder para desenvolvedores Delphi. Os conceitos e elementos práticos de relatórios com RB serão revelados ao longo deste livro, e os tutoriais transformarão os conceitos abstratos em relatórios e aplicativos concretos e úteis.

Criação de Relatório

A criação de relatórios com ReportBuilder pode ser dividida em quatro atividades principais:

Report Creation Activities



- **Selecione** - Refere-se à seleção de dados.
- **Projeto** - Refere-se à criação de um layout que descreve a aparência do documento.
- **Processar** - Refere-se à manipulação dos dados ou do layout para controlar com mais precisão a geração do documento.
- **Gerar** - Refere-se à criação do documento real.

O ReportBuilder lida com a etapa Gerar para você e, com a ajuda deste guia, você certamente pode dominar as atividades Selecionar, Projetar e Processar. Primeiro, vamos lidar melhor com esses conceitos bastante abstratos.

Selecione

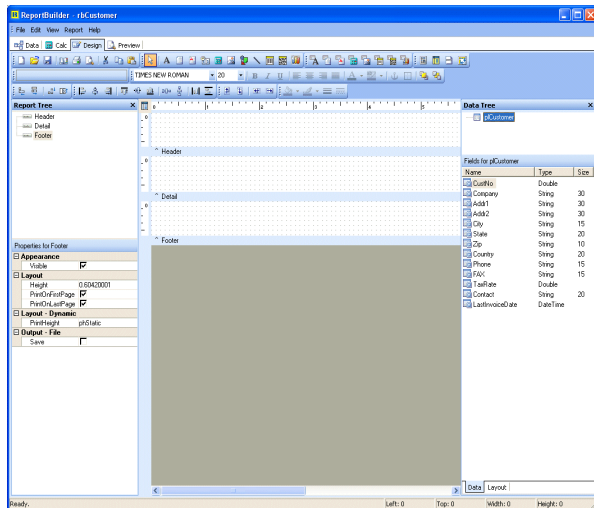
Seus dados provavelmente não estão bloqueados em documentos descartáveis; é organizado (de uma forma ou de outra) como dados. Quando você tem dados, geralmente tem um banco de dados. Os dados organizados são a primeira chave para a criação de documentos recicláveis. O ReportBuilder espera que os dados tenham um formato tabular. Sim, você pode criar relatórios com base em dados menos estruturados, mas em geral é mais vantajoso ter os dados organizados em uma tabela. Aqui está um exemplo:

CustNo	Company	Addr1	Addr2
1221	Kauai Dive Shoppe	4-976 Sugartoa Hwy	Suite 103
1231	Unisco	PO Box 2-547	
1351	Sight Diver	1 Neptune Lane	
1354	Cayman Divers World Unlimited	PO Box 541	
1356	Tom Sawyer Diving Centre	632-1 Third Frydenhoj	
1380	Blue Jack Aqua Center	23-738 Paddington Lane	Suite 310
1384	VIP Divers Club	32 Main St.	

Esta tabela contém informações do cliente. Em termos de banco de dados, cada linha da tabela é considerada um registro. Cada coluna da tabela é considerada um campo. Os nomes dos campos aparecem na primeira linha e não são considerados parte dos dados. Quando você está trabalhando na seleção de dados, o objetivo é criar uma tabela que permitirá que o ReportBuilder gere seu documento corretamente. Portanto, você precisará incluir todos os campos necessários para o relatório, limitar as linhas selecionadas a apenas aquelas que devem aparecer no relatório e classificar as linhas para que apareçam na ordem correta.

Projeto

Assim que os dados forem selecionados, você pode começar a projetar seu relatório. Você faz isso criando um layout. Um layout é uma combinação de objetos que descrevem como o documento deve ser.



Este é o Report Designer. Como você pode ver, ele se parece com muitos dos outros aplicativos do Windows com os quais você está acostumado. A grande diferença é que o Report Designer não contém um documento; ele contém um layout. O layout pode ser usado para gerar muitos documentos diferentes, todos com base nos dados que você selecionou. As áreas retangulares brancas com as barras horizontais abaixo são chamadas de faixas. Este relatório possui uma faixa de cabeçalho, detalhe e rodapé. Quando o ReportBuilder gera um documento a partir deste layout, os objetos na faixa do cabeçalho aparecem no topo de cada página. Os objetos na faixa do rodapé aparecerão na parte inferior de cada página. E os objetos na banda Detalhe se repetirão ao longo da página até que não haja mais espaço de página disponível, quando uma nova página será iniciada. A banda de detalhes é impressa uma vez para cada linha de sua seleção de dados. É assim que um documento é criado a partir do layout. Você pode gerar um documento diferente a partir do mesmo layout, simplesmente alterando sua seleção de dados.

Processar

Ao criar um layout, você informa ao ReportBuilder exatamente como deseja que o documento apareça. Mas e se o seu documento for tão complexo que você não consiga projetar um layout para descrevê-lo? Embora os layouts do ReportBuilder sejam bastante flexíveis e poderosos, como você verá em breve, mesmo o layout mais flexível ainda é fixo. A atividade Processo de criação de documento permite que você forneça instruções adicionais ao ReportBuilder sobre como o documento deve ser criado. Portanto, você pode alterar o layout do relatório à medida que o documento é gerado. Você também pode usar os processos para formatar dados e realizar cálculos. Os processos são criados usando código Delphi ou usando o ambiente de codificação em tempo de execução incluído no ReportBuilder Enterprise. A linguagem nativa do ReportBuilder é chamada RAP (para Report Application Pascal). É fácil de aprender e divertido de usar, especialmente quando você vê os efeitos legais que você pode criar com ele. Por exemplo, você pode usar o RAP para combinar os campos Nome e Sobrenome de um contato para que os dados sejam mais fáceis de ler no relatório ou para calcular uma média ponderada a ser exibida na seção de resumo no final de um relatório.

A maioria dos relatórios não requer um processo, então não se preocupe se você pode ou não dominar essa atividade. Este guia mostrará como fazer cálculos simples e até mesmo algumas coisas bastante complexas, se você estiver interessado. Lembre-se de que o Processo é o lugar para onde você vai quando não consegue que um relatório tenha a aparência que deseja e todas as outras vias parecem estar fechadas.

Geração

Geração é o que acontece quando você clica na guia 'Visualizar' no Report Designer e vê o documento gerado na janela de visualização de impressão. O documento terá a aparência correta ou não. Se não parecer certo, é hora de retornar à área Selecionar, Design ou Processo e fazer as alterações necessárias para que o documento seja gerado corretamente. Às vezes, você aprende mais mexendo em uma seleção de dados ou layout e, em seguida, verificando como o ReportBuilder gera o documento de maneira diferente.

Bem, isso é a criação de um documento ReportBuilder em poucas palavras. Agora vamos começar a criar nosso primeiro relatório!

Um teste rápido

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configurar objetos de acesso a dados Delphi para uso por relatórios
- Trabalhar com o Report Designer
- Crie e configure os componentes de relatório mais comuns

Criar uma nova aplicação

1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.

Crie uma tabela, uma fonte de dados e um componente de pipeline de dados

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblCustomer
Nome da tabela	CUSTOMER.DB

4 Clique duas vezes na propriedade Active no Object Inspector para defini-la como true.

5 Adicione um componente DataSource ao formulário.

6 Configure o componente DataSource:

DataSet tblCustomer
Nome dsCustomer

7 Selecione a aba RBuilder do Delphi compo-paleta atual.

8 Adicionar um componente DBPipeline  para o formulário.

9 Configure o componente DBPipeline:

DataSource dsCustomer

Nome plCustomer

Crie um relatório e conecte-o aos dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

Pipeline de dados plCustomer

Nome rbCustomerList

Chame o Report Designer

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

Coloque um componente de etiqueta na faixa do cabeçalho

1 Clique com o botão direito sobre o espaço em branco da banda do cabeçalho e acesse a caixa de diálogo Posição

2 Defina o valor da Altura como 0,9167 e clique no botão OK. A banda do cabeçalho deve expandir em altura.

NOTE: Você também pode alterar a altura da banda do cabeçalho arrastando o divisor cinza que aparece abaixo do espaço em branco da banda do cabeçalho. Este método é rápido e fácil, mas não tão preciso.

3 Clique no ícone do componente Label  no Paleta do componente Report Designer.

4 Clique no lado esquerdo da faixa do cabeçalho. Um rótulo será criado na banda de cabeçalho e se tornará a seleção atual no Inspetor de Objetos.

5 Localize a caixa Editar no canto superior esquerdo do Report Designer.

6 Selecione o texto dentro dele e substitua por Empresa.

7 Use os controles de fonte para definir a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito itálico

NOTE: Os controles de fonte aparecem à direita da caixa de edição e fornecem as seguintes funções:

- Lista suspensa de nomes de fontes
- Lista suspensa Tamanho da fonte
- Negrito
- itálico
- Estilo de fonte sublinhado
- Alinhamento de Texto
- Cor da fonte
- Cor de destaque

Coloque um componente DBText na faixa de detalhes

1 Clique no ícone do componente DBText  no Paleta do componente Report Designer.

2 Clique na faixa de detalhes. Um componente DBText será criado.

3 Localize as duas listas suspensas no canto superior esquerdo do Report Designer. A lista à esquerda contém uma lista de pipelines de dados. Ele deve usar como padrão o pipeline de dados do cliente, pois é o pipeline de dados do relatório. A lista à direita contém uma lista de campos para o pipeline de dados atualmente selecionado.

4 Selecione o campo Empresa na lista de campos. 'Kauai Dive Shoppe' deve ser exibido no componente DBText.

5 Use os controles de fonte para configurar a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	10
Cor da fonte	Preto

Visualize o relatório em tempo de design

1 Clique na guia Visualizar no Report Designer. Você deve obter um relatório de quatro páginas.

Visualize o relatório em tempo de execução

1 Feche o Report Designer.

2 Selecione a guia Padrão da paleta de componentes Delphi.

3 Adicione um componente Botão ao formulário.

4 Configure o componente Botão:

Nome btnPreview

Antevisão da legenda

5 Coloque o seguinte código no evento OnClick do botão:

```
rbCustomerList.Print;
```

6 Execute o projeto.

7 Clique no botão Visualizar. O relatório deve ser exibido no formulário Visualizar impressão.

A melhor maneira de aprender ReportBuilder?

Comece de forma simples e, em seguida, vá para o próximo nível fácil.

Este é o segredo dos campeões. Se você trabalhar de um nível fácil para o próximo, pode acabar se tornando um desenvolvedor muito habilidoso. Se você pular no topo, pode acabar bastante sobrecarregado. Quando você começa a usar o ReportBuilder, faça o relatório mais fácil que você pode imaginar. Melhor ainda, complete a seção 'Quick Test Spin' deste manual: ela mostrará como construir um relatório simples (e isso levará apenas cerca de 5 minutos). Sei que alguns de vocês vão ler e dizer: "Mas não tenho tempo para isso. Preciso saber se o ReportBuilder fará meus relatórios reais." Minha resposta é que você não tem tempo para não fazer isso. Enquanto gasta horas tentando completar relatórios reais, você aprenderá com a experiência, o que não é um exercício ruim em si, mas esta abordagem é bastante horrível quando comparada com o aprendizado do mestre. Reserve pelo menos 2 a 3 horas para passar pelos tutoriais. Eles foram ordenados do simples ao complexo. Os primeiros tutoriais são os mais fáceis; os últimos são os mais difíceis. Essa abordagem gradiente deve permitir que você desenvolva suas habilidades no ReportBuilder com o mínimo de frustração. Independentemente de você decidir usar este manual ou não, você deve se lembrar deste importante conceito: comece de forma simples e, em seguida, vá para o próximo nível fácil.

Use a ajuda online.

A ajuda pode ser acessada de três maneiras diferentes: acessando a seção Ajuda | Opção de menu de conteúdo do menu principal do Delphi (a referência do ReportBuilder deve estar listada no índice), selecionando um componente ou propriedade e clicando no F1

(o tópico de ajuda apropriado deve ser exibido automaticamente) e pesquisando o índice de ajuda de um tópico. A ajuda contém mais de 1.500 tópicos escritos para o desenvolvedor ReportBuilder. Vale a pena usar com frequência.

Use este manual.

Este manual foi elaborado para dar a você experiência prática com a construção de relatórios e material de base conceitual que deve permitir que você use melhor o ReportBuilder.

Execute os exemplos; Estude os exemplos; Conheça os exemplos.

Os exemplos que vêm com o ReportBuilder são uma parte valiosa do produto. Eles não apenas têm uma boa aparência e mostram muitos dos recursos do produto, mas também mostram como criar diferentes tipos de relatórios. Muitos dos exemplos foram reunidos em resposta direta às perguntas que os desenvolvedores colocaram sobre como usar o ReportBuilder. Por exemplo, um exemplo mostra como compilar e usar um relatório como DLL. Estude os exemplos completamente e use-os como referência quando estiver tentando construir um relatório e não tiver certeza de como juntar todas as peças. Se você estiver tendo problemas para fazer algo funcionar, consulte os exemplos e veja se há um exemplo de relatório semelhante. Os exemplos estão localizados no diretório \RBuilder \Demos.

NOTE: Se você concluiu todos os tutoriais neste manual, poderá separar os exemplos e identificar as principais técnicas em um curto espaço de tempo.

Elementos da interface do usuário

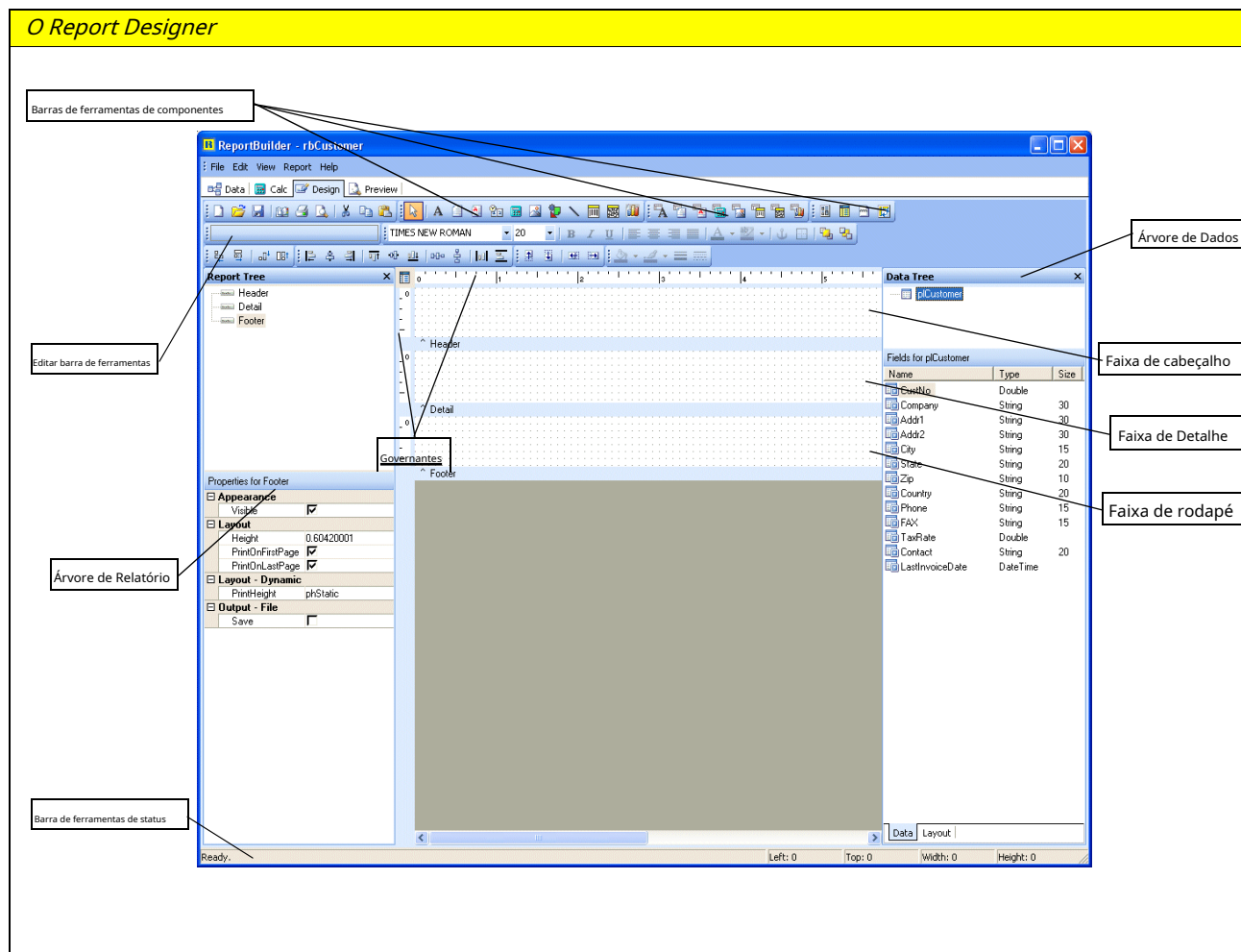
O Report Designer

O Report Designer é a sua chave para a produtividade ao criar relatórios no Delphi. Os engenheiros do ReportBuilder fizeram todos os esforços para garantir que a interface do Report Designer seja consistente com outros programas do Windows que você já usou. As principais áreas do Designer estão listadas abaixo.

Barras de ferramentas da paleta de componentes

Essas barras de ferramentas são usadas para criar novos componentes. Para criar um componente, clique no ícone e depois

clique no espaço em branco de uma banda. Existem três barras de ferramentas de componentes: Padrão, Dados e Avançado. Use os componentes padrão para criar texto, linhas, formas, memorandos, richtext, etc. Use os componentes de dados quando quiser exibir os dados de um banco de dados. Use os componentes avançados quando precisar criar relatórios mais complexos usando sub-relatórios, regiões ou crosstabs.



Editar barra de ferramentas

A barra de ferramentas Editar permite que você defina a propriedade mais importante para um determinado componente. Por exemplo, quando um componente Label é selecionado, aparece uma caixa de edição que permite definir a legenda. Quando um componente DBText é selecionado, duas listas suspensas são exibidas, permitindo que você defina o Data Pipeline e o DataField.

Barra de ferramentas de formatação

Esta barra de ferramentas aparece à direita da barra de ferramentas Editar. É usado para configurar a fonte dos componentes textuais e para controlar a estratificação do componente por meio dos comandos Trazer para a Frente e Enviar para Trás.

Governantes

A régua horizontal permite determinar a posição de um componente na página. A régua vertical para cada banda permite determinar a posição de um componente em relação à posição inicial de impressão da banda.

Todas as bandas

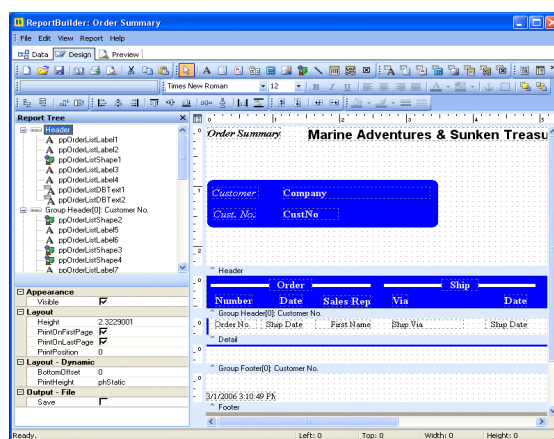
Observe a área retangular cinza abaixo do espaço em branco de cada banda. Esta área é arrastável e permite redefinir a altura da banda.

Barra de status

A barra de status mostra mensagens e posições de objetos.

A árvore do relatório

Você pode exibir a Árvore de Relatórios selecionando Exibir | Barras de ferramentas | Opção de menu Report Tree. Esta janela de ferramenta pode ser encaixada nos lados esquerdo e direito do Report Designer. O painel superior contém uma visualização em árvore da estrutura Report.Bands [], Objects [] que compõe o layout do relatório. Os elementos em cada banda são exibidos em ordem z. O painel inferior da Árvore do Relatório contém o Inspetor de Objetos, que pode ser usado para visualizar e modificar as propriedades do objeto selecionado no momento.

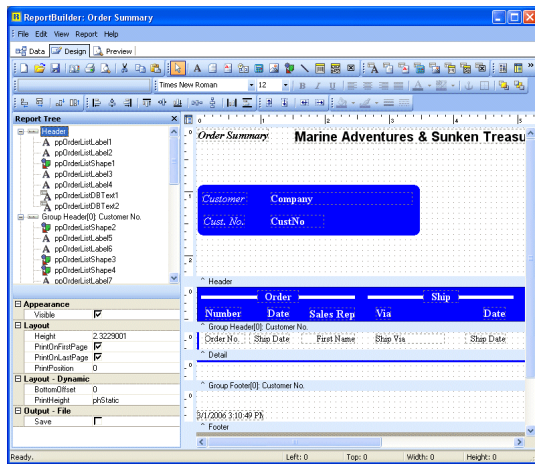


Você pode usar a Árvore de relatórios para ver exatamente quais componentes estão contidos no relatório e para selecionar componentes individuais.

Você pode ativar o Report Outline clicando com o botão direito do mouse na Report Tree. O Report Outline é útil quando você tem sub-relatórios em seu relatório. Você pode selecionar qualquer sub-relatório no Report Outline e as bandas e componentes para aquele sub-relatório aparecerão na Report Tree. (O sub-relatório também será exibido no Report Designer.)

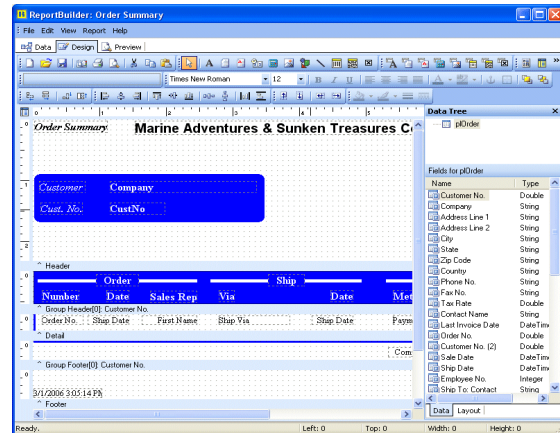
O Inspetor de Objetos

O Object Inspector aparece no painel inferior da janela da ferramenta Report Tree. Use o Inspetor de objetos para ver e modificar as propriedades do objeto atualmente selecionado. Quando vários objetos são selecionados, o Inspetor de Objetos filtra a lista de propriedades para mostrar as propriedades que são comuns a todos os objetos selecionados. A modificação de um valor de propriedade atualizará todos os objetos selecionados.



A Árvore de Dados

Você pode exibir os pipelines de dados que podem ser usados para criar componentes com reconhecimento de dados dentro do relatório, selecionando Exibir | Barras de ferramentas | Opção de menu Árvore de dados. A árvore de dados mostra uma lista de pipelines de dados na janela superior e uma lista de campos para o pipeline de dados atualmente selecionado na janela inferior.



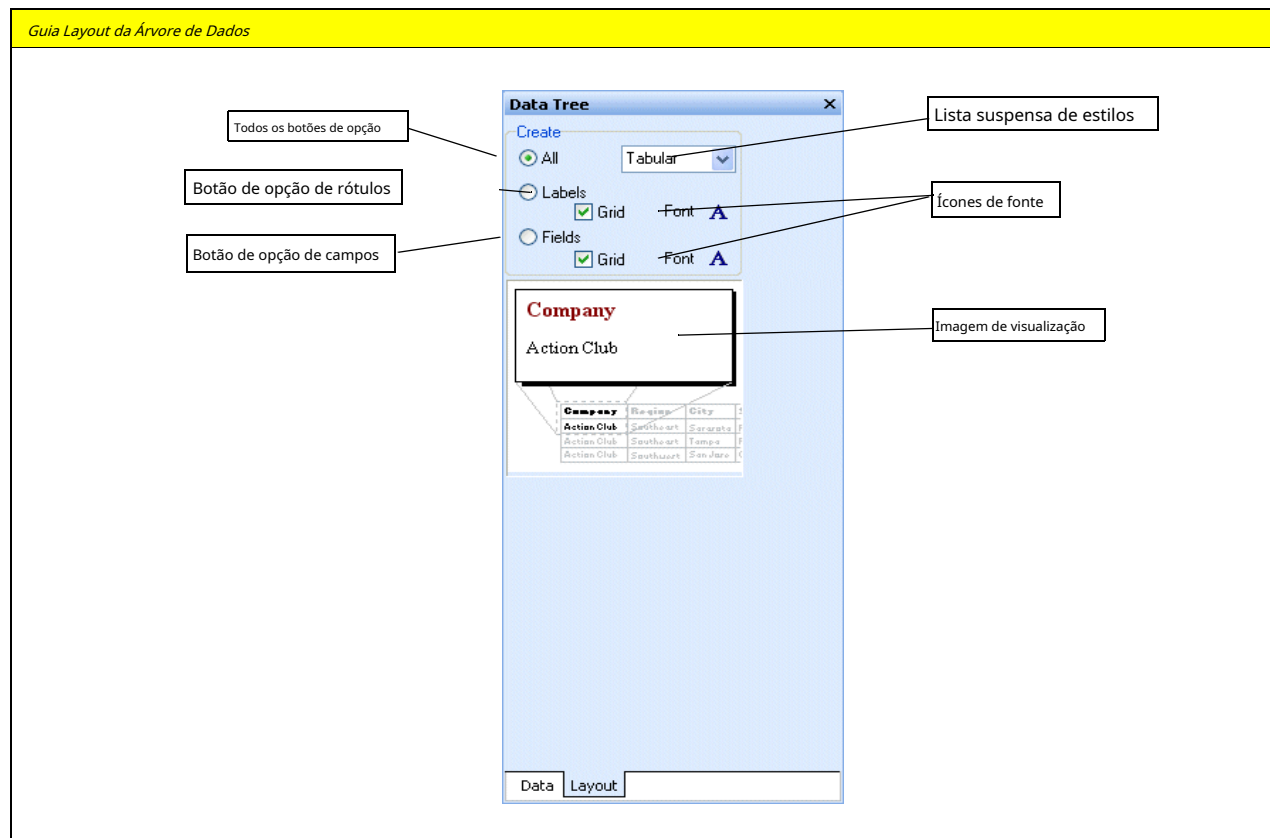
Você pode selecionar vários campos na lista de campos e arrastá-los para qualquer banda. Componentes com reconhecimento de dados e rótulos correspondentes serão então criados. Observe as guias 'Dados' e 'Layout' na parte inferior da Árvore de dados. Você pode usar a guia Layout para personalizar o comportamento dos recursos de arrastar e soltar da Árvore de Dados.

Árvore de dados - cont.

A guia Layout da Árvore de Dados contém muitas configurações que você pode usar para personalizar os recursos de arrastar e soltar da Árvore de Dados.

Lista suspensa de estilos

Controla se os componentes DBText e Label são orientados de forma colunar ou empilhada

**Todos os botões de opção**

Quando selecionado, ambos os componentes DBText (atribuídos aos campos selecionados) e os componentes de rótulo correspondentes são criados.

Botão de opção de campos

Quando selecionado, apenas componentes DBText (atribuídos aos campos selecionados) são criados.

Botão de opção de etiqueta

Quando selecionado, apenas os componentes Label (com as legendas definidas para o nome do campo) são criados.

Caixas de seleção de grade

Controla se uma forma é colocada atrás do componente DBText ou Label.

Ícones de fonte

Controla o nome, tamanho, estilo e cor da fonte por meio de uma caixa de diálogo de fonte padrão.

Imagem de visualização

Mostra como os componentes criados ficarão.

Trabalhando com o Report Designer

Visão geral

O Report Designer é na verdade um editor de componente sofisticado para o componente de relatório (outros exemplos de editores de componente no Delphi são o Editor de Menu do componente TMenu e o Editor de Campos dos componentes TTable e TQuery). O Report Designer permite que você faça o layout de relatórios complexos de maneira rápida e fácil, da mesma forma que o Delphi Form Designer.

No Delphi, quando você coloca um componente em um formulário, três coisas ocorrem:

- 1 O componente é desenhado em seu formulário.
- 2 A declaração do componente é adicionada à sua unidade de formulário (arquivo .pas).
- 3 As propriedades e eventos de tempo de design do componente tornam-se visíveis no Inspetor de objetos.

No ReportBuilder, quando você coloca um componente ReportBuilder de uma das paletas de componentes do Report Designer em uma banda de relatório, três coisas acontecem:

- 1 O componente é desenhado no layout do seu relatório.
- 2 O componente é adicionado à sua unidade de formulário (arquivo .pas).
- 3 As propriedades e eventos de tempo de design do componente tornam-se visíveis no Inspetor de objetos.

Em outras palavras, os componentes do relatório são como quaisquer outros componentes do Delphi: você pode configurar suas propriedades e atribuir manipuladores de eventos a eles usando o Object Inspector. Como alternativa, você pode usar o Report Designer.

Algumas dicas para ajudá-lo a obter o máximo do ReportBuilder:

- Maximize a janela Report Designer e use os menus rápidos tanto quanto possível.
- Para definir formatos de exibição, ative seu conjunto de dados e use os menus rápidos para acessar a caixa de diálogo Formato. A caixa de diálogo Formato determinará o tipo de dados do DataField atribuído ao componente e exibirá vários formatos apropriados para esse tipo de dados. (Por exemplo, um campo do tipo data forneceria uma lista de formatos de data comumente usados.)
- Para redimensionar as bandas usando o mouse, posicione o mouse sobre a área retangular cinza abaixo do espaço em branco da banda, pressione o botão esquerdo do mouse e arraste.
- Para acessar a ajuda online de um componente, selecione o componente e pressione F1.
- Para fazer uma determinada banda aparecer como o objeto selecionado no Object Inspector, posicione o cursor do mouse em qualquer lugar sobre o espaço em branco aberto da banda (não sobre um componente na banda) e clique com o botão esquerdo do mouse.
- Para fazer o relatório aparecer como o objeto selecionado no Inspetor de Objetos, posicione o mouse no canto superior esquerdo da área de trabalho do Report Designer (à esquerda da régua horizontal e acima da régua vertical) e clique em Selecionar Relatório ícone .
- Você pode recortar, copiar e colar um ou mais componentes do relatório por vez no mesmo relatório ou entre relatórios diferentes.

- Evite compartilhar uma fonte de dados para um relatório ReportBuilder com um controle ciente de dados em seu formulário Delphi (por exemplo, um DBGrid). Isso é ineficiente para relatórios porque quando ReportBuilder atravessa o conjunto de dados para gerar um relatório, seus controles com reconhecimento de dados serão notificados repetidamente de que o registro atual foi alterado. Se houver eventos anexados aos controles com reconhecimento de dados, os eventos também serão disparados repetidamente. Isso diminui muito o desempenho e pode causar resultados inesperados.
- Use a propriedade PageLimit do relatório para limitar o número de páginas visualizadas ao trabalhar em relatórios longos. Em outras palavras, se o relatório tiver 200 páginas, defina o PageLimit como 20 e apenas as primeiras páginas serão visualizadas.

Feche a janela Report Designer antes de fechar seu formulário ou sair do Delphi. Caso contrário, pode ocorrer uma violação de acesso.

Noções básicas de relatórios

Consultar tabelas / consultas

Um componente de relatório com reconhecimento de dados pode ser definido para qualquer pipeline de dados em seu formulário; portanto, se uma tabela de pesquisa ou consulta estiver conectada aos seus dados mestre ou de detalhes, você pode atribuir campos dos dados de pesquisa simplesmente criando um pipeline de dados para a pesquisa e, em seguida, atribuindo esse pipeline de dados a um componente.

Filtrando Dados

Sempre que você define um filtro nos dados que estão conectados a um relatório, você precisa chamar o método `Report.Reset` antes de chamar `Report.Print`. Isso notificará o `ReportBuilder` de que ele precisa acessar novamente os dados e gerar novamente as páginas do relatório, em vez de usar o cache do mecanismo interno. Chamando `Report.Reset` é uma boa técnica sempre que parece que um relatório não está sendo regenerado em resposta a alterações nos dados.

Executando Cálculos

Existem três maneiras de realizar cálculos para relatórios: campos calculados Delphi, componentes `TppDBCalc` e componentes `TppVariable`.

1 Use os campos calculados do Delphi quando quiser calcular um resultado para cada registro no conjunto de dados com base nos valores de um ou mais campos do registro. Você pode criar objetos `TField` clicando duas vezes no componente do conjunto de dados (`TTable`, `TQuery` ...) e acessando o Editor de Campos. Use o Editor de campos para criar objetos `TField` calculados. Em seguida, no evento `OnCalcFields` do conjunto de dados, adicione o código para calcular o resultado e atribua-o ao componente `TField`.

2 Use os componentes `TppDBCalc` quando precisar calcular `SUM`, `COUNT`, `MIN`, `MAX` ou `AVG` para um grupo ou um relatório inteiro. O `COUNT` `DBCalcType` também pode ser usado no `DetailBand` para exibir o número da linha para cada registro no relatório.

3 Use o componente TppVariable para realizar cálculos. Defina a propriedade DataType para o tipo de dados desejado. Clique com o botão direito sobre a variável e acesse a opção Tempo Defina o tempo conforme apropriado. Adicione código ao evento OnCalc para realizar cálculos. Se os cálculos forem baseados nos valores de outras variáveis, pode ser necessário definir a ordem de cálculo das variáveis. Para definir a ordem de cálculo, acesse a caixa de diálogo Ordem de cálculo clicando com o botão direito do mouse no espaço em branco da banda e selecionando a opção de menu Ordem de cálculo Você pode então ordenar as variáveis para cálculo. Retorne o valor do cálculo no parâmetro Value do evento OnCalc ou atribua o resultado a uma das seguintes propriedades: AsInteger, AsBoolean, AsString, AsFloat, AsDateTime, AsDate ou AsTime. O valor da propriedade Text refletirá o valor calculado.

NOTE: Lembre-se sempre de que os manipuladores de eventos Object Pascal não são executados em tempo de design; você deve compilar e executar seu projeto para ver os resultados.

Formatos de exibição

Você pode especificar os formatos de um componente DBText, DBCalc, Variable ou SystemVariable definindo a propriedade DisplayFormat. DisplayFormat difere da implementação Delphi no caso de tipos de string. Para formatar strings, simplesmente digite um EditMask válido na propriedade DisplayFormat. ReportBuilder aplicará o EditMask ao valor da string.

NOTE: ReportBuilder ignora quaisquer formatos de exibição que você especificar nos objetos TField de um conjunto de dados Delphi.

Bandas Dinâmicas

Defina a propriedade Band.PrintHeight como phDynamic quando desejar que a banda use o espaço da página conforme a necessidade, diminuindo ou ampliando para acomodar os componentes do relatório. Quando PrintHeight é definido como phStatic, a banda usa a quantidade exata de espaço de página especificada pela propriedade Height (a menos que não seja Visível, caso em que usa espaço de página zero).

Ampliando memorandos e formas

Defina a propriedade Stretch de um memorando quando desejar que a altura do memorando seja esticada automaticamente para permitir que todo o conteúdo do memorando seja impresso. Se você estiver enquadrando o memorando com uma Forma, defina a propriedade Shape.StretchWithParent como True e a Altura da forma será esticada para acomodar a altura do memorando. Por fim, use a propriedade ShiftWithParent dos outros componentes do relatório para determinar se a posição do componente do relatório deve se mover conforme o memorando se expande.

Controlando a visibilidade do componente

Você pode usar o evento BeforePrint de uma banda para controlar quais componentes aparecem quando a banda é impressa. Para ocultar todos os componentes em uma banda, defina a propriedade Visible da banda como False. Para ocultar um grupo de componentes, coloque os componentes dentro de uma região e defina a propriedade Visible da região como falsa. Para ocultar componentes individuais, defina o visível de cada componente como False.

REPORTBUILDER

FSEM DANOS

25 principais

Dados 49

Código 65

Design 89

Imprimir 115

Implantar 135

MAIN

Introdução	25
Os Componentes Delphi	27
Componentes do relatório	29
Layouts inteligentes	33
SubReports	39
Emulação de Formulário	43

MAIN

Introdução

Visão geral

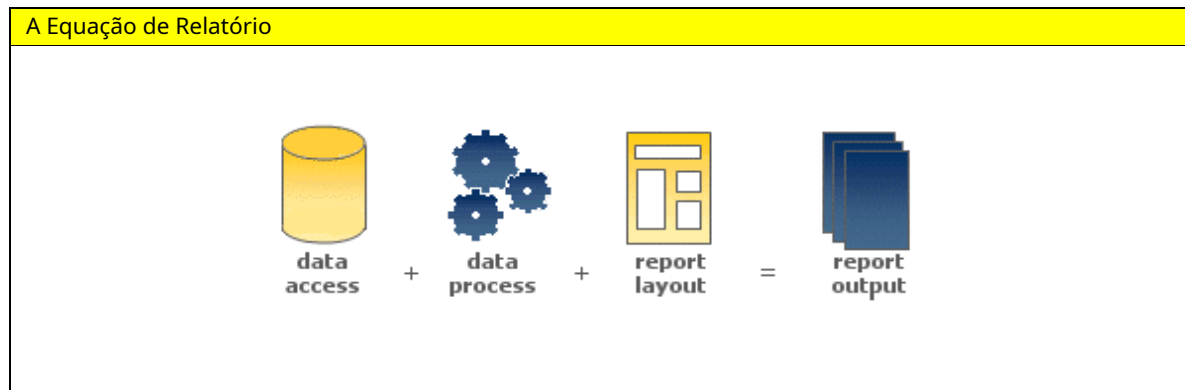
ReportBuilder é um ambiente de desenvolvimento que pode ser usado para construir relatórios, componentes de relatórios e aplicativos de relatórios. Porque os relatórios abrangem uma ampla gama de requisitos, muitas vezes é difícil limitar o que se deve esperar de uma ferramenta de relatório. Os designers do ReportBuilder reduziram esse amplo conjunto de requisitos à seguinte equação:

Layout de relatório

O layout do relatório é um conjunto de componentes que descreve a aparência do relatório e define o comportamento dos componentes durante a geração do relatório.

Saída de relatório

A saída do relatório é um conjunto de componentes que descrevem o conteúdo exato de cada página.



Acesso de dados

O acesso a dados é a recuperação de dados de uma tabela de banco de dados, arquivo de texto, objeto Delphi ou outro conjunto de dados de forma organizada (estruturado como registros e campos).

Processo de Dados

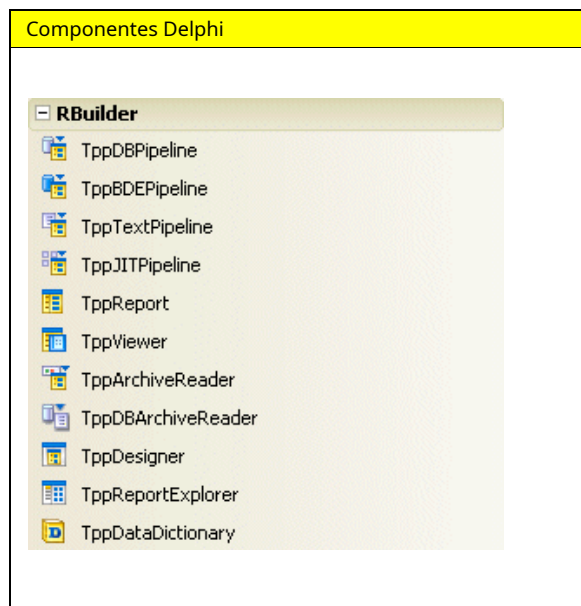
O processo de dados se refere ao cálculo de resultados intermediários com base nos dados e à modificação do layout do relatório à medida que ele é gerado.

Essa equação cobre mais ou menos toda a área de desenvolvimento conhecida como relatório. Em termos de importância, cada elemento da equação não é suportado ou tem suporte fraco, então a utilidade da solução de relatório é bastante reduzida.

As primeiras versões do ReportBuilder focavam no layout do relatório e nos elementos de saída do relatório, em grande parte devido ao fato de que Delphi possui soluções abundantes para o acesso a dados (ADO, BDE, dbExpress, IBExpress, componentes de acesso a dados de terceiros, String Lists, etc.) e para os elementos de processo de dados (eventos e Object Pascal). Na verdade, a utilização dessas soluções ainda é uma forma válida e produtiva para os desenvolvedores usarem o ReportBuilder. O desenvolvedor pode configurar componentes de acesso a dados Delphi padrão, conectá-los a um relatório por meio do componente de pipeline de dados, projetar e visualizar o relatório em tempo de design Delphi e, se necessário, codificar manipuladores de eventos em Object Pascal para realizar cálculos ou modificar o layout do relatório durante a geração.

Os desenvolvedores do ReportBuilder descobriram que soluções para o acesso aos dados e elementos do processo de dados também eram necessárias para os usuários finais que utilizavam o ReportBuilder Report Designer como parte de um aplicativo em execução. No contexto de um aplicativo em execução, o desenvolvedor Delphi não tem como fornecer acesso aos componentes de acesso a dados ou ao Object Pascal como ambientes de desenvolvimento. Os usuários finais não podiam, portanto, aproveitar essas ferramentas poderosas. Essa situação acabou levando ao desenvolvimento do DADE (o Data Access Development Environment) e do RAP (a linguagem de programação Report Application Pascal), que deram aos usuários finais soluções completas para acesso e processamento de dados, respectivamente. O DADE foi disponibilizado como parte da Professional Edition. Tanto o DADE quanto o RAP estão disponíveis como parte da Enterprise Edition.

Os Componentes Delphi



DBPipeline

Usado para acessar dados por meio de descendentes de TDataSet. O DBPipeline é conectado por meio da propriedade DataSource.



BDEPipeline

Nas versões anteriores do ReportBuilder, o BDEPipeline era usado para acessar dados por meio do BDE. Embora tenha sido substituído pelo DBPipeline, ele foi mantido para capacidade de retrocesso.



TextPipeline

Usado para acessar vírgula, guia e arquivos de texto de registro de comprimento fixo. Defina a propriedade FileName para especificar o arquivo. Clique duas vezes no componente para definir a estrutura do campo.



JITPipeline

Usado para acessar quaisquer dados não estruturados armazenados em objetos Delphi ou outras fontes. Fornece controle total sobre o processo de acesso aos dados. Defina as propriedades InitialIndex e RecordCount e codifique o evento OnGetFieldValue para utilizar este componente. Clique duas vezes no componente para definir a estrutura do campo.



Relatório

O principal componente. Clique duas vezes para invocar o Report Designer. Atribua a propriedade DataPipeline para que o relatório possa percorrer os dados. Atribua a propriedade DeviceType para controlar para onde a saída do relatório é direcionada. Chame Report.Print de Object Pascal para imprimir o relatório ou inicie o formulário de visualização de impressão.



Visualizador

Este objeto raramente é usado porque você pode substituir o formulário de visualização de impressão integrado do ReportBuilder por sua própria versão personalizada com muita facilidade (verifique o tutorial Construindo um Aplicativo de Relatório). Se você deve usar este componente, um exemplo é fornecido em \RBuilder\Demos\Reports.



Leitor de arquivo

Depois de imprimir um relatório em um arquivo compactado (extensão .raf), você pode ler e visualizar o arquivo por meio deste componente. Basta atribuir o ArchiveFileName ao arquivo e chamar o método Print. Em termos de exibição de um relatório, este componente funciona da mesma forma que o componente Relatório.

Componentes de relatório

Visão geral

A ReportBuilder Report Component Library (RCL) fornece um conjunto poderoso e robusto de componentes que foram projetados e otimizados especificamente para o ambiente de relatórios. A biblioteca inclui mais de 20 componentes que permitem colocar todos os tipos de dados em seus relatórios: Linhas, Formas, Texto, Memorandos, RichText, Imagens, Gráficos e Códigos de Barras. Componentes avançados, como Regions, SubReports e CrossTabs, podem ser usados para modelar relatórios complexos com elegância.



Memorando

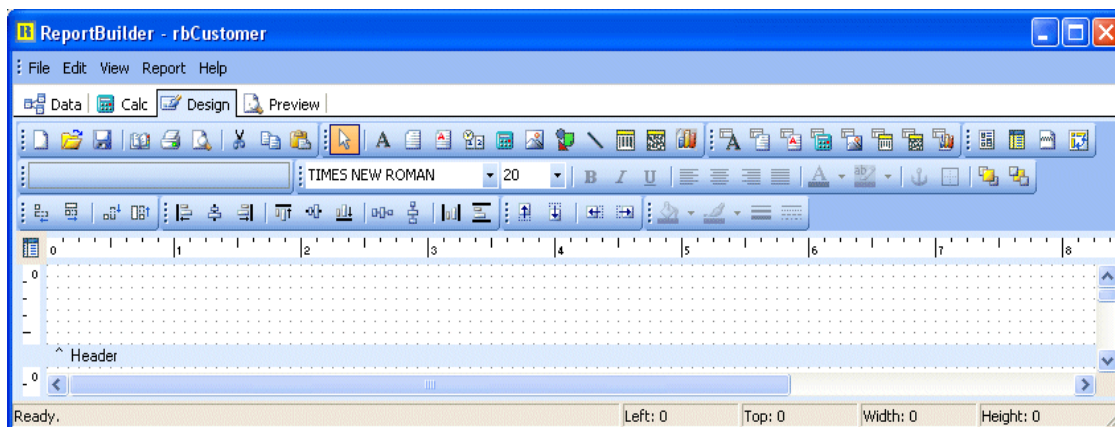
Usado para imprimir várias linhas de texto simples em um relatório. Para definir o valor, atribua uma lista de strings à propriedade Lines. Para redimensionar dinamicamente o memorando durante a impressão, defina a propriedade Stretch como True. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis.



RichText

Usado para imprimir texto formatado. Para definir o valor, atribua a propriedade RichText ou use o

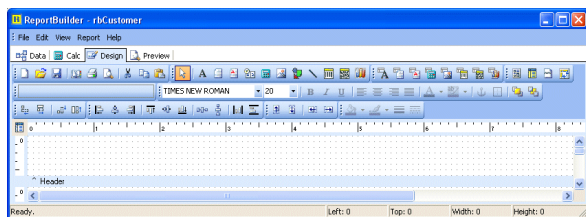
Delphi Report Components



Rótulo

Usado para exibir texto. Atribua a propriedade Caption para controlar o valor do texto. Para redimensionar o rótulo automaticamente para que se ajuste a uma legenda variável, defina a propriedade AutoSize como True.

Métodos LoadFromFile ou LoadFromRTFStream. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis. No momento do design, você pode usar o Editor RTF integrado do ReportBuilder para carregar, modificar e salvar dados de rich text armazenados em arquivos.

Componentes do relatório - cont. **SystemVariable**

Usado para exibir informações de relatório comuns, como número de página, contagem de página, data e hora de impressão, data, hora, etc. O tipo de informação exibida é controlado pela propriedade VarType. O formato é controlado pela propriedade DisplayFormat.

 Variável

Usado para cálculos por meio de um manipulador de eventos Object Pascal atribuído ao evento OnCalc ou um manipulador de eventos RAP atribuído ao evento OnCalc. Acesse a caixa de diálogo Cálculos (através do menu rápido) ou a guia Calc do Report Designer para codificar um cálculo RAP para este componente.

 Imagem

Usado para exibir bitmaps e meta-arquivos do Windows em relatórios. Atribua a propriedade Picture deste componente para colocar uma imagem em seu relatório. Use a caixa de diálogo de imagem integrada do Report Designer para carregar imagens em tempo de design.

 Forma

Use este componente para imprimir várias formas (quadrados, retângulos, círculos, elipses). Defina a propriedade Forma para selecionar um tipo de forma. Use as propriedades Pincel e Caneta para controlar a cor e a borda, respectivamente.

 TeeChart

Usado para exibir TeeCharts padrão (sem reconhecimento de dados). Este componente permite que você use o TeeChart dentro do Report Designer. Você pode acessar o editor TeeChart por meio de um menu pop-up.

 BarCode

Usado para renderizar códigos de barras. O valor da string atribuído à propriedade Data é codificado com base no BarCodeType. Se os dados a serem codificados estiverem em um banco de dados, use DBBarCode. As seguintes simbologias são suportadas: Codabar, Código 128, Código 39, EAN-13, EAN-8, FIM A, B, C, intercalado 2 de 5, PostNet, UPC-A, UPC-E.

 Código de Barras 2D

Usado para renderizar simbologias de código de barras bidimensionais. Suporta os tipos de código de barras PDF417 e MaxiCode

 CheckBox

Exibe uma caixa de seleção usando a fonte WingDings.

 DBText

Usado para exibir valores de todos os tipos de campos do banco de dados. Use a propriedade DisplayFormat para formatar o valor.

 DBMemo

Usado para imprimir texto simples de um campo de memorando de uma tabela de banco de dados. Este controle irá quebrar automaticamente o texto. Defina a propriedade Stretch como True e o componente será redimensionado dinamicamente para imprimir todo o texto. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis.

Componentes do relatório - cont.



DBRichText

Usado para imprimir texto formatado de um memorando ou campo BLOB de uma tabela de banco de dados. Este controle irá quebrar automaticamente o texto. Defina a propriedade Stretch como True e o componente será redimensionado dinamicamente para imprimir todo o texto. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis.



DBCalc

Usado para cálculos de banco de dados simples (soma, mínimo, máximo, contagem e média). O valor pode ser redefinido quando um grupo é interrompido usando a propriedade ResetGroup.



DBImage

Usado para imprimir bitmaps ou metarquivos do Windows, que são armazenados em um campo BLOB do banco de dados.



DBBarcode

Usado para renderizar códigos de barras com base no BarcodeType e no valor fornecido por meio da propriedade DataField. As seguintes simbologias são suportadas: Codabar, Código 128, Código 39, EAN-13, EAN-8, FIM A, B, C, Intercalado 2 de 5, PostNet, UPC-A, UPC-E.



DBBarcode 2D

Usado para renderizar código de barras bidimensional com base no Barcode Type e no valor fornecido por meio da propriedade DataField. As seguintes simbologias são suportadas: PDF417, MaxiCode.



DBTeeChart

Permite que TeeCharts com reconhecimento de dados sejam colocados em um relatório.



DBCheckBox

Exibe uma caixa de seleção com base no valor do campo especificado na propriedade DataField. Pode ser usado com um campo booleano (ou qualquer outro tipo de campo por meio das propriedades BooleanTrue, BooleanFalse).



Região

Usado para agrupar componentes logicamente. Use a propriedade ShiftRelativeTo para mover a região em relação a outro componente de redimensionamento dinâmico (como Memo, RichText ou SubReport do tipo filho).



SubRelatório

Usado para lidar com vários detalhes principais, criar efeitos de relatório lado a lado e conectar relatórios como um só. Se você precisar que um relatório seja impresso dentro do contexto de uma banda, use um sub-relatório do tipo filho. Se você precisar conectar relatórios, use um sub-relatório de tipo de seção. A propriedade PrintBehavior determina o tipo de sub-relatório.



CrossTab

Usado para apresentar dados resumidos em formato de grade.



Quebra de página

O componente TppPageBreak é um controle de relatório que permite ao usuário forçar uma nova página durante a geração do relatório. Colocar um componente TppPageBreak em um relatório fará com que todos os objetos criados após o PageBreak (Z-Order) sejam movidos para a próxima página no relatório em relação à posição superior do objeto PageBreak.

Layouts inteligentes

Visão geral

ReportBuilder permite que você crie layouts de relatório altamente dinâmicos. Os componentes SubReport, Memo, RichText e Region têm a capacidade de expandir ou contrair para acomodar as informações que contêm. Há uma série de propriedades projetadas para manter seus relatórios com boa aparência na variedade de situações criadas por esses componentes dinâmicos.

Âncoras

Use Âncoras para garantir que um componente de relatório mantenha sua posição atual em relação a uma borda de seu controle pai (ou seja, Banda / Região). Quando o pai é redimensionado, o componente manterá sua posição em relação às arestas às quais está ancorado.

StretchWithParent

Permite que uma forma ou linha se expanda ou contraia com base na mudança na altura da faixa ou região na qual está contida.

ShiftRelativeTo

Usado para especificar o posicionamento vertical que deve ocorrer entre vários componentes de alongamento em uma banda.

StopPosition (para sub-relatórios)

Usado para definir a posição na página onde um sub-relatório do tipo filho interromperá a impressão. Permite que um sub-relatório do tipo filho seja confinado a uma área retangular da página.

BottomOffset

Usado para criar espaço em branco entre vários objetos de alongamento que foram vinculados usando a propriedade ShiftRelativeTo.

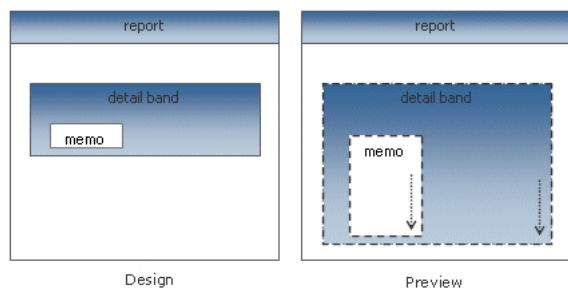
OverflowOffset

Controla a posição em que um componente de alongamento começará a ser impresso ao transbordar para páginas adicionais. Esta propriedade pode ser usada para imprimir um objeto em uma posição inicial diferente quando ele transborda para páginas adicionais.

ReprintOnOverflow

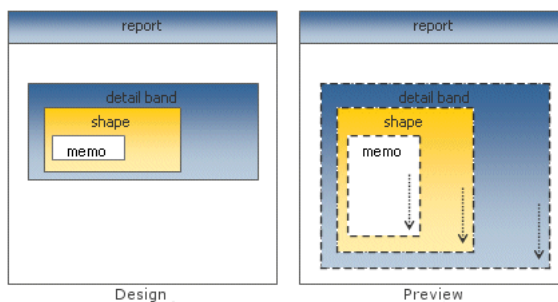
Usado para imprimir componentes que não são de alongamento quando os componentes de alongamento são impressos em páginas adicionais.

Um memorando na faixa de detalhes



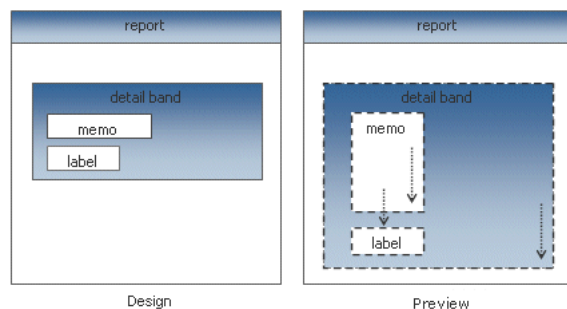
Aqui temos um único componente de memorando na faixa de detalhes. A propriedade `Stretch` do memorando foi definida como `True`. Cada vez que a banda de detalhes é impressa, a altura do memorando é recalculada com base na quantidade de texto que ele contém. Como resultado, o memorando pode aumentar ou diminuir de tamanho e a faixa de detalhes aumentará e diminuirá com ele. O memorando pode conter tanto texto que não cabe em uma única página. Nesse caso, o memorando será impresso em páginas adicionais até que seja concluído. No ReportBuilder, essa condição é conhecida como estouro. Os componentes `memo` e `banda` têm uma propriedade booleana `Overflow` que pode ser verificada enquanto o relatório está sendo gerado para determinar se o `memo` está sendo impresso em uma página adicional.

Um memorando com uma forma de fundo



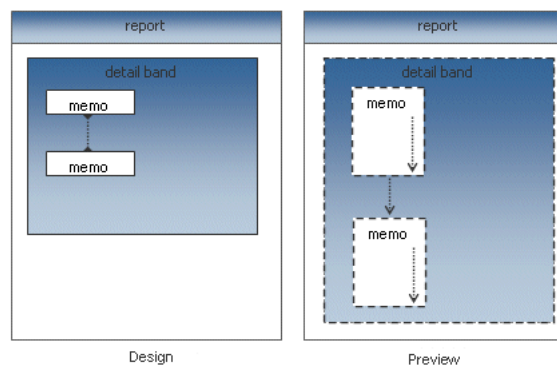
Aqui temos um único memorando na faixa de detalhes com uma forma atrás dela. A forma tem a propriedade `StretchWithParent` definida como `True`. O pai, neste caso, é a banda de detalhes. Quando a banda for gerada, o memorando será alongado com base no texto que contém; a banda será redimensionada para acomodar o memorando e a forma será redimensionada com base na alteração da altura da banda. Esse alongamento e redimensionamento cria o efeito de uma borda e um plano de fundo para o memorando. Se o memorando transbordar para páginas adicionais, também podemos instruir a impressão da forma definindo a propriedade `ReprintOnOverflow` da forma como `True`.

Um memorando com etiqueta abaixo



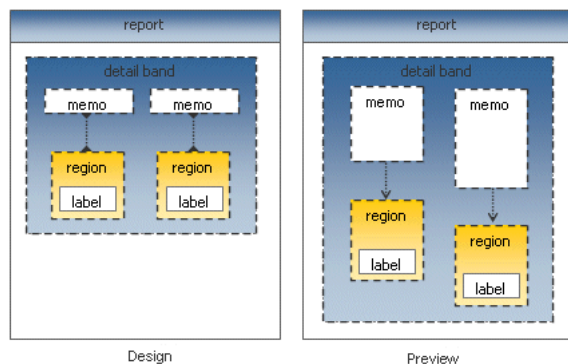
Aqui temos um único memorando na banda de detalhes com um rótulo abaixo dele. O rótulo possui a propriedade `ShiftWithParent` definida como `True`. O pai, neste caso, é a banda de detalhes. Quando a banda é gerada, o memorando se estica e a altura da banda aumenta ou diminui de acordo. O rótulo mudará com base na mudança na altura da banda.

Dois memorandos empilhados na faixa de detalhes



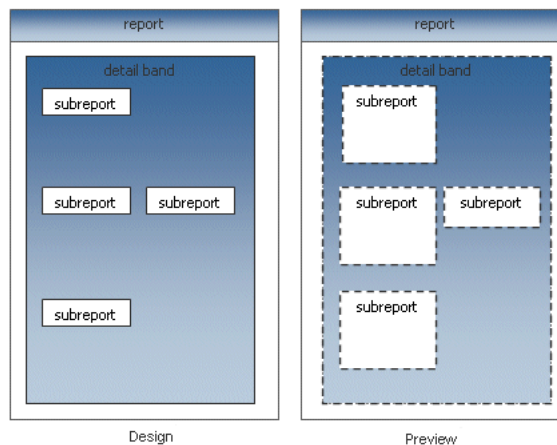
Este relatório requer a impressão de dois memorandos, um após o outro. Esse requisito é atendido definindo-se a propriedade `ShiftRelativeTo` do segundo memorando de forma que aponte para o primeiro. Com esta configuração, o primeiro memorando será impresso, estendendo-se para acomodar o texto que contém, e então o segundo memorando será impresso até o fim.

Dois memorandos lado a lado com rótulos abaixo



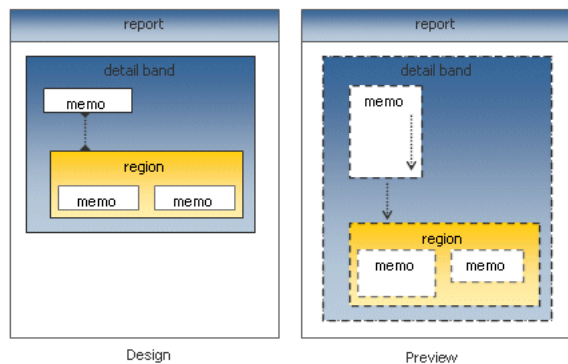
Aqui temos dois memorandos na faixa de detalhes, cada um configurado para se esticar. Quando o relatório for gerado, a banda aumentará ou diminuirá para acomodar o memorando que contém a maior parte do texto. Cada memorando possui um rótulo associado abaixo dele. Queremos que os rótulos mudem em relação ao memorando acima. Para fazer isso, colocamos os rótulos em uma região e, em seguida, definimos a propriedade `ShiftRelativeTo` da região para apontar para o memorando acima. Agora, quando o relatório é gerado, cada rótulo muda em relação ao memorando associado.

Sub-relatórios filhos em posições fixas



Este relatório emula um formulário que contém informações em áreas retangulares fixas da página. Podemos fazer com que o relatório 'preencha' este formulário colocando sub-relatórios do tipo `child` no início de cada área retangular e definindo a propriedade `StopPosition` igual à parte inferior da área retangular.

Um memorando com dois memorandos lado a lado abaixo



Neste relatório, temos um único memorando extenso que precisa ser impresso até a conclusão e, em seguida, dois memorandos adicionais precisam ser impressos, começando imediatamente após o primeiro. Podemos obter esse efeito colocando os memorandos adicionais em uma região e definindo a propriedade `ShiftRelativeTo` da região para o primeiro memorando.

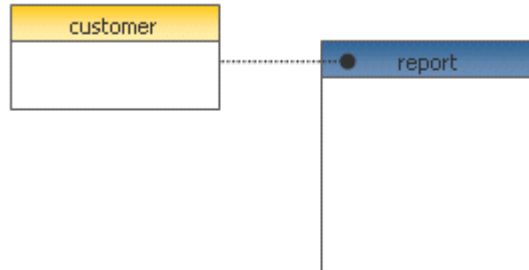
SubReports

Visão geral

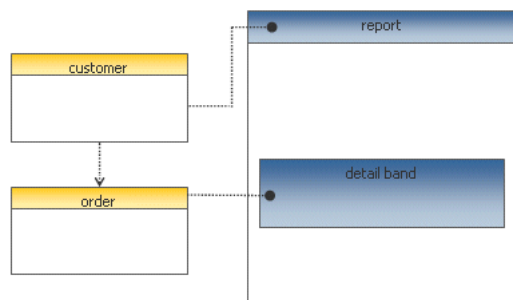
Em redatores de relatórios tradicionais em faixas, os relatórios que podem ser impressos a partir de uma única fonte de dados são muito fáceis de criar. Mas se o conteúdo do relatório consistir em informações de várias fontes de dados diferentes, as opções tornam-se bastante limitadas. Uma opção é usar o SQL para juntar os dados em uma tabela virtual e, em seguida, construir o relatório com base nessa tabela. Se muitas tabelas estiverem envolvidas, o desempenho dessa abordagem pode ser proibitivo.

O Delphi oferece uma alternativa a essa abordagem, permitindo que ligações sejam estabelecidas entre os objetos de acesso a dados. No ReportBuilder, podemos usar sub-relatórios de formato livre para tirar proveito das muitas configurações que esses objetos de acesso a dados tornam possíveis.

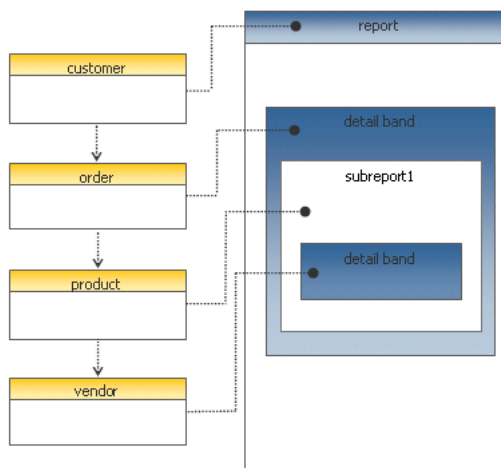
Conjunto de dados único



Um único conjunto de dados pode ser conectado diretamente ao relatório por meio da propriedade DataPipeline. Quando impresso, o relatório irá gerar uma banda de detalhes para cada registro fornecido pelo conjunto de dados.

Conjunto de dados mestre com conjunto de dados de detalhe único

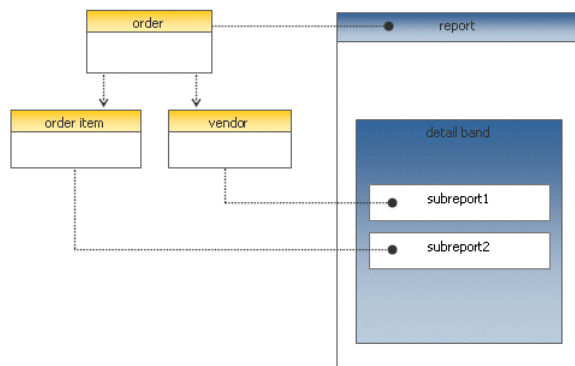
Nesse cenário, os dados mestre são conectados aos dados detalhados por meio de um campo ou conjunto de campos. Presume-se que essa conexão resulte na seleção de vários registros de detalhes para cada registro mestre individual. Os dados mestre são atribuídos ao relatório e os dados detalhados são atribuídos a um sub-relatório. Quando o relatório é gerado, o relatório principal percorre todos os registros do cliente e o sub-relatório percorre todos os pedidos de cada cliente.

Conjunto de dados mestre com conjuntos de dados de detalhes aninhados

Aqui temos o conjunto de dados mestre contendo uma lista de clientes. Cada cliente tem vários pedidos; cada pedido tem vários produtos; e cada produto tem vários fornecedores potenciais. Essa configuração é chamada de 'aninhada' porque cada conjunto de registros é selecionado com base na ligação estabelecida com o conjunto de dados anterior.

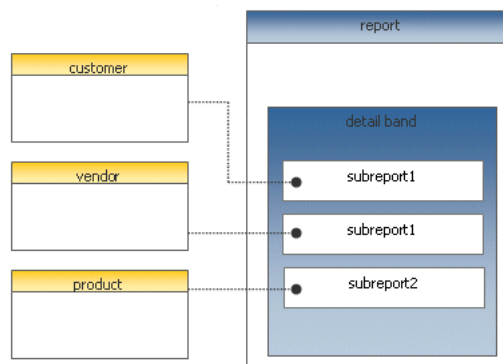
Podemos atravessar essa configuração de dados aninhando sub-relatórios na banda de detalhes. O conjunto de dados do cliente é atribuído ao relatório principal. O conjunto de dados do pedido é atribuído a um sub-relatório na faixa de detalhes do relatório principal. O conjunto de dados do produto é atribuído a um sub-relatório na faixa de detalhes do sub-relatório do produto e assim por diante. Dessa forma, o relatório é construído para corresponder aos dados e cada conjunto de dados tem um layout completo que pode ser usado para renderizar seu conteúdo.

Conjunto de dados mestre com vários conjuntos de dados independentes



Aqui temos a tabela de pedidos. Cada pedido tem muitos itens de pedido. Para cada pedido, há também um grupo de fornecedores que podem fornecer os produtos para esse pedido. Ambos os conjuntos de dados estão vinculados ao conjunto de dados mestre, em vez de serem aninhados um no outro; portanto, os conjuntos de dados são 'independentes'. Este tipo de dados pode ser tratado colocando dois sub-relatórios na faixa de detalhes. O primeiro sub-relatório pode ser vinculado ao conjunto de dados do item do pedido e o segundo sub-relatório pode ser vinculado ao conjunto de dados do fornecedor. Para imprimir os dados do fornecedor após os dados do item do pedido, precisamos vincular o sub-relatório do fornecedor ao sub-relatório do item do pedido por meio da propriedade `ShiftRelativeTo`.

Conjuntos de dados independentes



Aqui temos três fontes de dados sem ligação entre elas. Neste relatório, queremos imprimir todos os clientes, depois todos os produtos e, por fim, todos os fornecedores. O relatório precisa se encaixar como um livro, com cada conjunto de dados fornecendo um capítulo. Aqui, usamos o relatório principal para lançar um sub-relatório para cada um dos conjuntos de dados. Os sub-relatórios seriam definidos com um `PrintBehavior` da seção, o que significa que cada sub-relatório iniciaria uma nova página, geraria um conjunto de páginas conforme necessário para percorrer todos os dados e, em seguida, retornaria o controle ao relatório principal. O relatório principal não está conectado a nenhum conjunto de dados e, portanto, imprimirá apenas uma única banda de detalhes.

Emulação de Formulário

Visão geral

A emulação de formulário é o processo de pegar um formulário em papel ou eletrônico e reproduzir uma semelhança com ele. A semelhança pode incluir a formatação do próprio formulário ou pode conter apenas os dados que irão 'preencher' o formulário. Existem dois problemas básicos que uma solução de emulação de formulário deve resolver:

1 Como será gerada a formatação do formulário?

2 Como serão gerados os dados que preenchem o formulário?

O primeiro problema é resolvido usando uma faixa do tamanho de uma página no relatório ou um estilo de página. Um estilo de página pode ser projetado como uma faixa, mas pode ser gerado como um plano de fundo para as faixas do relatório.

O segundo problema é resolvido usando componentes simples com reconhecimento de dados ou usando componentes mais complexos de regiões ou sub-relatórios.

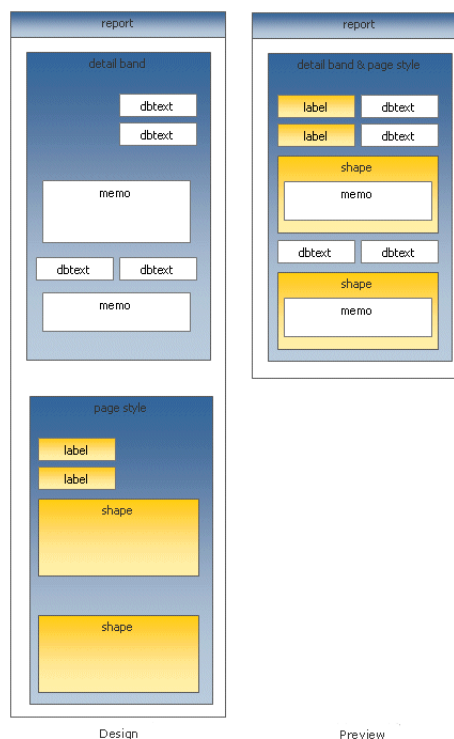
No ReportBuilder, existem várias abordagens que podem realizar a emulação de formulário. Essas abordagens são discutidas em detalhes nesta seção.

Formulários de página única

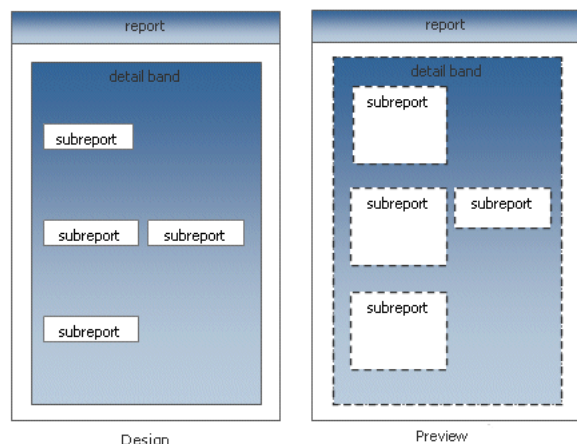
Formulários de página única podem ser emulados de várias maneiras:

1 Expanda a banda de detalhes até a altura imprimível da página, oculte as bandas de cabeçalho e rodapé e coloque todos os controles de formatação de formulário e de reconhecimento de dados na banda de detalhes. Essa abordagem produz um formulário por registro. O diagrama a seguir mostra um formulário inteiro na faixa de detalhes.

2 Adicione um estilo de página ao relatório. Coloque toda a formatação do formulário no estilo de página. Coloque os controles dataaware nas faixas como faria ao construir um relatório normal. Essa abordagem produz um número variável de registros por formulário, dependendo da altura das bandas. Nesta abordagem, a faixa de detalhes é usada para preencher o formulário. O diagrama a seguir mostra como uma banda de detalhes pode preencher um estilo de página.



3 Expanda a banda de detalhes para a altura imprimível da página. Oculte a faixa de cabeçalho e rodapé e coloque toda a formatação do formulário na faixa de detalhes. Coloque sub-relatórios do tipo filho em diferentes áreas do formulário onde uma tabela ou fonte de dados específica é necessária para preencher essa parte do formulário. Coloque componentes com reconhecimento de dados em áreas do formulário onde o pipeline de dados principal fornece os dados. Esse posicionamento produz a solução de emulação de formulário mais flexível e poderosa, mas só é necessária quando vários conjuntos de dados são usados. O diagrama a seguir mostra sub-relatórios preenchendo um formulário na faixa de detalhes.



Quando você tem um formulário em uma imagem WMF, o usuário final obtém uma excelente capacidade de visualização de impressão. Se você estiver imprimindo em formulários pré-impressos, poderá definir a propriedade visible como False quando o relatório for enviado para a impressora. Desta forma, você pode fornecer um formulário preenchido na janela de visualização da impressão, mas apenas o texto necessário para preencher o formulário é realmente enviado para a impressora.

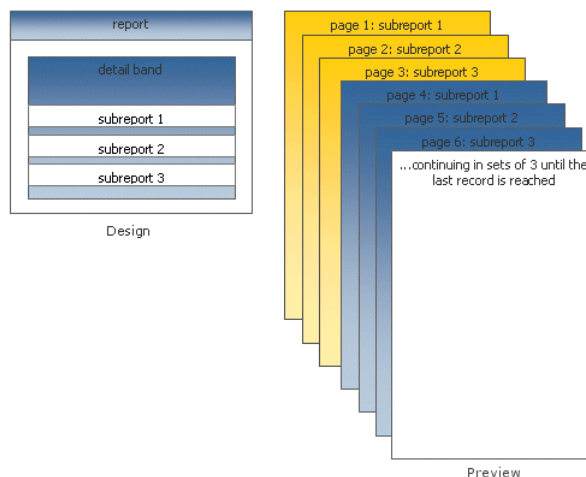
Você pode converter um formulário digitalizado em uma imagem de metarquivo do Windows (WMF) por meio de um produto como o Transform Suite, da MIPS.

O formulário em si pode ser muito complexo para recriar usando componentes de relatório. Nesse caso, é recomendável digitalizar o formulário e convertê-lo em um metarquivo do Windows. Você pode então colocar o metarquivo do Windows no relatório como um plano de fundo para a banda de detalhes ou pode colocá-lo no estilo de página, onde funcionará naturalmente como um plano de fundo para o relatório.

Formulários de várias páginas

Todas as abordagens discutidas no tópico de formulário de página única são aplicáveis a formulários de várias páginas. O problema adicional de organizar formulários de uma única página em conjuntos de formulários de várias páginas pode ser resolvido usando sub-relatórios de estilo de seção. Quando sub-relatórios de estilo de seção são renderizados, eles geram uma nova página dentro do relatório pai, continuam gerando páginas até que todos os dados tenham sido percorridos (ou eles são parados manualmente por meio de uma chamada de procedimento para o mecanismo de relatório) e, em seguida, retornam o controle ao pai relatório. Um comportamento especial dos sub-relatórios do tipo seção é que o relatório pai não gera páginas quando a seção é colocada em uma banda de detalhes de altura dinâmica. Nesse caso, o relatório principal é usado como plataforma de lançamento para as seções. Uma vez que cada seção é um relatório completo em seu próprio direito,

O diagrama abaixo mostra três sub-relatórios do tipo seção na faixa de detalhes do relatório principal. O relatório principal é atribuído a um pipeline de dados, de modo que a banda de detalhes será impressa uma vez para cada registro. Os sub-relatórios contêm controles com reconhecimento de dados que apontam para esse pipeline de dados. Este relatório fornece três páginas de conteúdo para cada registro, e cada página tem seu próprio formato exclusivo.



DATA

Introdução 49

Suporte de banco de dados 53

Alternativas de banco de dados 55

Arquivos de texto 57

Objetos Delphi 59

Acesso nativo a dados proprietários 61

DATA

Introdução

Visão geral

No ReportBuilder, o acesso aos dados é fornecido por meio do componente de pipeline de dados. ReportBuilder inclui pipelines de dados para acessar dados de uma variedade de fontes.

BDEPipeline

Retido para compatibilidade com versões anteriores.

Use o DBPipeline para novos aplicativos.

DBPipeline

Usado para acessar dados por meio de um descendente de TDataSet. Suporta ADO, BDE, dbExpress, IBExpress e conjuntos de dados de terceiros, como DBISAM, DOA, NexusDB, etc.

TextPipeline

Usado para acessar dados em arquivos de texto ASCII.

JITPipeline

Pipeline Just-In-Time para acessar dados em objetos Delphi.

Independentemente do tipo de pipeline ou do tipo de dados acessado, o componente de pipeline de dados tem duas finalidades básicas:

1 Para fornecer dados

2 Para controlar a passagem de dados

Fornecendo Dados

Os pipelines de dados fornecem dados por meio de campos.

Por exemplo, o código a seguir recuperaria o valor do campo atual de um campo denominado 'Empresa':

```
IValue: = DataPipeline1 ['Company'];
```

Cada vez que um componente de relatório com reconhecimento de dados é impresso, ele usa essa abordagem para recuperar os dados do pipeline de dados. Os componentes de relatório com reconhecimento de dados têm duas propriedades que determinam os dados que eles recuperarão: DataPipeline e DataField. Depois que essas duas propriedades são atribuídas, o componente com reconhecimento de dados tem a capacidade de recuperar dados diretamente do pipeline de dados, independentemente do relatório no qual o componente com reconhecimento de dados reside.

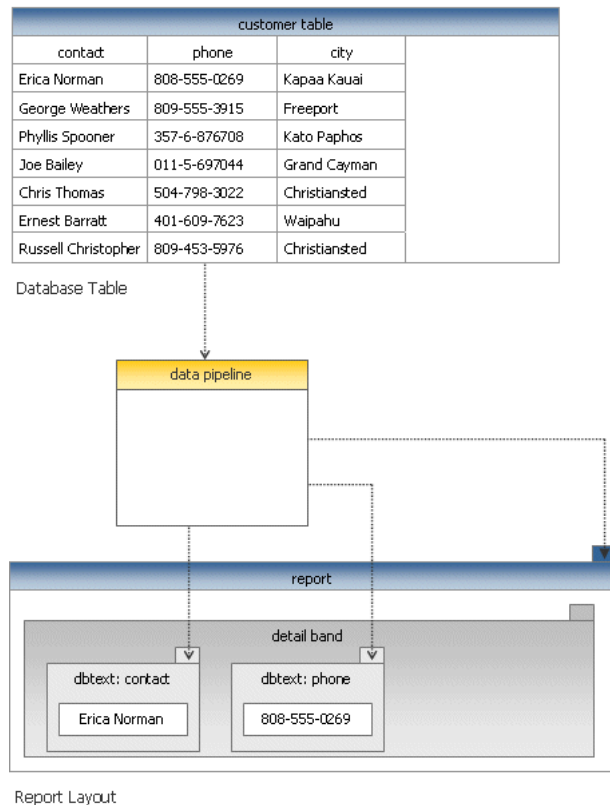
Controle de transferência de dados

O segundo objetivo do pipeline de dados é controlar a passagem de dados. A passagem de dados é o ato de passar do primeiro registro dos dados para o último registro. Quando um relatório é impresso, o mecanismo de relatório atravessa os dados concluindo as seguintes etapas:

- 1 Abre o pipeline de dados.
- 2 Vai para o primeiro registro.
- 3 Começa a imprimir a página e, em seguida, dá à banda de detalhes a oportunidade de imprimir.
- 4 Vai para o próximo registro.
- 5 Dá à banda de detalhes a oportunidade de imprimir.
- 6 Continua as etapas 4 e 5 até que não haja mais espaço na página.
- 7 Completa a página.
- 8 Continua as etapas 4 a 7 até que todos os registros tenham se esgotado.

É importante observar que quando a banda de detalhes tem a oportunidade de imprimir, os componentes com reconhecimento de dados dentro da banda de detalhes são renderizados; neste ponto, eles recuperam o valor do campo do registro atual. O mecanismo então se move para o próximo registro e imprime a banda de detalhes novamente. É essa combinação do relatório percorrendo os dados e os componentes dataaware recuperando os dados que cria as páginas do relatório. Se o pipeline de dados não for atribuído a nenhuma dessas entidades (com reconhecimento de dados

componente ou relatório), o relatório não funcionará. Portanto, o relatório deve ser atribuído a um pipeline de dados e cada componente com reconhecimento de dados deve ser atribuído a um pipeline de dados e a um campo de dados.



Dissemos que o mecanismo de relatório percorre os dados. Isso não é totalmente verdade. O mecanismo de relatório faz solicitações ao pipeline de dados (como abrir, primeiro, próximo, último) e, a seguir, depende do pipeline de dados para fazer o trabalho. Portanto, o pipeline de dados controla a passagem de dados. Este controle pode ser usado com grande vantagem.

Por exemplo, digamos que você esteja exibindo uma grade de banco de dados em um formulário. O usuário selecionou um registro individual da grade e deseja imprimir esse registro. Se você tiver um pipeline de dados apontado para a mesma fonte de dados que a grade, ele pode acessar

todos os registros. No entanto, podemos instruir o pipeline de dados a percorrer apenas o registro atual, simplesmente definindo as propriedades RangeBegin e RangeEnd como CurrentRecord. Quando o relatório for impresso, ele enviará solicitações de passagem ao pipeline de dados, e o pipeline de dados percorrerá apenas um registro. Em seguida, ele informará ao mecanismo de relatório que todos os registros foram percorridos e o relatório imprimirá apenas um registro.

Vamos levar este exemplo adiante e dizer que você permite que o usuário selecione vários registros da grade do banco de dados. Em seguida, você deseja que o relatório contenha apenas os registros selecionados. Nesse caso, podemos atribuir os favoritos da grade usando o método AddBookmark do pipeline de dados. Quando o relatório for impresso, o pipeline de dados percorrerá apenas os registros que estão na lista de marcadores e apenas esses registros aparecerão no relatório. Essencialmente, o mecanismo faz as mesmas solicitações de passagem do pipeline de dados para cada relatório, mas é o pipeline de dados que controla como os dados são realmente percorridos.

Suporte de banco de dados

Visão geral

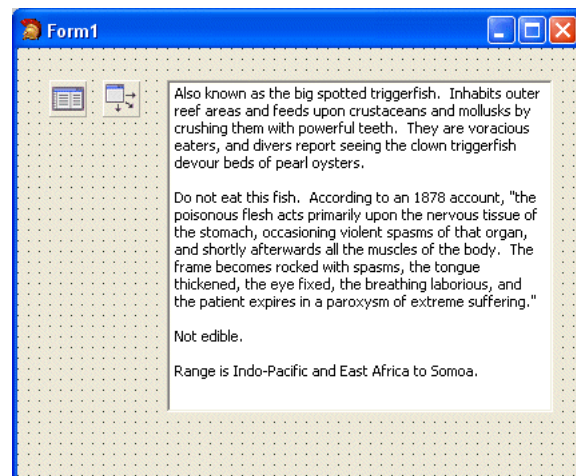
Delphi tem uma arquitetura forte para acesso a banco de dados. Enquanto as primeiras versões do Delphi dependiam de uma tecnologia de propriedade chamada BDE (Borland Database Engine), a Borland posteriormente redesenhou a biblioteca de classes de acesso a dados para ser aberta e extensível. Hoje, o Delphi inclui componentes de acesso a dados que suportam várias tecnologias diferentes, incluindo BDE, ADO, IBExpress e dbExpress. Além disso, um grande número de soluções de acesso a dados de terceiros estão disponíveis e são amplamente utilizadas.

O ReportBuilder aproveita totalmente o poder e a flexibilidade fornecidos pela arquitetura de acesso a dados da Delphi. Assim, você tem a liberdade de escolher a solução de acesso a dados que melhor se adapta aos requisitos de seus clientes e aplicativos.

Acesso de dados

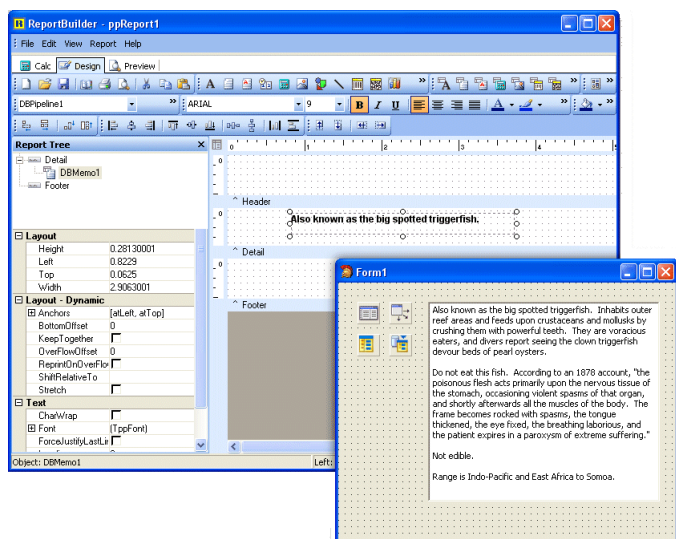
No Delphi, a classe TDataSet é a classe base abstrata para todos os componentes do conjunto de dados que representam dados em linhas e colunas. Cada tipo de conectividade, como BDE, ADO, etc. inclui um ou mais descendentes de conjuntos de dados que são instalados na paleta de componentes do Delphi e podem ser usados em seus aplicativos.

Os controles com reconhecimento de dados são normalmente conectados a um conjunto de dados por meio de um componente TDataSource. A captura de tela a seguir mostra um valor de campo sendo recuperado em um componente de memorando com reconhecimento de dados. Neste exemplo, um DBMemo está conectado ao TDataSource que está conectado ao componente TTable. (O componente TTable é uma classe descendente de TDataSet fornecida pela biblioteca de classes BDE.)



ReportBuilder aumenta esse modelo de acesso a dados adicionando outro componente: o pipeline de dados.

Existem vários tipos de pipelines de dados, mas para fins de conexão a um conjunto de dados, um componente DBPipeline é usado. A tela a seguir mostra um DBPipeline recuperando o mesmo valor de campo.



O valor é exibido em um componente de memorando de dataaware ReportBuilder. Você pode notar que apenas a primeira linha do memorando é mostrada no ReportBuilder Report Designer. Isso se deve ao fato de que, quando o relatório for gerado, o componente memo irá recalcular a altura com base no texto que ele contém. Portanto, não há razão para especificar a altura do memorando no layout do relatório, pois ele será redimensionado automaticamente na página gerada.

Alternativas de banco de dados

Visão geral

O ReportBuilder aproveita totalmente o poder e a flexibilidade fornecidos pela arquitetura de acesso a dados da Delphi. Assim, você tem a liberdade de escolher a solução de acesso a dados que melhor se adapta aos requisitos de seus clientes e aplicativos. Algumas das soluções de acesso a dados mais populares usadas por nossos clientes estão listadas abaixo.

ALVOROÇO

Incluído no Delphi. Usado para conectar ao MS SQL Server e MS Access

IBExpress

Incluído no Delphi. Usado para conectar ao Interbase e Firebird.

DBExpress

Incluído no Delphi. Usado para conectar-se a vários produtos de banco de dados diferentes. dbExpress tem uma arquitetura baseada em driver que pode ser estendida para suportar qualquer banco de dados. Delphi inclui um conjunto padrão de drivers dbExpress, e drivers dbExpress de terceiros estão cada vez mais disponíveis.

Acesso direto ao Oracle (DOA)

por Allround Automations Usado

para se conectar ao Oracle.

mySQLDAC

por MicrOLAP

Usado para conectar ao MySQL.

Vantagem

por iAnywhere

Usado para se conectar ao Advantage Database Server.

DBISAM e ElevateDB

por Elevate Software

Produtos de banco de dados Delphi nativos que podem ser incorporados em aplicativos.

NexuxDB

por Nexus Database Systems

Produto de banco de dados Delphi nativo que pode ser embutido em aplicativos.

Arquivos de Texto

O componente TextPipeline

Os relatórios podem ser impressos diretamente de arquivos de texto sem usar um produto de banco de dados. Essa funcionalidade é fornecida por meio do componente TextPipeline.

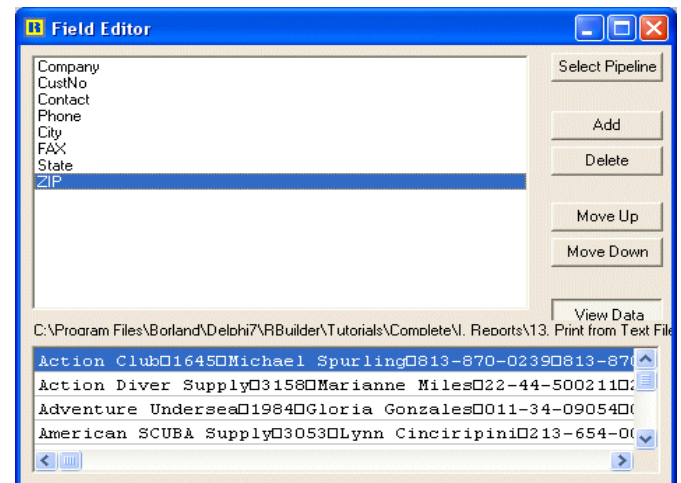
Os seguintes formatos são suportados:

- Delimitado por vírgulas
- Delimitado por tabulação
- Registros de comprimento fixo
- Delimitado personalizado (onde você especifica o delimitador)

O componente TextPipeline é essencialmente um mecanismo de recuperação de dados simples que permite acessar dados em arquivos de texto da mesma maneira que os dados armazenados em uma tabela de banco de dados. Você pode até definir relacionamentos mestre / detalhes entre os dados em dois arquivos de texto. O TextPipeline contém um Editor de campo que é usado para definir os campos de dados para o arquivo de texto. Depois que os campos de dados forem definidos, você pode acessar o Report Designer e atribuir esses campos aos componentes de reconhecimento de dados do ReportBuilder.

O Editor de Campo

Cada um dos pipelines de dados no ReportBuilder tem um Editor de campo que pode ser acessado clicando duas vezes no componente do pipeline em tempo de design. O Editor de campo do TextPipeline é ilustrado abaixo. Observe que o botão Exibir Dados foi ativado para exibir o conteúdo do arquivo de texto. Este recurso pode ser uma ajuda útil ao definir os campos para o arquivo de texto.



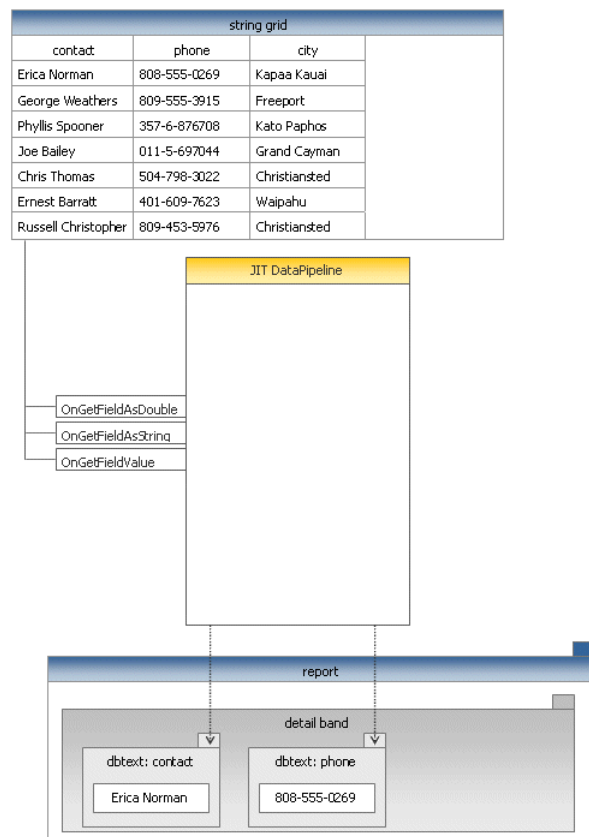
Objetos Delphi

Visão geral

Os pipelines de dados apresentam um conjunto estruturado de dados para um relatório. Essa estrutura assume a forma de registros e campos. O mecanismo de relatório espera que operações como abrir, primeiro, depois e por último forneçam uma determinada resposta do pipeline de dados. Os componentes com reconhecimento de dados dentro do relatório esperam que os valores dos campos sejam recuperados simplesmente passando um nome de campo para o pipeline de dados. No componente DBPipeline, essa funcionalidade é implementada chamando os métodos de um objeto TDataSet. No entanto, essa não é a única maneira de um pipeline de dados fornecer a funcionalidade de acesso a dados necessária para um relatório. O JITPipeline (JIT significa Just-In-Time) aciona eventos para obter os mesmos resultados que o DBPipeline.

JITPipeline

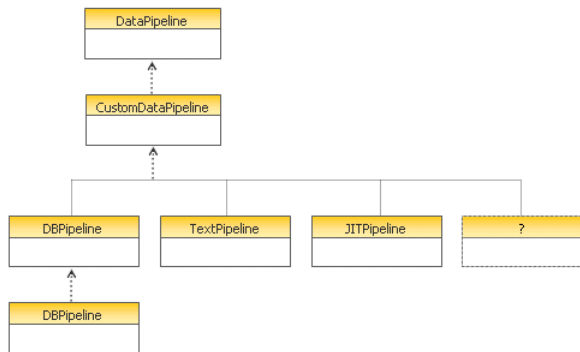
O JITPipeline fornece um conjunto de eventos que, quando codificados corretamente, permitem a impressão de relatórios de qualquer fonte de dados referenciada pelos manipuladores de eventos. O diagrama a seguir mostra um JITPipeline que é atribuído a manipuladores de eventos que fazem referência a uma grade de string Delphi padrão. No relatório, os componentes com reconhecimento de dados são criados da mesma forma que seriam para qualquer outro relatório de banco de dados - atribuindo nomes de campo do JITPipeline aos controles com reconhecimento de dados no relatório. Isso resulta em uma implementação limpa e sustentável. Se, posteriormente, você decidir fornecer dados ao relatório por meio de um arquivo de texto ou tabela de banco de dados, pode simplesmente trocar o JITPipeline por um novo pipeline do tipo correto.



Acesso nativo a dados proprietários

Visão geral

Se você tiver uma quantidade considerável de dados em um formato proprietário e muitos relatórios precisarem ser criados com base nesses dados, o mais alto nível de manutenção e facilidade de uso será fornecido por um componente de pipeline de dados personalizado. Esse tipo de componente pode ser criado descendo da classe `TppCustomDataPipeline` e implementando os métodos necessários. O resultado final de tal esforço será um novo componente de pipeline de dados que pode ser instalado no Delphi IDE e usado da mesma forma que os outros pipelines de dados fornecidos com o ReportBuilder. O modelo de objeto para as classes ReportBuilder `DataPipeline` é mostrado abaixo:



CTRIBUTO

O Delphi Event Model 65

Configuração Dinâmica 67

Executando Cálculos 73

Criando Relatórios no Código 83

CTRIBUTO

O modelo de evento Delphi

Visão geral

No contexto dos componentes Delphi, os eventos têm duas qualidades fundamentais:

1 Os eventos são disparados como resultado lógico de uma ação realizada em ou por um componente.

2 Os eventos disparam no momento em que uma ação significativa pode ser realizada.

Significado

Um evento é algo que acontece como resultado lógico de uma ação realizada em ou por um componente. Por exemplo, uma ação realizada em um controle de botão é um clique do mouse. Em resposta, um controle TButton disparará seu evento OnClick. Um exemplo de ação realizada por um componente seria o método Next de uma TTable. Quando este método é chamado, o evento OnCalcFields (entre outros) é disparado.

Tempo

Os eventos disparam no momento em que uma ação significativa pode ser realizada. No caso de um evento TButton.OnClick, a ação significativa seria aquela pretendida pelo desenvolvedor como a finalidade básica do botão. Essa finalidade pode ser abrir outra janela ou fechar o formulário atual. Mas nem toda ação é apropriada dentro do contexto do evento. Certamente não seria apropriado liberar o botão no OnClick ou chamar o manipulador de eventos OnClick de dentro dele. Essas chamadas podem ser OK para outros eventos do aplicativo, mas não dentro do contexto do evento OnClick.

Eventos ReportBuilder

No ReportBuilder, uma definição de relatório é composta por um conjunto de objetos: Report.Bands []. Objetos [] em que cada banda representa uma região retangular da página que contém Lables, DBText, Linhas, Imagens, etc. O relatório e cada subelementos é um objeto com um rico conjunto de propriedades e eventos. O arquivo de ajuda ReportBuilder contém informações de referência sobre cada propriedade e evento. Muitos dos demos instalados com o produto mostram como usar os eventos para controlar a geração de relatórios. Aqui estão algumas diretrizes gerais.

Report.OnIntializeParameters

Este evento é disparado sempre que um relatório é gerado por meio de uma chamada ao método Print. É o primeiro evento a disparar. Use este evento para inicializar Report.Parameters [] e Report.AutoSearchFields, exibir uma caixa de diálogo personalizada ou inicializar uma consulta SQL.

Report.BeforeOpenDataPipelines

BeforeOpenDataPipelines é acionado antes do mecanismo de relatório abrir as linhas de dados associadas ao relatório, relatórios filho e controles com reconhecimento de dados. Use este evento para aplicar valores de parâmetros personalizados ao SQL ou gerar SQL personalizado.

Band.BeforePrint

Use o evento BeforePrint de uma banda para executar ações que determinam o que é impresso. Por exemplo, para definir a propriedade Visible da banda ou a propriedade Visible dos objetos contidos na banda.

Variable.OnCalc

Use o evento Variable OnCalc para realizar cálculos.

Nota: Assim que o relatório começar a ser gerado, não use manipuladores de eventos para manipular os dados subjacentes associados ao relatório de nenhuma maneira.

Configuração Dinâmica

Configurar relatórios durante a geração

Quando falamos sobre a configuração de um relatório durante a geração, queremos dizer qualquer coisa, desde definir a legenda de um rótulo até controlar condicionalmente a visibilidade de um sub-relatório inteiro. Lembre-se sempre de que os relatórios do ReportBuilder são compostos por uma coleção de objetos: Report.Bands []. Objetos []. O objeto de relatório é o pai de uma coleção de objetos de banda. Cada objeto de banda representa uma área retangular da página e contém uma coleção de objetos imprimíveis, como texto, imagens e memorandos. Todos esses objetos têm propriedades e eventos. Os eventos são disparados durante a geração do relatório, permitindo-nos manipular as propriedades e controlar o comportamento do relatório.

Para ter uma ideia prática de como isso realmente funciona, vamos codificar alguns manipuladores de eventos que darão uma ideia do que pode ser feito durante a geração de um relatório.

Cor da fonte

O manipulador de eventos abaixo é atribuído ao evento OnPrint de um componente DBText. Quando o componente é impresso, o valor atual do campo PRICE_CHG é verificado. Se este valor for menor que zero, o componente imprime em vermelho; caso contrário, o componente será impresso em preto.

É importante observar que a cor da fonte é definida com a cor apropriada sempre que o componente é impresso. Se o código abaixo simplesmente definir a cor da fonte para vermelho quando o valor for negativo, o primeiro valor negativo será impresso em vermelho e todos os valores subsequentes serão impressos em vermelho, independentemente do sinal do valor. Ao definir as propriedades do componente do relatório, lembre-se de lidar com todos os casos, pois os componentes permanecem em qualquer estado para o qual foram definidos.

Código *Definir a cor de um componente DBText numérico com base no sinal*

```
procedimento Form1.ppDBText4Print (Sender: TObject);  
começar  
  
  E se(DBPipeline1 ['PRICE_CHG']>= 0)  
  então  
    ppDBText4.Font.Color:= clBlack  
  senão  
    ppDBText4.Font.Color:= clRed;  
fim;
```

Concatenação

Este manipulador de eventos é atribuído ao evento OnPrint de um componente Label. Quando o componente é impresso, o valor atual dos campos FirstName e LastName são recuperados. Se o FirstName tiver um comprimento maior que zero, ele será incluído na legenda. As variáveis locais usadas aqui são prefixadas com 'ls', que significa string local.

Código *Usando um rótulo para exibir valores de campo concatenados*

```
procedimento TForm1.ppLabel3Print (Sender: TObject);  
var  
    lsFirstName: String;  
    lsLastName: String;  
  
    começar  
        lsFirstName = DBPipeline1 ['FirstName']; lsLastName:  
        = DBPipeline1 ['LastName'];  
  
        E se (Comprimento (lsFirstName) > 0) então  
            ppLabel3.Caption := lsFirstName + " " + lsLastName  
        senão  
            ppLabel3.Caption := lsLastName;  
  
fim;
```

Address Squeeze

O evento BeforePrint de uma banda é um local aceitável para configurar qualquer componente dentro da banda. Colocar várias etapas de configuração de componentes diferentes em um manipulador de eventos de nível de banda pode tornar o código mais sustentável (em oposição a colocar o

código no OnPrint de cada componente). Este manipulador de eventos recupera os valores de campo apropriados de uma tabela de banco de dados, concatena-os em uma variável de string local e, em seguida, adiciona essa variável como uma linha ao memorando.

Código Usando um memorando para criar informações de endereço de tamanho dinâmico

procedimento TForm1.ppReport1DetailBand1BeforePrint (Sender: TObject);

var

IsLine: String;
IsState: String;
IsZIP: String;

começar

{limpar memo}

ppMemoAddress.Lines.Clear;

{adicionar contato}

IsLine: = DBPipeline1 ['Contato'];

ppMemoAddress.Lines.Add (IsLine);

{adicionar empresa}

IsLine: = DBPipeline1 ['Company'];

ppMemoAddress.Lines.Add (IsLine);

{adicionar linha de endereço 1}

IsLine: = DBPipeline1 ['Addr1'];

E se IsLine <> " **então**

ppMemoAddress.Lines.Add (IsLine);

{adicionar linha de endereço 2}

IsLine: = DBPipeline1 ['Addr2'];

E se IsLine <> " **então**

ppMemoAddress.Lines.Add (IsLine);

{adicionar cidade, CEP do estado}

IsLine: = DBPipeline1 ['Cidade']; IsState: =
DBPipeline1 ['State'];

E se IsState <> " **então**

IsLine: = IsLine + ',' + IsState;

IsZIP: = DBPipeline1 ['ZIP'];

E se IsZIP <> " **então**

IsLine: = IsLine + " " + IsZIP;

ppMemoAddress.Lines.Add [IsLine];

{adicionar país}

IsLine: = DBPipeline1 ['País'];

ppMemoAddress.Lines.Add (IsLine);

fim;

Grupo Continuado

Este manipulador de eventos é anexado ao evento OnPrint de um rótulo na faixa de detalhes do relatório. A instrução 'if' verifica se esta é a primeira página do grupo e a primeira faixa de detalhes da página. Se não for a primeira página do grupo, o grupo está imprimindo em páginas adicionais. Nesse caso, o rótulo fica visível. A legenda do rótulo é definida como 'Continuação ...'; isso cria o efeito de rotular linhas de detalhe que são impressas em páginas adicionais para o mesmo grupo.

Código Colocando um rótulo 'Continuação ...' em uma banda de detalhes

```
procedimento TForm1.ppLabelContinuedPrint (Sender: TObject);  
começar  
    se não(ppReport1.Groups [0] .FirstPage) e  
        (ppReport1.DetailBand1.Count = 1) então  
            ppLabelContinued.Visible: = True  
senão  
    ppLabelContinued.Visible: = False;  
fim;
```

Regiões

A banda do cabeçalho do grupo neste relatório contém duas regiões. Uma região contém um conjunto detalhado de dados; a outra região contém um conjunto mais resumido dos mesmos dados. Na primeira página do grupo, o conjunto detalhado de dados (na região 1) é exibido. Se o grupo continuar em páginas adicionais, a versão mais resumida dos dados (na região 2) será exibida. A propriedade FirstPage do grupo é usada para determinar se o relatório está na primeira página ou em páginas adicionais do grupo. No layout do relatório, a Região 2 foi colocada à direita da Região 1 para facilitar a manutenção do layout do relatório. Portanto, a primeira vez que a Região 2 é exibida, ela deve ser movida para a mesma posição da Região 1.

Código *Controle de componentes usando regiões*

```
procedimento TForm1.ppGroupHeaderBand1BeforePrint (Sender: TObject);  
começar  
  E se ppGroup1.FirstPage então  
    começar  
      ppRegion1.Visible: = True;  
      ppRegion2.Visible: = False;  
    fim  
  senão  
    começar  
      ppRegion1.Visible: = False;  
      ppRegion2.Visible: = True;  
  
    E se (ppRegion2.Left <> ppRegion1.Left) então  
      ppRegion2.Left: = ppRegion1.Left;  
    fim;  
fim;
```

SubReports

Este manipulador de eventos está associado a um relatório mestre / detalhado. O relatório contém um sub-relatório que imprime os detalhes. Este relatório reside em um formulário com um botão intitulado 'Ocultar detalhes'. Quando o botão é clicado, o manipulador de eventos alterna o valor da propriedade visible do sub-relatório e, a seguir, imprime o relatório. Como resultado, o relatório é um resumo de alto nível ou uma lista de detalhes completos; assim, o relatório fornece a funcionalidade de dois relatórios de apenas um layout.

Código *Controlando a visibilidade de um sub-relatório*

```
procedimento TfrmShowHideDetail.btnDetailClick (Sender: TObject);  
começar  
    ppSubReport1.Visible: = not (ppSubReport1.Visible);  
  
    E se (ppSubReport1.Visible) então  
        btnDetail.Caption: = 'Ocultar detalhes'  
    senão  
        btnDetail.Caption: = 'Mostrar detalhes';  
  
    ppReport1.Print;  
  
fim;
```

Executando Cálculos

Visão geral

Os cálculos são uma parte vital dos relatórios, e o ReportBuilder fornece um rico conjunto de componentes e eventos que permitem realizar uma ampla variedade de cálculos. Cálculos simples podem ser executados sem qualquer codificação por meio do componente DBCalc. Este componente fornece as funções Soma, Média, Mínimo, Máximo e Contagem. Os cálculos podem ser baseados em grupos ou baseados em relatórios. Cálculos mais complexos podem ser realizados por meio do componente Variável. No mínimo, o componente Variável requer que um manipulador de eventos OnCalc seja atribuído. O tempo desse evento pode ser correlacionado a qualquer número de ocorrências no processo de geração de relatório. Por exemplo, o componente pode calcular uma vez para cada uma das seguintes ocorrências: o início do relatório, uma travessia de registro, uma quebra de grupo, o início de uma página e o início de uma coluna. O componente Variável também possui um evento OnReset que pode ser correlacionado de forma semelhante. Isso fornece o máximo em flexibilidade de cálculo.

Componente DBCalc

O componente DBCalc:

- Executa cálculos simples sem qualquer codificação
- Fornece funções de Soma, Média, Mínimo, Máximo e Contagem
- Executa cálculos que podem ser baseados em grupos ou relatórios

Componente variável

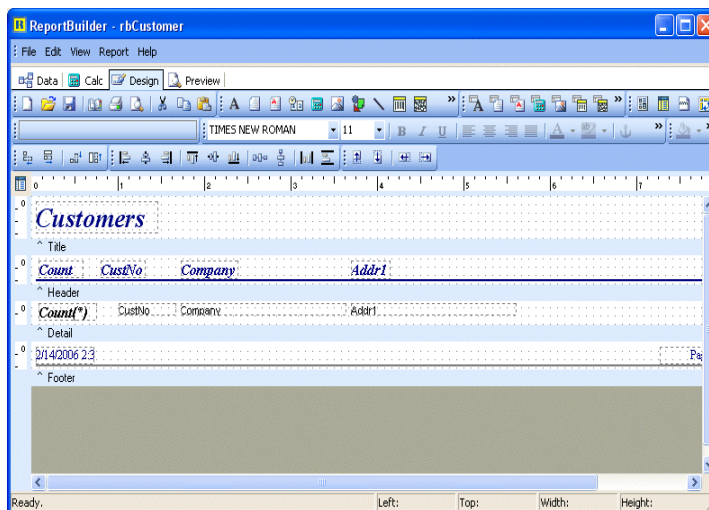
O componente variável:

- Requer a atribuição de um manipulador de eventos OnCalc
- Contém propriedades para controlar o tempo dos eventos OnCalc e OnReset
- Contém uma propriedade CalcOrder para controlar a ordem dos cálculos

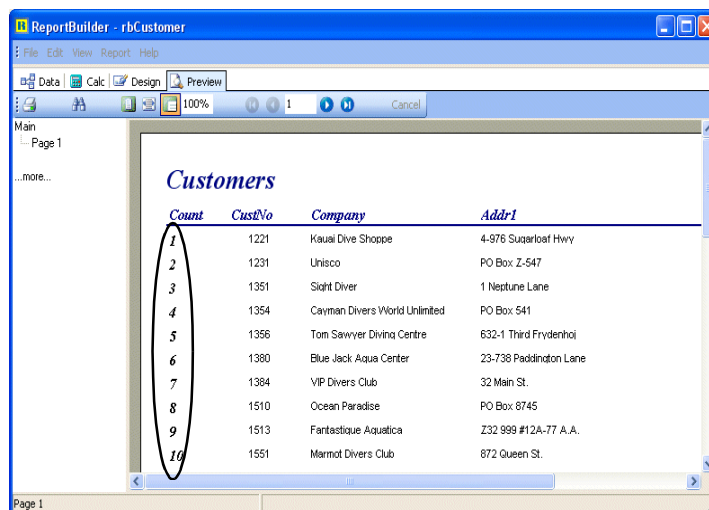
Contar

Para numerar as bandas de detalhes, coloque um componente DBCalc na banda de detalhes de um relatório. Clique com o botão direito sobre o componente e selecione o item de menu Cálculos Uma caixa de diálogo será exibida. Defina o tipo de cálculo como Contagem e clique em OK. Ao retornar ao Report Designer, o DBCalc conterá esta legenda: 'Contagem (*)'. Não há necessidade de selecionar um campo individual para o DBCalc porque a função Contagem não exige um. Visualize o relatório. Cada banda de detalhes será numerada conforme mostrado abaixo.

Usando um componente DBCalc para numerar a banda de detalhes



Vista de Design



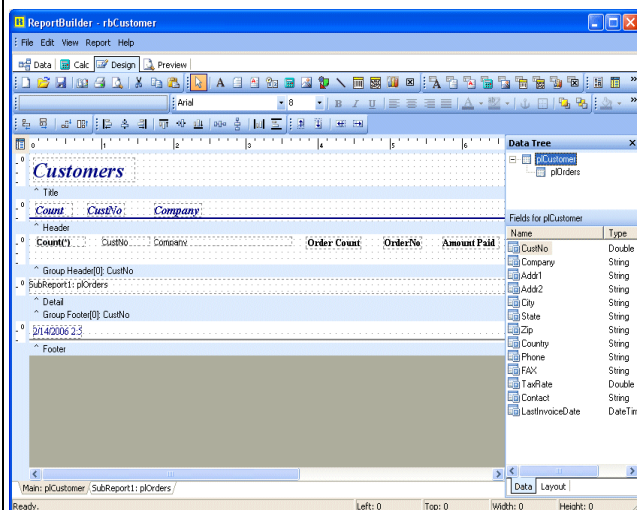
Antevisão

Mestre / Detalhe de Contagem

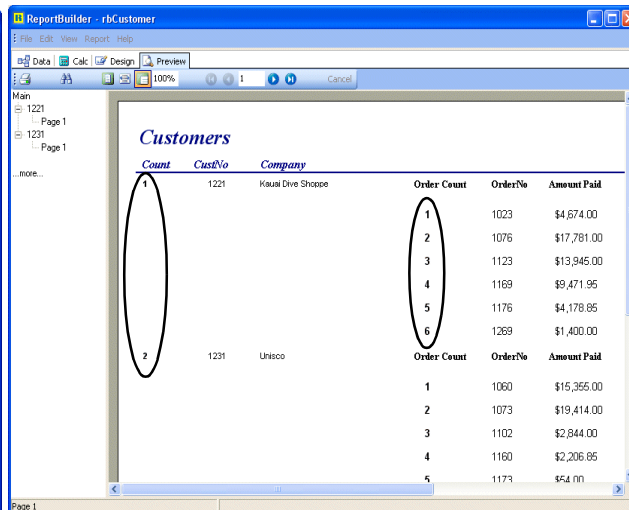
Um relatório mestre / detalhado geralmente contém um grupo com base no campo-chave da tabela mestre. Isso permite que os campos da tabela mestre sejam impressos no cabeçalho do grupo, em vez de serem repetidos na banda de detalhes. Para contar os registros mestre, coloque um DBCalc na banda de cabeçalho do grupo e defina o CalcType para contar. Em seguida, acesse a caixa de diálogo Cálculos ... e limpe a configuração Redefinir grupo. O ResetGroup é atribuído automaticamente quando um DBCalc é colocado em uma banda de grupo. Quando definida, essa propriedade faz com que o DBCalc seja redefinido para zero sempre que o grupo for interrompido. Aqui, queremos contar cada grupo (não zerar quando cada grupo for interrompido) e, portanto, eliminamos esta tarefa.

Em seguida, coloque um componente DBCalc na banda de detalhes do sub-relatório e defina o Calc Type para contar. Quando o relatório é visualizado, os registros mestre e os registros de detalhes serão numerados conforme mostrado abaixo

Usando um componente DBCalc para numerar os registros mestre e de detalhes



Vista de Design

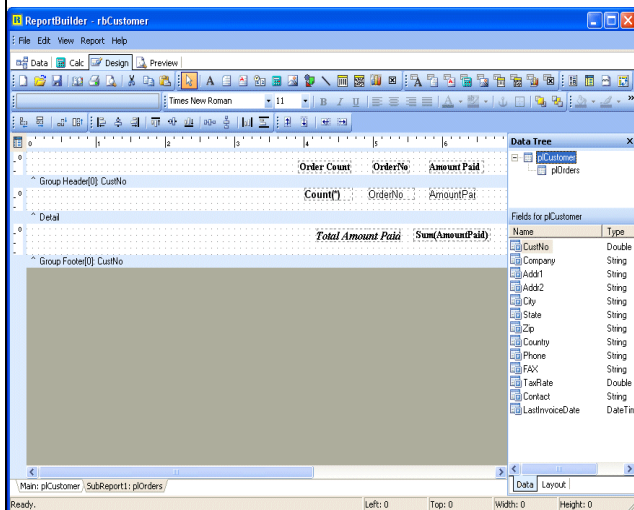


Antevisão

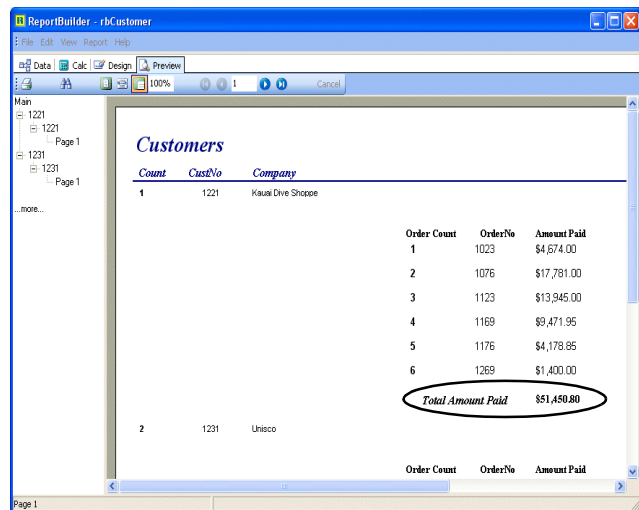
Total do Grupo

Os totais do grupo de relatórios podem ser facilmente calculados colocando um DBCalc na faixa de rodapé do grupo. O ResetGroup é atribuído automaticamente quando um DBCalc é colocado em uma banda de grupo. Assim, cada vez que o grupo for interrompido, o total do grupo será redefinido para 0.

Usando um componente DBCalc para numerar os registros mestre e de detalhes



Vista de Design

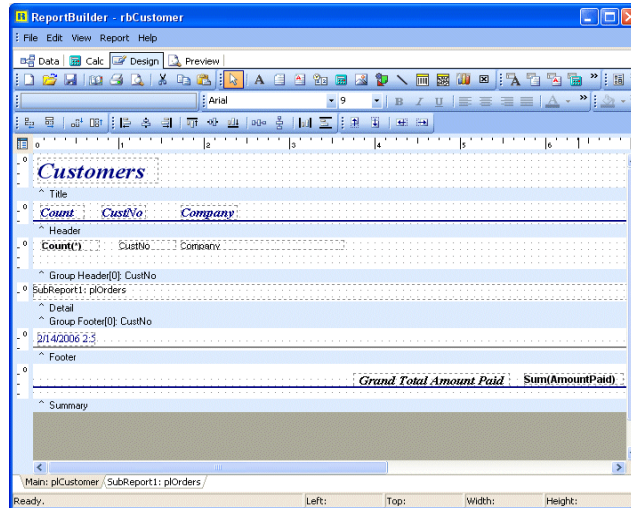


Antevisão

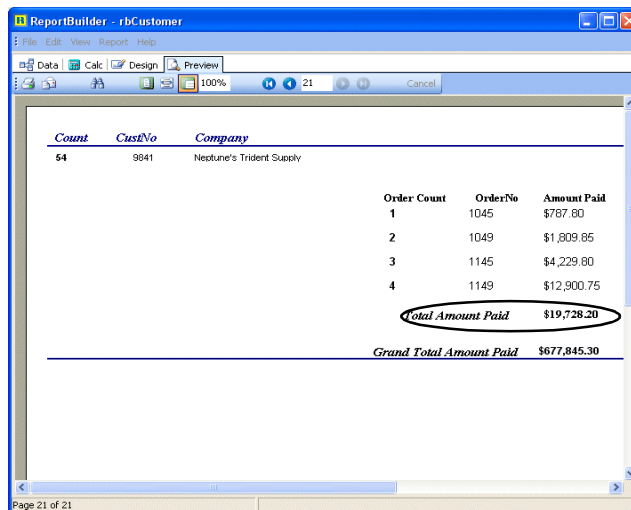
Total geral

Para calcular um total geral, adicione uma faixa de resumo a um relatório e coloque um DBCalc na faixa de resumo.

Usando um componente DBCalc para calcular um total geral



Vista de Design

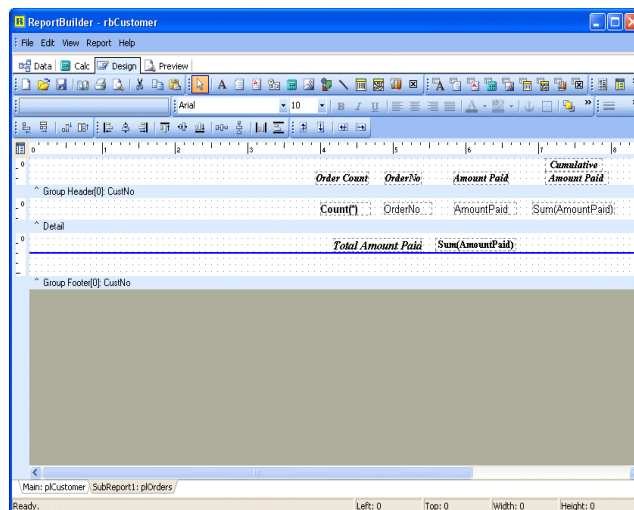


Antevisão

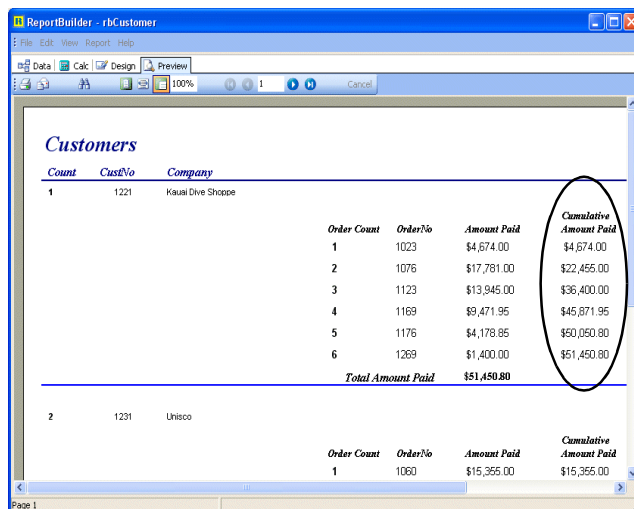
Soma Cumulativa

Para calcular uma soma cumulativa na banda de detalhes, adicione um DBCalc na banda de detalhes e atribua-o ao campo e pipeline de dados apropriados.

Usando um componente DBCalc para calcular uma soma cumulativa na banda de detalhes



Vista de Design



Antevisão

Total condicional do grupo

Quando você precisar excluir certos valores de um total de grupo, adicione um componente Variável à faixa de rodapé do grupo. Acesse a caixa de diálogo Timing e defina 'Calculate On' para DataPipeline Traversal, selecione o Data Pipeline apropriado e, em seguida, defina 'Reset On' para Group Start e selecione o grupo apropriado. Codifique o manipulador de eventos OnCalc como:

Este manipulador de eventos acumulará o valor da variável apenas quando o valor pago for maior que \$ 5.000.

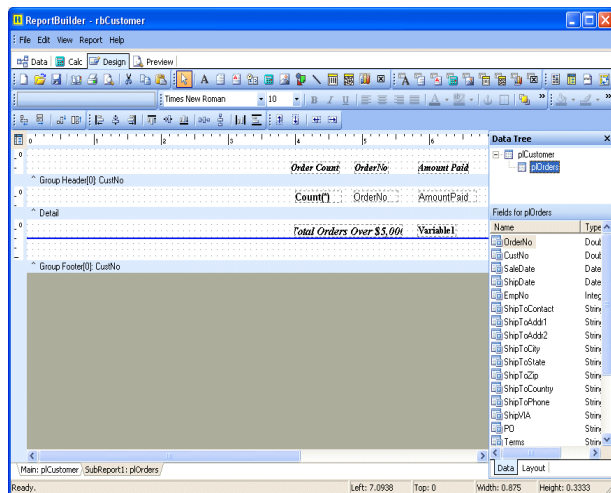
```

procedimento TForm1.ppVariable1Calc (Sender: TObject; var Value: Variant);
var
    lcValue: Moeda;

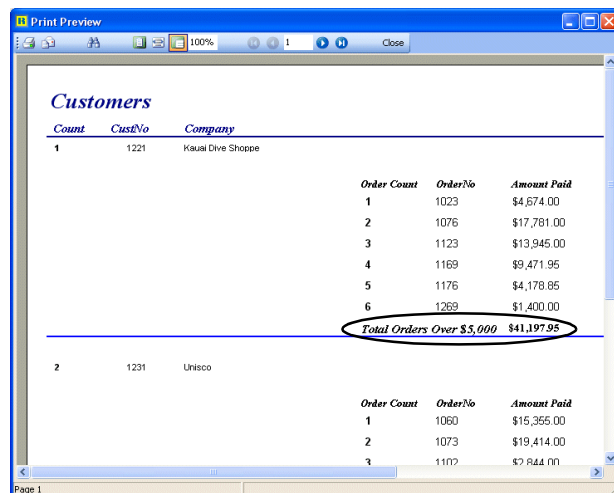
começar
    lcValue := Table2.FieldByName ('AmountPaid'). AsCurrency;

    E se (lcValue >= 5000) então
        Valor := Valor + lcValue;
fim;
    
```

Usando uma variável para excluir certos valores de um total do grupo



Vista de Design



Antevisão

Grande Total Condicional

Para excluir certos valores de um total geral, adicione um componente Variável à faixa de resumo. Em seguida, codifique o manipulador de eventos OnCalc como:

```
Procedimento TForm1.ppVariable2Calc (Sender: TObject; var Value: Variant);
var
    lcValue: Moeda;

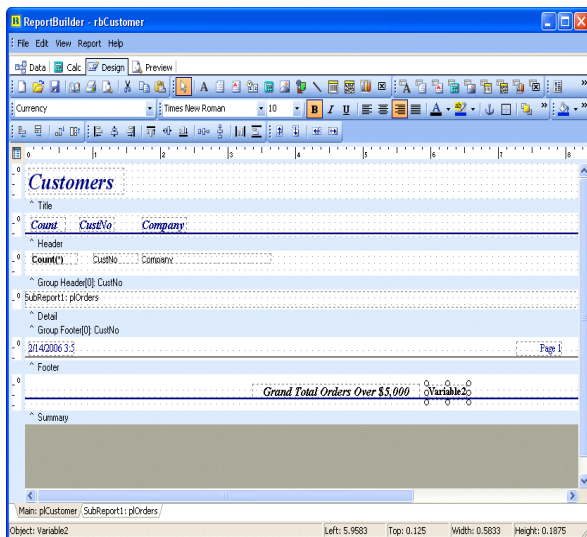
começar
    lcValue := Table2.FieldByName ('AmountPaid'). AsCurrency;

    E se (lcValue >= 5000) então
        Valor := Valor + lcValue;

fim;
```

Este manipulador de eventos acumulará o valor da variável apenas quando o valor pago for maior que \$ 5.000. A captura de tela abaixo mostra o resultado. Observe que este relatório também tem um total de grupo condicional na faixa de rodapé do grupo.

Usando uma variável para excluir certos valores de um total geral



Vista de Design

Order Count	OrderNo	Amount Paid
1	1040	\$3,632.00
2	1140	\$1,240.00
Total Orders Over \$5,000		\$0.00
54	9841	Nestune's Trident Supply
Order Count	OrderNo	Amount Paid
1	1045	\$767.80
2	1049	\$1,809.85
3	1145	\$4,229.80
4	1149	\$12,900.75
Total Orders Over \$5,000		\$12,900.75
Grand Total Orders Over \$5,000		\$2,544,261.25

Antevisão

Total Look Ahead

O ReportBuilder tem flexibilidade para operar como um mecanismo de relatório de uma ou duas passagens. Este exemplo mostra como calcular os totais do grupo para que possam ser exibidos no cabeçalho do grupo.

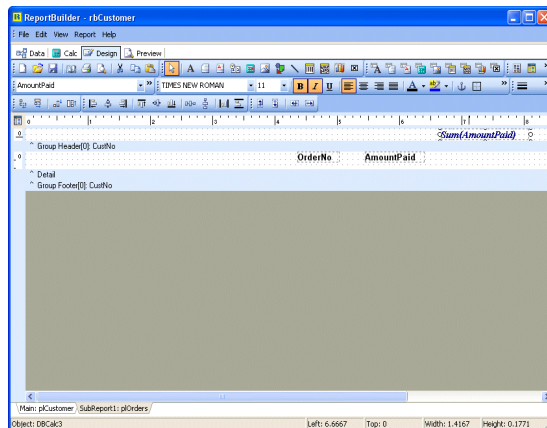
1 Crie um grupo.

2 Coloque um componente DBCalc na banda de cabeçalho do grupo.

3 Clique com o botão direito e defina a propriedade LookAhead como True.

4 Pré-visualizar. O total de impressões antes do detalhe.

Usando um DBCalc e uma variável para exibir um total de grupo no início de um grupo



Vista de Design

The screenshot shows the ReportBuilder Preview View for the same report. The report displays a table of customer orders with a total amount paid for each customer group. The table is titled 'Customers' and has columns 'Count', 'CustNo', 'Company', and 'Amount Paid'.

Count	CustNo	Company	Amount Paid
1	1221	Kauai Olive Shoppers	\$51,450.80
		1023	\$4,674.00
		1076	\$17,781.00
		1123	\$13,945.00
		1169	\$9,471.95
		1176	\$4,178.85
		1269	\$1,400.00
2	1221	Unico	\$85,643.60
		1060	\$15,355.00
		1073	\$19,414.00
		1102	\$2,844.00
		1160	\$2,206.85
		1173	\$54.00
		1178	\$5,511.75

The status bar at the bottom indicates: Page 1 of 11.

Antevisão

Criação de relatórios em código

Codificando um Relatório

Um layout de relatório é composto por um conjunto de componentes. Como outros componentes Delphi padrão, os componentes que compõem um layout de relatório têm uma interface de tempo de execução. Ou seja, eles podem ser criados e configurados usando o código Object Pascal. Esse recurso pode ser usado para criar relatórios inteiros dinamicamente. Siga estas etapas para criar um relatório dinamicamente:

1 Crie componentes de acesso a dados.

2 Crie o relatório.

3 Crie as faixas de relatório.

4 Adicione componentes com reconhecimento de dados.

5 Adicione componentes de formatação.

Essas são as mesmas etapas que você seguiria para criar um relatório usando o IDE Delphi e o ReportBuilder Report Designer. Criar os componentes manualmente requer algum conhecimento adicional de como as diferentes partes do relatório se encaixam. A discussão a seguir o orienta através do código real necessário para construir um relatório simples.

1 Declare a cláusula de usos necessária.

usa

```
DB,           {contém TDataSource}
DBTables,    {contém TTable}
ppReport,    {contém TppReport}
ppDBBDE,     {contém TppBDEPipeline} {contém
ppBands,     todas as classes de banda} {contém
ppCtrls,     componentes padrão} {contém todos os
ppTypes,     tipos enumerados do ReportBuilder}

ppVar;       {contém as classes SystemVariable e
              Variable}
```

Os tipos que estão sendo usados em cada unidade estão documentados nos comentários deste código.

2 Declare as variáveis locais necessárias para criar o relatório.

```
procedimento TForm1.Button1Click (Remetente:
    TObject);
var
    ITable: TTable;
    IDataSource: TDataSource;
    IDataPipeline: TppBDEPipeline; IReport:
    TppReport;
    ILabel1: TppLabel;
    ILabel2: TppLabel;
    IDBText1: TppDBText;
    IDBText2: TppDBText;
    ISysVar: TppSystemVariable;
```

O prefixo 'I' significa local. Este padrão de codificação torna as variáveis locais facilmente distinguíveis das variáveis de nível de componente (que são prefixadas com um 'F') e de nomes de componentes apropriados (como 'Relatório1').

3 Crie componentes de acesso a dados.

```
ITable := TTable.Create (Self); ITable.Name :=
'tblCustomer'; ITable.DatabaseName := 'DBDemos';
ITable.TableName := 'customer.db'; IDataSource :=
TDataSource.Create (Self); IDataSource.Name :=
'dsCustomer'; IDataSource.DataSet := ITable;
```

```
IDataPipeline := TppBDEPipeline.Create.
    (Auto);
IDataPipeline.Name := 'plCustomer';
IDataPipeline.DataSource := IDataSource;
```

Esses são componentes de acesso a dados Delphi padrão. Não é necessário atribuir a propriedade Name a esses componentes; fizemos isso para lhe dar uma ideia de como eles seriam nomeados se fossem criados dentro do IDE Delphi.

4 Crie o relatório.

```
IReport: = TppReport.Create (Self);
IReport.DataPipeline: = IDataPipeline;
```

A atribuição do pipeline de dados é vital aqui. Sem um pipeline de dados atribuído, esse relatório geraria um número infinito de páginas.

5 Crie as faixas de relatório.

```
IReport.CreateDefaultBands;
```

Este método cria uma banda de cabeçalho, uma banda de detalhes e uma banda de rodapé. Também é possível criar as bandas individualmente e atribuí-las ao relatório. Um relatório deve sempre ter uma faixa de detalhes para ser gerado corretamente.

6 Adicione rótulos à banda do cabeçalho.

```
ILabel1: = TppLabel.Create (Self); ILabel1.Band:
= IReport.HeaderBand; ILabel1.spLeft: = 2;
```

```
ILabel1.spTop: = 2;
ILabel1.Caption: = 'Cliente No.'; ILabel2: =
TppLabel.Create (Self); ILabel2.Band: =
IReport.HeaderBand; ILabel2.spLeft: =
ILabel1.spLeft + ILabel1.spWidth + 3;
```

```
ILabel2.spTop: = 2;
ILabel2.Caption: = 'Nome da empresa';
```

Essas duas etiquetas serão impressas no topo de cada página. O 'sp' nas propriedades spLeft e spTop refere-se aos pixels da tela. Todas as propriedades posicionais dos componentes em um relatório são expressas nas unidades do próprio relatório (ou seja, o valor da propriedade Report.Units). Usar a versão de pixel da tela dessas propriedades nos permite dimensionar e posicionar os componentes sem nos preocupar com o valor atual da propriedade Unit.

7 Adicione componentes com reconhecimento de dados à banda de detalhes.

```
IDBText1: = TppDBText.Create (Self); IDBText1.Band:
= IReport.DetailBand; IDBText1.spLeft: =
ILabel1.spLeft; IDBText1.spTop: = ILabel1.spTop;
IDBText1.DataPipeline: = IDataPipeline;
IDBText1.DataField: = 'CustNo';
```

```
IDBText2: = TppDBText.Create (Self); IDBText2.Band:
= IReport.DetailBand; IDBText2.spLeft: =
ILabel2.spLeft; IDBText2.spTop: = ILabel2.spTop;
IDBText2.DataPipeline: = IDataPipeline;
IDBText2.DataField: = 'Empresa';
```

Esses componentes são posicionados na banda de detalhes diretamente abaixo de seus componentes de etiqueta correspondentes.

8 Adicione um número de página e um carimbo de data / hora à faixa de rodapé.

```
ISysVar: = TppSystemVariable.Create (Self);
```

```
ISysVar.Band: = IReport.FooterBand;
ISysVar.VarType: = vtPrintDateTime;
ISysVar.spLeft: = 2;
ISysVar.spTop: = 2;
ISysVar: = TppSystemVariable.Create. (Auto);
```

```
ISysVar.Band: = IReport.FooterBand;
ISysVar.VarType: = vtPageNoDesc;
ISysVar.Alignment: = taRightJustify;
ISysVar.spLeft: =
(IReport.PrinterSetup.PageDef.spPrintableWidth
- ISysVar.spWidth) -2; ISysVar.spTop: = 2;
```

Observe como o segundo componente da variável do sistema é justificado à direita. O padrão da propriedade AutoSize é True para variáveis do sistema, portanto, precisamos apenas definir a propriedade Alignment e a posição do componente. Aqui, definimos spLeft para que o componente fique alinhado com a margem direita da página (menos 2 pixels para espaçamento). Por ser justificado à direita, esse componente se expandirá para a direita ao imprimir. O objeto PrinterSetup.PageDef contém todas as dimensões da página. A propriedade spPrintableWidth contém a largura do papel menos as margens esquerda e direita (em pixels da tela).

9 Visualize o relatório.

```
IReport.Print;
```

A propriedade `DeviceType` de um relatório é padronizada como 'Tela', portanto, chamar `print` aqui faz com que o formulário Visualizar Impressão seja exibido.

10 Liberte o relatório.

```
IReport.Free;
```

A propriedade `ModalPreview` do relatório é padronizada como `True`; portanto, essa linha de código só será acionada depois que o formulário Visualizar impressão for fechado. Quando o relatório for liberado, ele irá liberar automaticamente as bandas e componentes que ele contém.

11 Liberte os componentes de acesso aos dados.

```
ITable.Free;  
IDataSource.Free;  
IDataPipeline.Free;
```

Os componentes de acesso a dados não são liberados pelo relatório e, portanto, devem ser liberados separadamente.

DESIGN

O Report Designer 89

Dialogs 91

Barras de Ferramentas 95

Suporte para arrastar e soltar 105

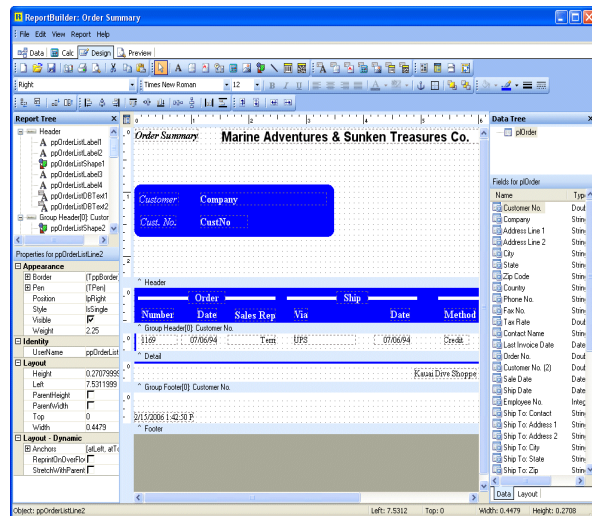
O Assistente de Relatório 107

DESIGN

O Report Designer

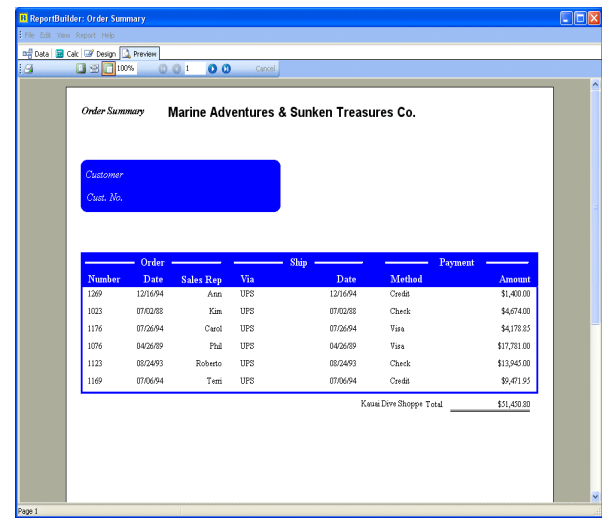
Guia Design

A área de trabalho de design é onde o layout do relatório é criado. Este espaço de trabalho contém todos os menus, barras de ferramentas e diálogos que compõem o Report Designer.



Guia de visualização

A área de trabalho Visualização mostra uma representação do relatório como ele aparecerá quando for impresso na impressora. O processo iterativo de aperfeiçoamento de um relatório geralmente é realizado movendo-se para frente e para trás entre as guias Visualização e Design do Report Designer.



Diálogos

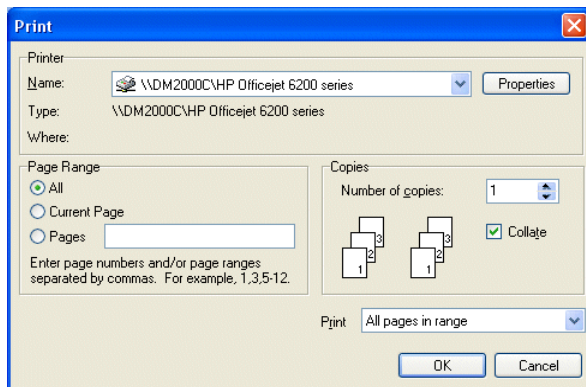
Imprimir Diálogo

A caixa de diálogo Imprimir é exibida automaticamente quando o relatório é enviado para a impressora, permitindo que você selecione as páginas, o número de cópias e a impressora para o relatório. Quando o AllowPrintToFile ou

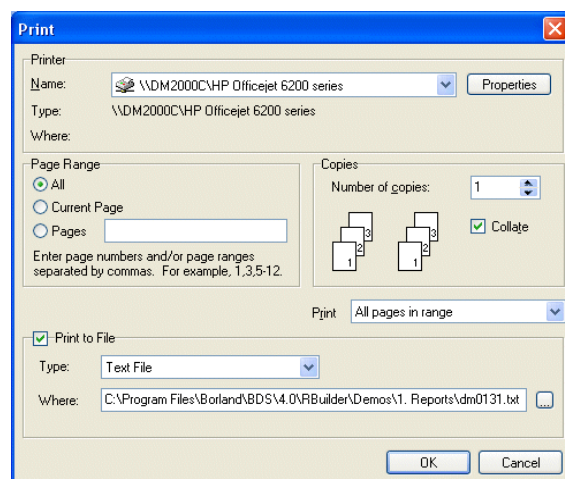
As propriedades AllowPrintToArchive do Relatório são definidas como True, esta caixa de diálogo exibe opções adicionais de impressão em arquivo.

Imprimir Diálogo

1 As configurações do trabalho de impressão podem ser definidas por meio da caixa de diálogo padrão.



2 Caixa de diálogo com as opções de impressão em arquivo.



Diálogo de configuração de página

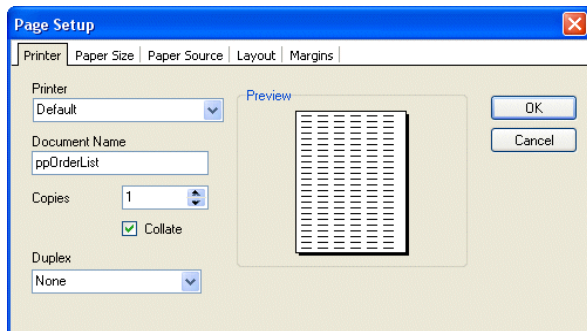
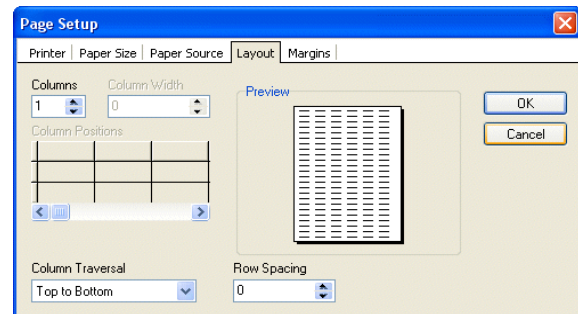
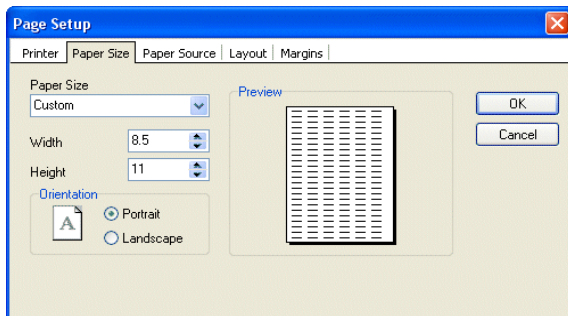
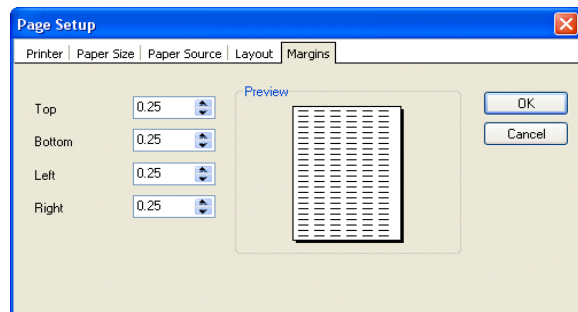
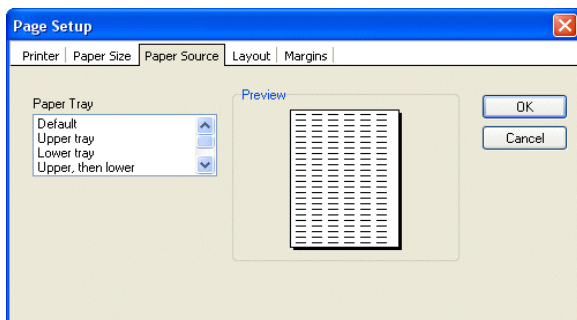
A caixa de diálogo Configurar página pode ser acessada em

Arquivo | Opção de menu Configuração de página do Report

Designer. Você pode definir as seguintes propriedades na caixa

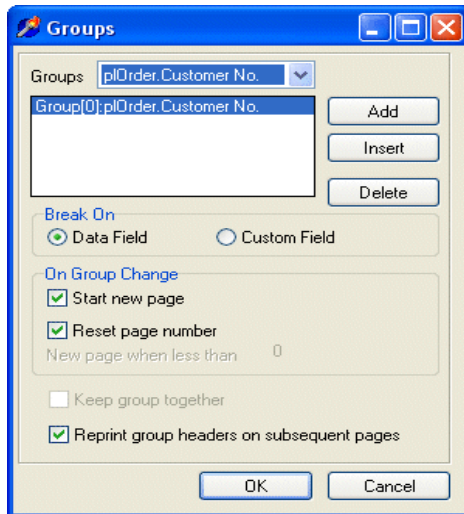
de diálogo Configurar página:

Imprimir Diálogo

1 Impressora**4 Layout para relatórios colunares****2 Tamanho do papel****5 Margens****3 Fonte de papel**

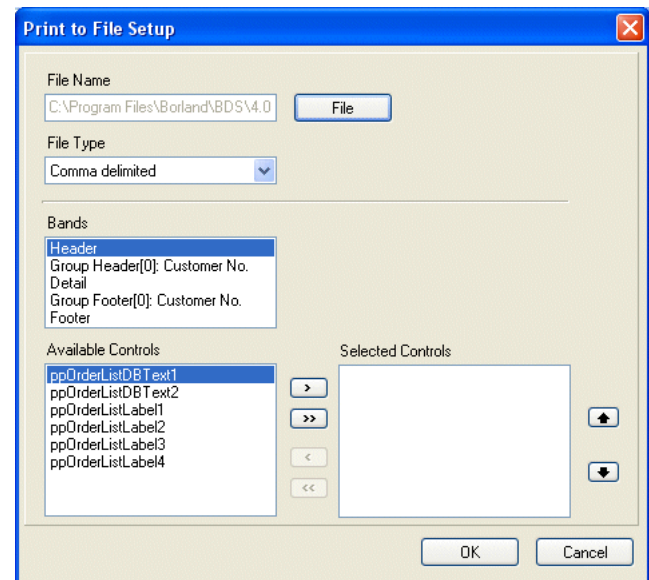
Diálogo de Grupos

A caixa de diálogo Grupos pode ser acessada em Relatório | Opção de menu Grupos do Report Designer. Você pode separar seu relatório em diferentes seções usando grupos. Várias opções estão disponíveis para controlar o comportamento de cada grupo. Por exemplo, você pode querer que cada grupo comece em uma nova página ou reimprima o cabeçalho do grupo quando o grupo continuar em páginas adicionais. Outro recurso poderoso é a opção Manter grupos juntos, que pode ser usada para garantir que todas as informações de um grupo caibam em uma página.



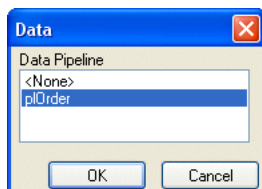
Imprimir para caixa de diálogo de configuração de arquivo

A caixa de diálogo Configurar impressão em arquivo pode ser acessada por meio de Arquivo | Imprimir para a opção de menu Configuração de arquivo do Report Designer. Esta caixa de diálogo é usada para especificar o formato e o conteúdo do arquivo ASCII que será criado se o relatório for impresso em arquivo.



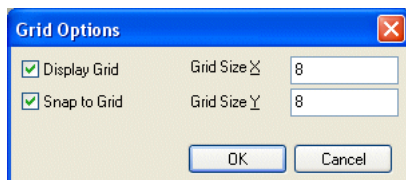
Diálogo de Dados

A caixa de diálogo Dados pode ser acessada em Relatório | Opção de menu de dados do Report Designer. Ele pode ser usado para especificar o pipeline de dados para o relatório.



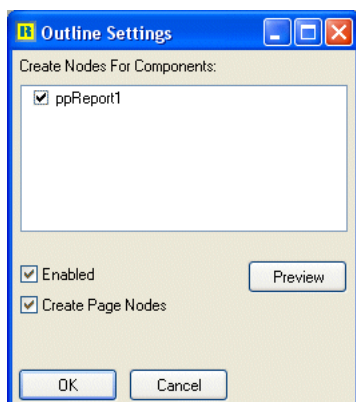
Opções de grade

A caixa de diálogo Grid Options pode ser acessada em Exibir | Menu de opções de grade do Report Designer. Use a caixa de diálogo Opções de grade para controlar como a grade da área de trabalho é desenhada e se os elementos de layout se encaixam automaticamente nos pontos da grade.



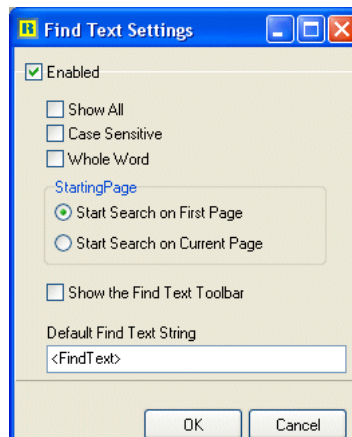
Configurações de esboço

A caixa de diálogo Configurações da estrutura de tópicos pode ser acessada em Relatório | Opção de menu Configurações de estrutura de tópicos do Report Designer. Use esta caixa de diálogo para controlar o comportamento da geração do esboço do relatório. Quando ativado, uma estrutura de árvore de esboço é gerada dinamicamente pelo mecanismo de relatório e renderizada pelo visualizador de relatório.



Encontrar configurações de texto

A caixa de diálogo Localizar configurações de texto pode ser acessada em Relatório | Encontre a opção de menu Configurações de texto do Report Designer. Use esta caixa de diálogo para configurar as opções de localização de texto usadas pelo visualizador de relatório. Quando ativado, o visualizador pode ser usado para localizar e destacar o texto que aparece nas páginas do relatório.



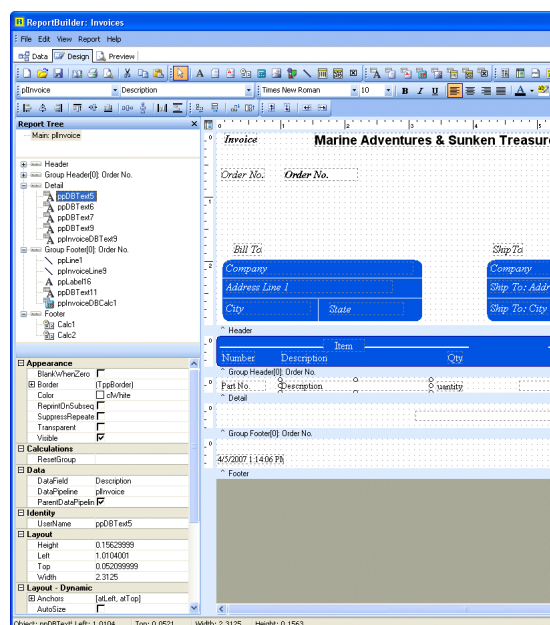
Barras de Ferramentas

Visão geral

As várias barras de ferramentas acessíveis a partir da área de trabalho de design estão documentadas nesta seção. As barras de ferramentas são encaixáveis e seguem o estilo de interface do Office. As barras de ferramentas podem ser acessadas em Exibir | Opção de menu de barras de ferramentas do Report Designer ou clicando com o botão direito na área de encaixe na parte superior do Report Designer.

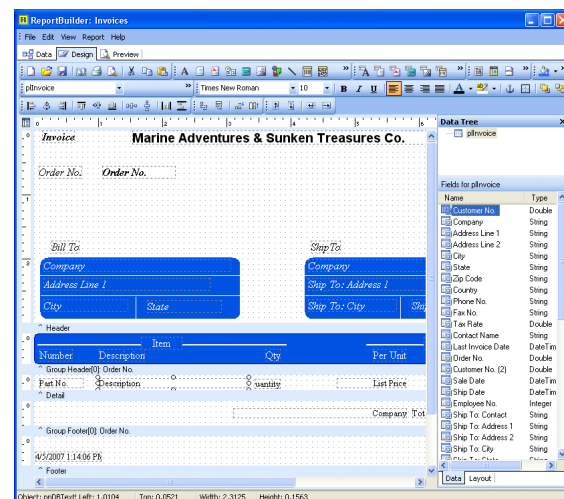
A árvore do relatório

Para acessar esta janela de ferramenta, selecione a opção Exibir | Barras de ferramentas | Opção de menu Árvore de relatórios no menu principal do Designer de relatórios. Esta janela de ferramenta pode ser encaixada nos lados esquerdo e direito do Report Designer. O painel superior contém uma visualização em árvore da estrutura Report.Bands [], Objects [] que compõe o layout do relatório. Os elementos em cada banda são exibidos em ordem z. O painel inferior da Árvore do Relatório contém o Inspetor de Objetos, que pode ser usado para visualizar e modificar as propriedades do objeto selecionado no momento.



A Árvore de Dados

Para acessar esta janela de ferramenta, selecione a opção Exibir | Barras de ferramentas | Opção de menu Árvore de dados no menu principal do Report Designer. Esta janela de ferramenta pode ser encaixada nos lados esquerdo e direito do Report Designer. Ele pode ser usado para criar componentes dentro de qualquer banda. Basta selecionar um conjunto de campos e arrastar a seleção para a banda. Um conjunto de componentes com reconhecimento de dados correspondentes será criado.



Paleta de componentes padrão

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Opção de menu Componentes padrão no menu principal do Report Designer. Esta barra de ferramentas ajudará na criação dos componentes de relatório mais comumente usados.



Rótulo

Exibe texto. Atribua a propriedade Caption para controlar o valor do texto. Você pode fazer com que o rótulo seja redimensionado automaticamente para se ajustar a uma alteração da legenda se definir a propriedade AutoSize como True.

Memorando

Imprime várias linhas de texto simples em um relatório. Para definir o valor, atribua uma lista de strings à propriedade Lines. Para redimensionar dinamicamente o memorando durante a impressão, defina a propriedade Stretch como True. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis.

RichText

Imprime texto formatado. Para definir o valor, atribua a propriedade RichText ou use os métodos LoadFromFile ou LoadFromRTFStream. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis. Em tempo de design, você pode usar o Editor RTF embutido do ReportBuilder para carregar, modificar e salvar dados de rich text armazenados em arquivos.

SystemVariable

Exibe informações de relatório comuns, como número de página, contagem de páginas, data e hora de impressão, data e hora. O tipo de informação exibida é controlado pela propriedade VarType. O formato é controlado pela propriedade DisplayFormat.

Variável

Usado para cálculos por meio de um manipulador de eventos Object Pascal atribuído ao evento OnCalc ou um manipulador de eventos RAP atribuído ao evento OnCalc. Acesse a caixa de diálogo Cálculos (através do menu rápido) ou a guia Calc do Report Designer para codificar um cálculo RAP para este componente.

Imagem

Exibe bitmaps e meta-arquivos do Windows em relatórios. Atribua a propriedade Picture deste componente para colocar uma imagem em seu relatório. Use a caixa de diálogo de imagem integrada do Report Designer para carregar imagens em tempo de design.

Linha

Exibe linhas simples e duplas (verticais ou horizontais). Defina a propriedade Estilo para controlar se a linha é simples ou dupla. Defina a propriedade Peso para controlar a espessura da linha em pontos. Defina a propriedade Posição para controlar se a linha é vertical ou horizontal.

Forma

Imprime várias formas (quadrados, retângulos, círculos, elipses). Defina a propriedade Forma para selecionar um tipo de forma. Use as propriedades Pincel e Caneta para controlar a cor e a borda, respectivamente.

TeeChart

Exibe TeeCharts padrão (sem reconhecimento de dados). Este componente permite que você use TeeCharts dentro do Report Designer. Você pode acessar o editor TeeChart por meio de um menu pop-up.

BarCode

Processa códigos de barras. O valor da string atribuído à propriedade Data é codificado com base no BarCodeType. Se os dados a serem codificados estiverem em um banco de dados, use DBBarCode. As seguintes simbologias são suportadas: Codabar, Código 128, Código 39, EAN-13, EAN-8, FIM A, B, C, Intercalado 2 de 5, PostNet, UPC-A, UPC-E.

2DBarCode

Renderiza códigos de barras 2D. O valor da string atribuído à propriedade Data é codificado com base no BarCodetype. As seguintes simbologias são suportadas: PDF417, Maxicode.

CheckBox

Exibe uma caixa de seleção usando a fonte WingDings.

Paleta de componentes de dados

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Opção de menu Componentes de dados no menu principal do Report Designer. Esta barra de ferramentas ajudará na criação de componentes de relatório com reconhecimento de dados.



DBText

Exibe valores de todos os tipos de campos do banco de dados. Use a propriedade DisplayFormat para formatar o valor.

DBMemo

Imprime texto simples de um campo de memorando de uma tabela de banco de dados. Este controle irá quebrar automaticamente o texto. Defina a propriedade Stretch como True e o componente será redimensionado dinamicamente para imprimir todo o texto. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis.

DBRichText

Imprime texto formatado de um memorando ou campo BLOB de uma tabela de banco de dados. Este controle irá quebrar automaticamente o texto. Defina a propriedade Stretch como True e o componente será redimensionado dinamicamente para imprimir todo o texto. Use a propriedade ShiftRelativeTo para definir relacionamentos dinâmicos com outros objetos extensíveis.

DBCalc

Usado para cálculos de banco de dados simples (soma, mínimo, máximo, contagem e média). O valor pode ser redefinido quando um grupo é interrompido usando a propriedade ResetGroup.

DBImage

Imprime bitmaps ou metarquivos do Windows, que são armazenados em um campo BLOB do banco de dados.

DBBarCode

Renderiza códigos de barras com base no BarCodeType e no valor fornecido por meio da propriedade DataField. As seguintes simbologias são suportadas: Codabar, Código 128, Código 39, EAN-13, EAN-8, FIM A, B, C, Intercalado 2 de 5, PostNet, UPC-A, UPC-E.

2DDBBarCode

Renderiza códigos de barras 2D com base no BarCodetype e no valor fornecido por meio da propriedade DataField. As seguintes simbologias são suportadas: PDF417, Maxicode.

DBTeeChart

Permite que TeeCharts com reconhecimento de dados sejam colocados em um relatório.

DBCheckBox

Exibe uma caixa de seleção com base no valor do campo especificado na propriedade DataField. Este componente pode ser usado com um campo booleano (ou qualquer outro tipo de campo por meio das propriedades BooleanTrue, BooleanFalse).

Paleta de componentes avançados

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Opção de menu Componentes avançados no menu principal do Report Designer. Esta barra de ferramentas ajudará na criação de componentes de relatório avançados.



Região

Agrupa componentes de forma lógica. Use a propriedade ShiftRelativeTo para mover a região em relação a outro componente de redimensionamento dinâmico (como Memo, RichText ou SubReport do tipo filho).

SubRelatório

Lida com vários detalhes principais, cria efeitos de relatório lado a lado e conecta os relatórios como um só. Se você precisar que um relatório seja impresso dentro do contexto de uma banda, use um sub-relatório do tipo filho. Se você precisar conectar relatórios, use um sub-relatório de tipo de seção. A propriedade PrintBehavior determina o tipo de sub-relatório.

Quebra de página

Força uma nova página durante a geração do relatório. Colocar um PageBreak em um relatório fará com que todos os objetos criados após o PageBreak (Ordem Z) sejam movidos para a próxima página. Para forçar uma nova página após a impressão de um componente extensível, atribua a propriedade TppPageBreak.ShiftRelativeTo para fazer referência ao componente Expandível.

CrossTab

Permite gerar um conjunto de cálculos que resume os dados de uma tabela de banco de dados. Ele exibe os cálculos em formato de grade.

Barra de ferramentas padrão

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Opção de menu padrão do menu principal do Report Designer. Esta barra de ferramentas ajudará a salvar o layout do relatório, acessando as opções de impressão e visualização de impressão e acessando as operações de recortar e colar.



Novo Relatório

Cria um layout de relatório em branco.

Relatório Aberto

Exibe a caixa de diálogo Abrir, permitindo que você abra um layout de relatório existente.

Salvar o relatório

Salva um layout de relatório em arquivo.

Configurações da página

Exibe a caixa de diálogo Configurar página, permitindo definir o tamanho do papel e configurar o layout do relatório.

Impressão

Exibe a caixa de diálogo Imprimir antes de enviar o relatório para a impressora.

Antevizão da Impressão

Exibe a janela de visualização de impressão.

Cortar

Corta os componentes selecionados atualmente na área de transferência.

cópia de

Copia os componentes atualmente selecionados para a área de transferência.

Colar

Cola os componentes da área de transferência no relatório.

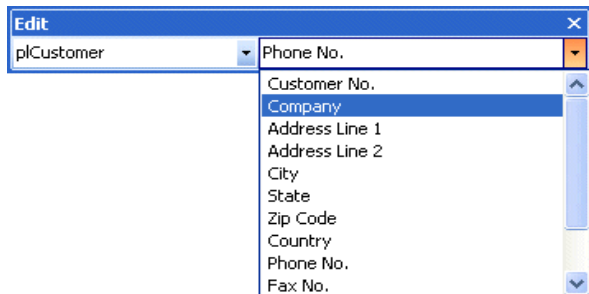
Editar barra de ferramentas

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Editar opção de menu no menu principal do Report Designer. Esta barra de ferramentas ajudará a definir a propriedade ou propriedades mais importantes para o componente selecionado no momento.

1 Nenhum componente selecionado



2 Componente com reconhecimento de dados selecionado



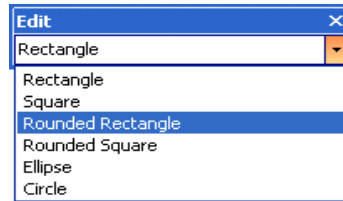
Esta configuração permite que o pipeline de dados e o campo de dados do componente sejam definidos. A lista suspensa à esquerda mostra o pipeline de dados. A lista suspensa à direita mostra o nome do campo.

3 Componente de etiqueta selecionado



Aqui, um componente de rótulo foi selecionado no Report Designer. A barra de ferramentas Editar exibe uma caixa de edição a partir da qual a legenda do rótulo pode ser definida.

4 Componente de forma selecionado



Aqui, um componente de forma foi selecionado no Report Designer. A barra de ferramentas Editar exibe os diferentes tipos de forma.

5 Componente de linha selecionado



Esta configuração permite que você mova a linha para cima, para baixo, para a esquerda ou para a direita nas alças de seleção da linha.

Barra de ferramentas de formatação

Para acessar esta barra de ferramentas, selecione a opção Exibir |

Barras de ferramentas | Opção de menu Formatar no menu principal do

Report Designer. Esta barra de ferramentas ajudará na configuração da

fonte e das cores. Também ajudará na estratificação dos componentes.

**Nome da fonte**

Selecione o nome da fonte para componentes textuais. Usar

Fontes TrueType (indicadas por um ) quando pos-

sível. Eles são reproduzidos bem na tela e na impressora. Se você

estiver usando uma impressora matricial, o driver de impressão

pode fornecer fontes de impressora (indicadas por

ícone ) que você pode usar para acelerar a impressão do

relatório. Finalmente, as fontes que não têm ícone à

esquerda do nome da fonte são fontes de tela e não

devem ser usadas em relatórios onde WYSIWYG é

necessário.

Tamanho da fonte

Selecione o tamanho da fonte. Você também pode digitar nesta caixa para

definir o tamanho da fonte exatamente.

** Negrito**

Define a fonte em negrito.

** itálico**

Define a fonte para itálico.

** Sublinhado**

Define a fonte para sublinhado.

** Justificar à Esquerda**

Justifica à esquerda o texto no componente.

** Centro**

Centraliza o texto no componente.

** Justificar à Direita**

Justifica à direita o texto no componente.

** Cor da fonte**

Define a cor da fonte.

** Cor de destaque**

Define a cor de fundo do componente textual.

** Âncoras**

Especifica como um componente de relatório é ancorado a seu pai. Use Âncoras para garantir que um elemento de relatório mantenha sua posição atual em relação a uma borda de seu controle pai (ou seja, Banda / Região), mesmo se o pai for redimensionado.

** Fronteiras**

Especifica quais das linhas de borda externas de um componente de relatório são renderizadas.

** Traga para frente**

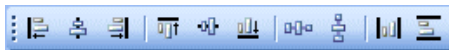
Traz o componente para a frente. Os componentes da frente são impressos por último e os componentes do verso são impressos primeiro. Use a árvore de relatório para ver a estratificação exata dos componentes dentro da banda.

** Enviar para trás**

Envia o componente para trás. Os componentes da frente são impressos por último e os componentes do verso são impressos primeiro. Use a árvore de relatório para ver a estratificação exata dos componentes dentro da banda.

Alinhar ou Barra de Ferramentas de Espaço

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Opção de menu Alinhar ou Espaço no menu principal do Report Designer. Esta barra de ferramentas ajudará a posicionar os componentes uns em relação aos outros e em relação à banda em que aparecem.

 **Alinhar Bordas Esquerdas**

Alinha um grupo de componentes com a posição mais à esquerda do componente que foi selecionado primeiro.

 Alinhar ao Meio

Centraliza um grupo de componentes com base no centro horizontal do componente que foi selecionado primeiro.

 Alinhar as bordas direitas

Alinha um grupo de componentes com a posição mais à direita do componente que foi selecionado primeiro.

 Alinhar as bordas superiores

Alinha um grupo de componentes com a posição superior do componente que foi selecionado primeiro.

 Alinhar Centro

Alinha um grupo de componentes com base no centro vertical do componente que foi selecionado primeiro.

 Alinhar Bordas Inferiores

Alinha um grupo de componentes com a posição mais inferior do componente que foi selecionado primeiro.

 Espaço horizontalmente

Distribui um conjunto de componentes com base na posição mais à esquerda do primeiro componente selecionado e na posição mais à direita do último componente selecionado.

 Espaço Verticalmente

Distribui um conjunto de componentes com base na posição superior do primeiro componente selecionado e na posição inferior do último componente selecionado.

 Centralizar horizontalmente na banda

Centraliza um componente horizontalmente dentro de uma banda.

 Centralizar verticalmente na banda

Centraliza um componente verticalmente dentro de uma banda.

Barra de ferramentas de tamanho

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Opção de menu Tamanho no menu principal do Report Designer.

 **Reduzir Largura**

Determina a largura mínima de todos os componentes selecionados e, a seguir, define a largura dos componentes com esse valor.

 GrowWidth

Determina a largura máxima de todos os componentes selecionados e, a seguir, define a largura dos componentes com esse valor.

 Diminuir Altura

Determina a altura mínima de todos os componentes selecionados e, a seguir, define a altura dos componentes com esse valor.

 Aumentar a altura

Determina a altura máxima de todos os componentes selecionados e, a seguir, define a altura dos componentes para esse valor.

Barra de ferramentas Nudge

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Opção de menu Tamanho no menu principal do Report Designer.

 **Empurrar para cima**

Move todos os componentes selecionados um pixel para cima.

 Nudge Down

Move todos os componentes selecionados um pixel para baixo.

 Deslocar para a esquerda

Move todos os componentes selecionados um pixel para a esquerda.

 Deslocar para a direita

Move todos os componentes selecionados um pixel para a direita.

Barra de Ferramentas Desenhar

Para acessar esta barra de ferramentas, selecione a opção Exibir | Barras de ferramentas | Desenhe a opção de menu do menu principal do Report Designer. Esta barra de ferramentas auxiliará na configuração das cores e bordas dos componentes.



Cor de Preenchimento

Apenas para formas, linhas e componentes de região. Define a propriedade Brush.Color. Para definir a cor de um componente textual, marque a ação Destacar Cor da barra de ferramentas Formatar.

Cor da linha

Apenas para formas, linhas e componentes de região. Define a propriedade Pen.Color.

Espessura da linha

Para uso apenas com um componente Line. Define a propriedade Weight.

Estilo de Linha

Para uso apenas com um componente Line. Define a propriedade Pen.Style.

Suporte para arrastar e soltar

Visão geral

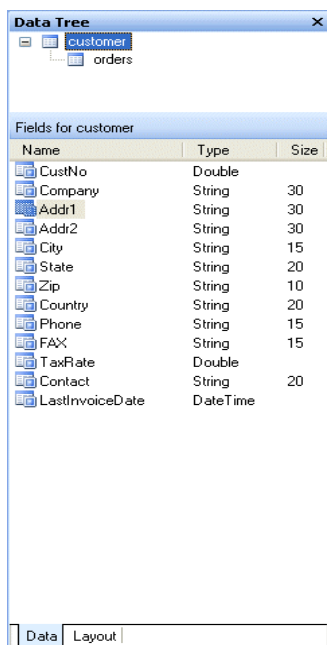
ReportBuilder contém um Assistente de relatório que permite criar rapidamente um layout de relatório inteiro. Isso é ótimo para gerar um relatório inteiro, mas e se você precisar criar apenas uma parte de um relatório complexo? A funcionalidade arrastar e soltar é uma solução ideal para esse problema porque permite criar um conjunto de componentes dentro do contexto de um layout de relatório existente. No ReportBuilder, o suporte para arrastar e soltar é fornecido por meio da Árvore de Dados.

A Árvore de dados possui duas guias. Na visualização em árvore superior, a guia Dados contém uma lista de pipelines de dados aos quais o relatório tem acesso. Na lista inferior

visualização, todos os campos para o pipeline de dados atualmente selecionado são exibidos. Os campos podem ser selecionados na visualização de lista inferior e arrastados para qualquer parte do layout do relatório. O componente com reconhecimento de dados apropriado para o campo fornecido será criado junto com um rótulo e uma borda.

A segunda guia da Árvore de dados é intitulada Layout. Esta guia permite controlar o comportamento de arrastar e soltar. Um rótulo e uma borda são criados para cada componente com reconhecimento de dados por padrão. Você pode desligar o rótulo e a borda, controlar a cor do rótulo ou borda e controlar a fonte do rótulo e do componente com reconhecimento de dados nesta guia. Depois de definir o comportamento de arrastar e soltar, ele será retido para futuras sessões de design.

Abas da árvore de dados



Árvore de dados - guia Dados



Árvore de dados - guia Layout

O Assistente de Relatório

Visão geral

O Report Wizard é uma das muitas partes do ReportBuilder que reflete um nível de profissionalismo e atenção aos detalhes encontrados em nenhum outro relatório. Você será capaz de reconhecer e usar rapidamente o ReportBuilder Report Wizard.

O Assistente de Relatório pode ser acessado em Arquivo | Nova opção de menu do Report Designer. Uma série de telas é apresentada, cada uma solicitando informações sobre o relatório. Quando a última página é

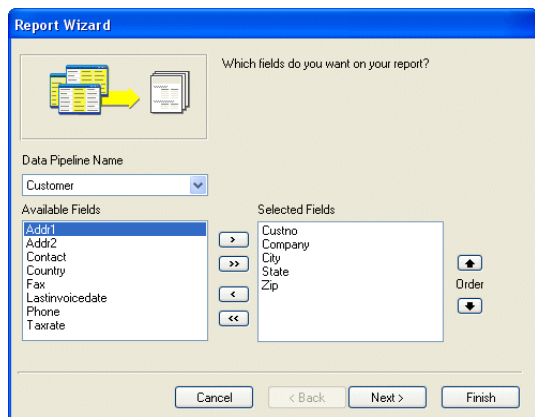
alcançado, uma pré-visualização ou opção de design pode ser selecionada. Quando o botão 'Concluir' é clicado, ele faz com que um relatório seja criado e exibido conforme solicitado.

Assistente de relatório: crie um relatório simples

As capturas de tela a seguir percorrem a criação de um relatório simples por meio do Report Wizard.

Crie um relatório simples

1 Selecione os campos.



2 Pule a página de grupos.



Assistente de Relatório: Criar um Relatório Simples - cont.

Crie um relatório simples - cont.

3 Selezione o layout.

Report Wizard

How would you like to lay out your report?

Customers				
COMPANY	REGION	CITY	<u>STATE</u>	
ActionClub	Southeast	Sarasota	FL	
ActionClub	Southeast	Tampa	FL	
ActionClub	Southeast	San Jose	CA	
ActionClub	Southeast	Phoenix	AZ	
ActionClub	South	Dallas	TX	
ActionClub	South	Atlanta	GA	
ActionClub	South	El Paso	TX	
ActionDriver	Southeast	Charlotte	NC	
ActionDriver	Southeast	Miami	FL	
ActionDriver	Southeast	Columbia	SC	
ActionDriver	Southeast	Louisiana	LA	

Layout

☐ Vertical

☒ Tabular

Oriention

☒ Portrait

☐ Landscape



☒ Adjust field widths so all fields fit on page.

Cancel
< Back
Next >
Finish

5 Seleccione Design ou Visualizar e Concluir.

Report Wizard



That is all the information needed to create your report

Do you want to preview the report or modify the report's design?

☒ Preview the report

☐ Modify the report's design

Cancel

< Back

Next >

Finish

4 Selezione o estilo.

Report Wizard

What style would you like?

Customers

COMPANY	REGION	CITY	STATE
Action Club	Southwest	Sedona	FL
Action Club	Southwest	Tucson	AZ
Action Club	Southwest	Flag	AZ

Customers

Company

Action Club

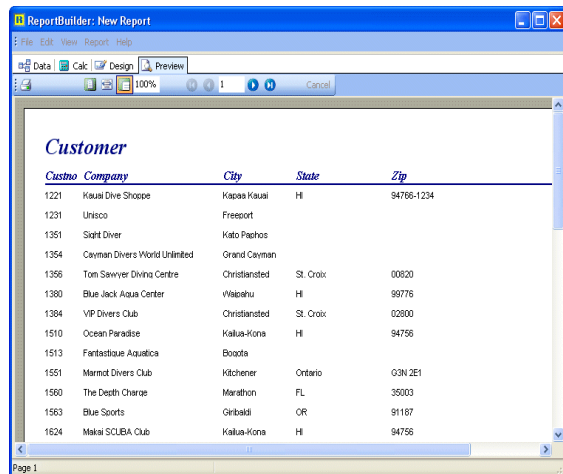
- Bold
- Casual
- Compact
- Compact**
- Formal
- Soft Gray

Cancel < Back Next > Finish

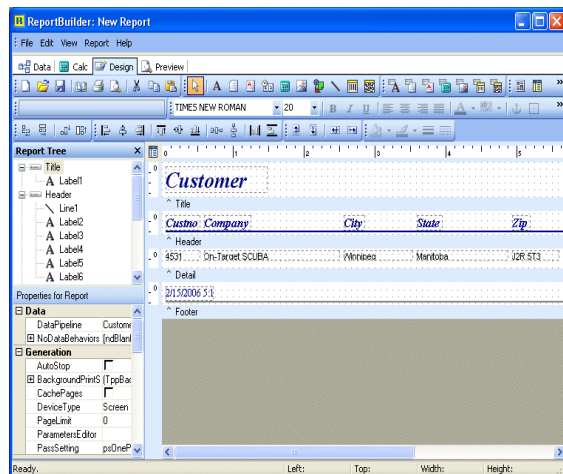
Assistente de Relatório: Criar um Relatório Simples - cont.

Crie um relatório simples - cont.

Relatório como aparece na área de trabalho de visualização

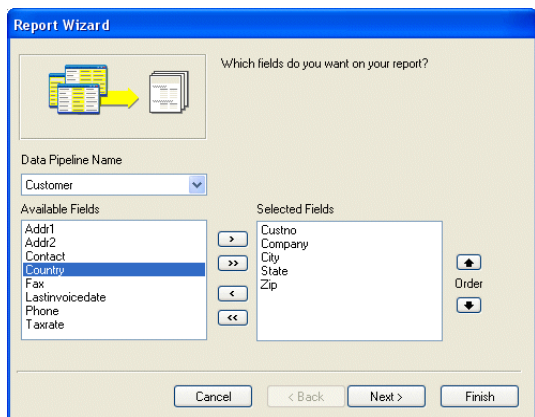


Relatório como aparece na área de trabalho do Design



Assistente de relatório: criar um relatório baseado em grupo

As capturas de tela a seguir percorrem a criação de um relatório baseado em grupo por meio do Assistente de Relatório.

Crie um relatório baseado em grupo**1 Selecione os campos.**

Report Wizard

Which fields do you want on your report?

Data Pipeline Name: Customer

Available Fields:

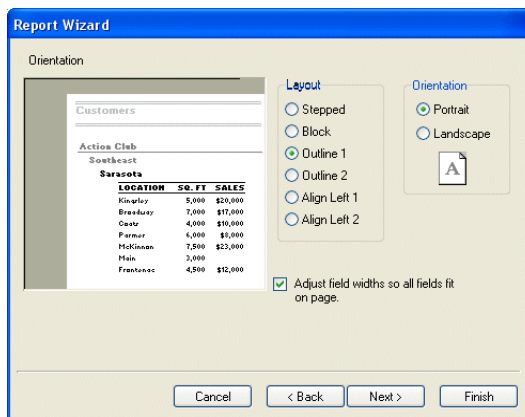
- Addr1
- Addr2
- Contact
- Country
- Fax
- LastInvoiceDate
- Phone
- TaxRate

Selected Fields:

- Custno
- Company
- City
- State
- Zip

Order:

Cancel < Back Next > Finish

3 Selecione o layout.

Report Wizard

Orientation:

Customers

Action Club

Southwest

Sarasota

LOCATION	SQ. FT.	SALES
Kingsley	5,000	\$20,000
Bradbury	7,000	\$17,000
Cesar	4,000	\$10,000
Parker	6,000	\$15,000
McKinney	7,500	\$23,000
Main	3,000	
Franklin	4,500	\$12,000

Layout:

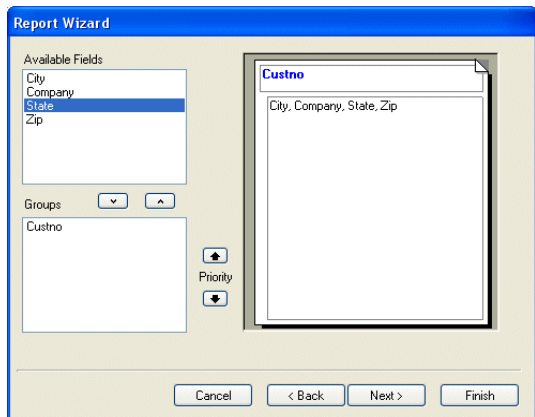
- Stepped
- Block
- Outline 1
- Outline 2
- Align Left 1
- Align Left 2

Orientation:

- Portrait
- Landscape

Adjust field widths so all fields fit on page.

Cancel < Back Next > Finish

2 Selecione um campo de grupo.

Report Wizard

Available Fields:

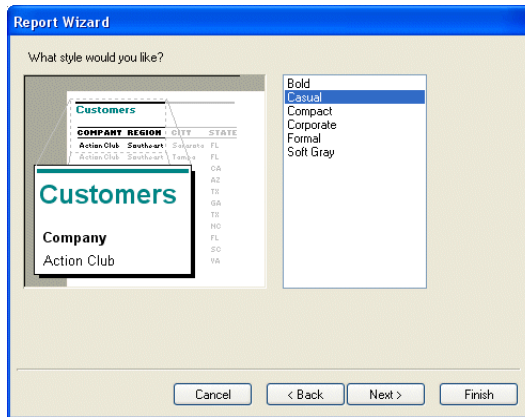
- City
- Company
- State
- Zip

Groups:

- Custno

Priority:

Cancel < Back Next > Finish

4 Selecione o estilo.

Report Wizard

What style would you like?

Customers

COMPANY REGION CITY STATE

Action Club Southwest Sarasota FL

Customers

Company

Action Club

Style:

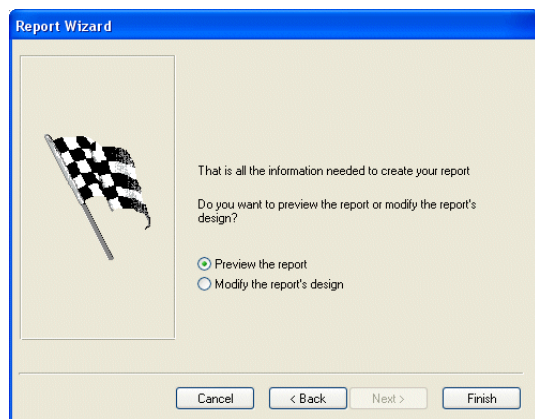
- Bold
- Normal
- Compact
- Corporate
- Formal
- Soft Gray

Cancel < Back Next > Finish

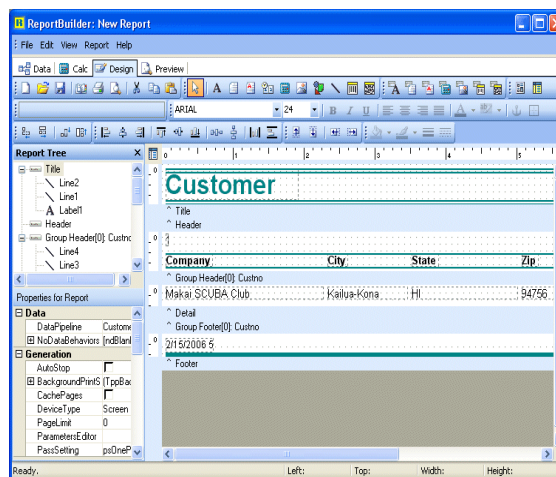
Assistente de relatório: Criar um relatório baseado em grupo - cont.

Criar um relatório baseado em grupo (cont.)

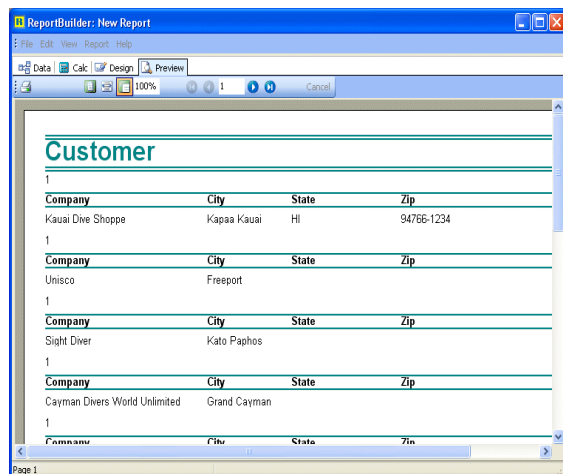
5 Selecione Design ou Visualizar e Concluir.



Relatório como aparece na área de trabalho de design.



Relatório como aparece na área de trabalho de Visualização.



PRINT

Introdução 115

Visualizando 117

PDF de configurações personalizadas da 119

impressora 121

Arquivo de Relatório 123

Imprimir em arquivo de texto ASCII 125

Arquivo de texto de emulação de

relatório 127 RTF, HTML e outros 129

formatos Relatórios de e-mail 131

PRINT

Introdução

Geração

A saída do relatório é gerada principalmente por meio de uma chamada ao método Print do objeto de relatório.

1 Imprimir na tela

O código a seguir faria com que o formulário de visualização de impressão fosse exibido e a primeira página do relatório a ser apresentada neste formulário:

usa

ppTypes;

```
ppReport1.DeviceType: = dtScreen;
ppReport1.Print;
```

2 Imprimir para impressora

Este código faria com que a caixa de diálogo Imprimir fosse exibida. O relatório seria então enviado para a impressora:

usa

ppTypes;

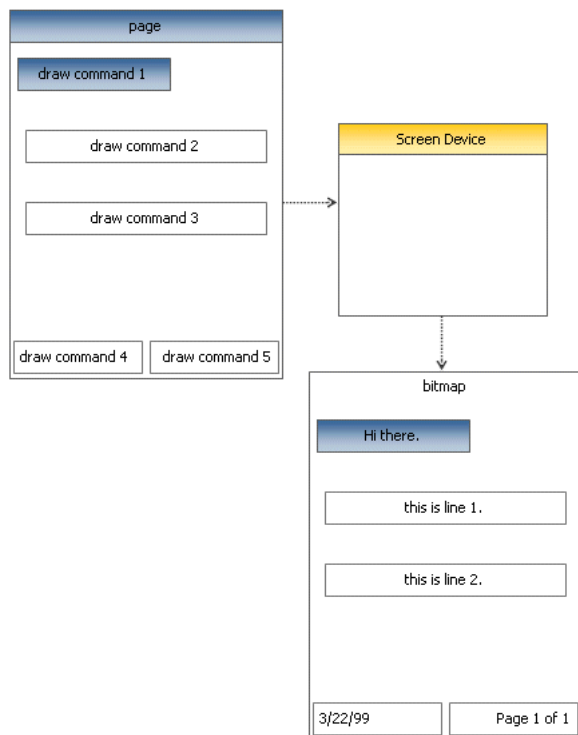
```
ppReport1.DeviceType: = dtPrinter;
ppReport1.Print;
```

Como você pode ver, é muito fácil gerar um relatório usando o ReportBuilder. Para usar alguns dos recursos mais avançados relacionados à saída do relatório, você precisará entender um pouco mais sobre o que está acontecendo nos bastidores.

Como você já deve saber, a saída do relatório é um dos elementos da equação do relatório. A equação de relatório foi descrita na introdução do ReportBuilder e é a seguinte:



A saída de relatório nativo no ReportBuilder assume a forma de objetos de página. Os objetos de página contêm comandos de desenho, que descrevem o que a página contém. Os objetos de página são enviados aos dispositivos, que os convertem em um formato apropriado para o dispositivo. O diagrama a seguir mostra objetos de página que são enviados para o ScreenDevice e, em seguida, convertidos em um bitmap para exibição no formulário Visualizar impressão.

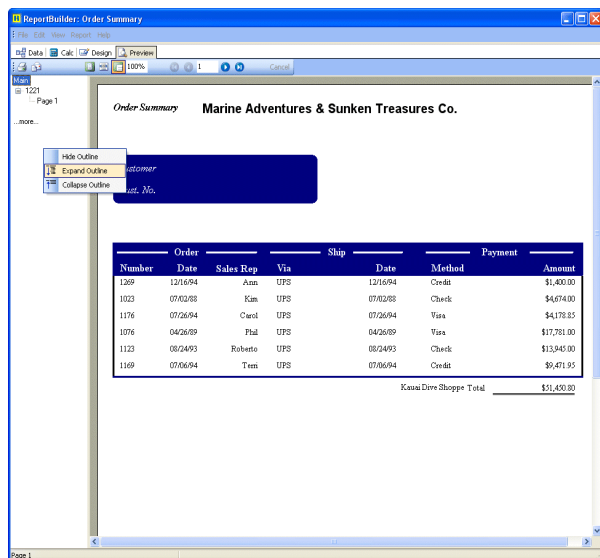


Várias classes de dispositivos são fornecidas como padrão com o ReportBuilder. Cada classe de dispositivo oferece suporte a um dispositivo físico ou formato de arquivo diferente, mas a finalidade do dispositivo permanece a mesma: converter a saída de relatório nativo do ReportBuilder para o formato apropriado para o dispositivo fornecido. O restante desta seção descreve os diferentes dispositivos e seus recursos.

Visualizando

Antevisão da Impressão

A forma de saída de relatório mais conhecida e frequentemente usada está contida no formulário Visualizar impressão. O formulário Print Preview é iniciado automaticamente quando o método Print do componente de relatório é chamado (e a propriedade DeviceType foi configurada para 'Screen'). O formulário de visualização de impressão é mostrado abaixo.



O formulário de visualização permite que o usuário controle e visualize a saída do relatório. Um esboço de nós de página é exibido no lado esquerdo do formulário, adjacente ao visualizador de página. A barra de ferramentas na parte superior do formulário de visualização pode ser usada para realizar as seguintes atividades.

Impressão

Use o botão Imprimir para enviar a saída do relatório para a impressora ou para um arquivo de saída, como PDF.

E-mail

Use o botão E-mail para criar uma mensagem de e-mail que contenha um anexo em PDF da saída do relatório. Pressionar o botão E-mail inicia automaticamente seu cliente de e-mail padrão com o relatório anexado como um documento PDF. O Report.EmailSettings é usado para configurar se esta opção está disponível.

Encontrar Texto

Use este botão para exibir a barra de ferramentas Localizar texto que pode ser usada para pesquisar e destacar palavras que aparecem no relatório.

Controles de zoom

Use os controles de zoom para dimensionar o tamanho da página renderizada para o visualizador.



Controles de navegação

Use os controles de navegação para especificar qual página é exibida para o visualizador. Opcionalmente, você pode clicar nos nós da página Esboço para navegar para uma página.



Embora o formulário de visualização de impressão fornecido com o ReportBuilder tenha a aparência e o desempenho de uma interface de usuário verdadeiramente profissional, você pode querer personalizá-lo para emular a aparência do resto do seu aplicativo. Essa personalização pode ser feita criando um formulário especial que desce da classe CustomPreviewer e registrando-o como o formulário de visualização de impressão 'oficial'. Esse processo é totalmente descrito no tutorial 'Construindo um aplicativo de relatório'.

Configurações de impressão personalizadas

Visão geral

ReportBuilder contém uma variedade de propriedades que permitem controlar vários aspectos do trabalho de impressão.

As propriedades a seguir estão contidas na propriedade PrinterSetup do objeto Relatório e podem ser configuradas por meio do Inspetor de Objetos ou do Arquivo | Caixa de diálogo Configuração de página do Report Designer.

BinName

O nome da bandeja (bandeja de papel) que contém o papel no qual o relatório será impresso. Um uso comum dessa propriedade é configurá-la como Alimentação manual em relatórios de etiqueta de endereçamento. Dessa forma, você pode fornecer um meio de carregar o papel de etiqueta especial sem configurar manualmente a impressora.

Collation

Ao imprimir várias cópias de um relatório, esta propriedade determina se cada cópia é impressa como um conjunto separado de páginas.

Cópias

O número de cópias do relatório que devem ser impressas.

Nome do Documento

O nome do trabalho de impressão a ser usado quando o relatório for impresso - este é o nome como aparecerá no Gerenciador de Impressão.

Duplex

Para impressoras que suportam duplex, esta configuração determina o tipo de impressão duplex que ocorrerá.

Margens

As propriedades da margem determinam as margens esquerda, superior, direita e inferior do relatório.

Orientação

Paisagem ou retrato.

PaperHeight e PaperWidth

As dimensões do papel selecionado, com base no valor da propriedade PaperName. Se esses valores forem inseridos manualmente e nenhum tamanho de papel correspondente for encontrado, o PaperName será definido como 'Personalizado'.

PaperName

O nome do tipo de papel, como 'Carta' ou 'A4'.

Nome da impressora

O nome da impressora na qual o relatório deve ser impresso. Esta propriedade é padronizada como 'Padrão', o que fará com que o ReportBuilder use a impressora padrão declarada em seu sistema.

Como todos os relatórios ReportBuilder contêm um objeto PrinterSetup, é possível usar sub-relatórios do tipo seção para atingir alguns níveis bastante incríveis de personalização do comportamento de impressão. Por exemplo, diferentes páginas de um relatório podem ser impressas de diferentes escaninhos da mesma impressora ou totalmente em impressoras diferentes. Alguns desses recursos são apresentados nos exemplos de configuração da impressora fornecidos com o ReportBuilder (exemplos 121-124).

PDF

Imprimir em PDF

O Portable Document Format (PDF) da Adobe é um formato de documento padrão usado em todo o mundo. Os documentos PDF podem ser visualizados usando o Adobe Acrobat Reader, que pode ser baixado gratuitamente.

ReportBuilder inclui a capacidade de gerar saída de relatório como um documento PDF. Use as propriedades Report.PDFSettings para configurar as opções de PDF. Quando PDFSettings.OpenPDFFile é definido como True, o arquivo PDF gerado será aberto automaticamente no Acrobat Reader.

Imprimindo diretamente para arquivo PDF

Você pode imprimir um relatório diretamente em arquivo PDF com o seguinte código:

```
usa
    ppTypes;

ppReport1.ShowPrintDialog: = False;
ppReport1.DeviceType: = dtPDF;
ppReport1.PDFSettings.OpenPDFFile: = True;

ppReport1.TextFileName: = 'c: \
myReport.pdf';
ppReport1.Print;
```

Permitir que o usuário final imprima em PDF

O código a seguir daria ao usuário final a opção de imprimir em PDF a partir da caixa de diálogo de impressão.

```
usa
    ppTypes;

ppReport1.AllowPrintToFile: = True;
ppReport1.ShowPrintDialog: = True;
ppReport1.DeviceType: = dtPrinter
ppReport1.PDFSettings.OpenPDFFile: = True;

ppReport1.Print;
```

Imprimindo diretamente para fluxo de PDF

É possível gerar um PDF diretamente em um fluxo, em vez de criar um arquivo. Para obter um exemplo, consulte a demonstração 109 (dm0109.pas) no Projeto RBuilder \ Demos \ Report \ Demo.dpr.

Arquivamento de relatórios

Visão geral

O arquivamento é um processo pelo qual uma instância de saída de relatório é salva para uso futuro. Esse recurso é mais frequentemente usado para salvar um instantâneo de um relatório conforme ele apareceu em um determinado dia e em um determinado horário. Se os dados que foram usados para gerar o relatório forem alterados, uma representação do relatório como ele apareceu naquele momento ainda estará disponível. O arquivamento também pode ser usado para gerar relatórios diários ou mensais fora do horário de pico e, em seguida, permitir que os usuários visualizem ou imprimam os relatórios posteriormente.

1 Arquivando um Relatório

UMA Imprima diretamente para o arquivo.

Você pode imprimir um relatório diretamente para arquivar com o seguinte código:

```
usa
    ppTypes;

ppReport1.ShowPrintDialog: = False;
ppReport1.DeviceType: = dtArchive;
ppReport1.ArchiveFileName: =
    'c: \ test.raf';
ppReport1.Print;
```

Nesse código, a caixa de diálogo Imprimir está oculta; o tipo de dispositivo é definido como 'Arquivo'; o nome do arquivo é especificado e a saída do relatório é gerada. Os arquivos compactados geralmente recebem a extensão 'raf', que significa Arquivo de Arquivo de Relatório.

B Permitir que os usuários finais imprimam para arquivamento.

Para dar ao usuário final a oportunidade de imprimir o relatório para arquivar sempre que o método Print for chamado, use o seguinte código:

```
usa
    ppTypes;

ppReport1.AllowPrintToArchive: = True;
ppReport1.ShowPrintDialog: = True;
ppReport1.DeviceType: = dtPrinter;
ppReport1.ArchiveFileName: = 'c: \ test.raf';
ppReport1.Print;
```

Quando o método de impressão é chamado, a caixa de diálogo Imprimir será exibida. Como AllowPrintToArchive foi definido como True, a caixa de diálogo Imprimir conterá duas guias do bloco de notas: 'Impressora' e 'Arquivo'. Se o usuário final clicar na guia 'Arquivo', o nome do arquivo compactado será padronizado.

2 Usando o ArchiveReader

Após a criação dos arquivos compactados, eles podem ser lidos pelo componente ArchiveReader. O componente ArchiveReader possui um conjunto de recursos semelhantes ao componente Report. Na verdade, esses dois componentes descendem do mesmo ancestral comum. Você pode imprimir o conteúdo de um arquivo em uma janela de visualização de impressão padrão com o seguinte código:

```
usa
    ppTypes;

ppArchiveReader1.ArchiveFileName: = 'c: \
test.raf';
ppArchiveReader1.DeviceType: = dtScreen;
ppArchiveReader1.Print;
```

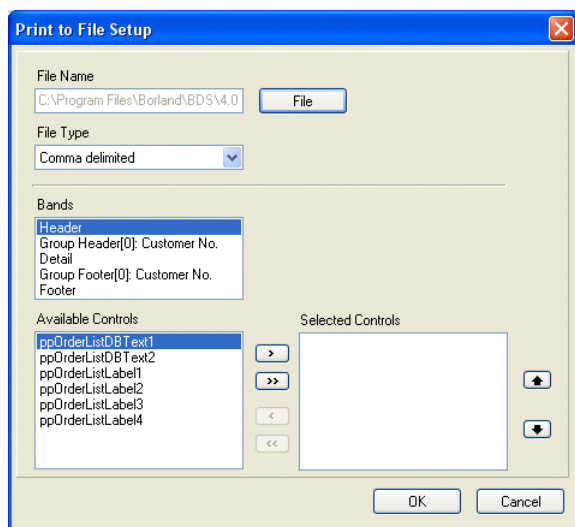

Imprimir em texto ASCII

Visão geral

O dispositivo TextFile no ReportBuilder converte a saída do relatório em um arquivo de texto ASCII. A intenção deste dispositivo é criar dados estruturados que podem ser importados para outro aplicativo, como uma planilha. Os seguintes formatos de arquivo de texto são suportados:

- delimitado por vírgulas
- delimitado por tabulação
- registros de comprimento fixo
- delimitado de forma personalizada

Como a saída do relatório é essencialmente de formato livre e pode conter muitos componentes não textuais, o ReportBuilder fornece uma maneira de controlar rigidamente o formato do arquivo de texto gerado por meio deste dispositivo. Esse controle é fornecido pela caixa de diálogo Configuração de impressão para arquivo (acessível na opção de menu Arquivo | Configuração de impressão para arquivo do Report Designer). Esta caixa de diálogo é mostrada abaixo:



Esta caixa de diálogo permite definir o nome do arquivo padrão, selecionar o formato do arquivo e especificar os componentes textuais que serão salvos no arquivo e a ordem em que serão salvos. Ao utilizar o dispositivo TextFile, é obrigatório que essas configurações sejam concluídas. Se nenhum componente tiver sido selecionado, o resultado será um arquivo de texto em branco.

Imprima diretamente para o arquivo de texto ASCII

Depois que um relatório for configurado para saída de arquivo de texto ASCII, o código a seguir irá gerar o relatório diretamente para o arquivo:

usa

ppTypes;

```
ppReport1.ShowPrintDialog: = False;  
ppReport1.DeviceType: = dtTextFile;  
ppReport1.Print;
```

Permitir que o usuário final imprima no arquivo de texto ASCII

O código a seguir dará ao usuário final a opção de imprimir em texto quando o método de impressão for chamado:

usa

ppTypes;

```
ppReport1.AllowPrintToFile: = True;  
ppReport1.DeviceType: = dtPrinter;  
ppReport1.ShowPrintDialog: = True;  
ppReport1.Print;
```

O componente de relatório tem um evento OnSaveText que dispara sempre que um componente textual é salvo no arquivo de texto. Este evento pode ser usado para personalizar a saída gerada para o arquivo de texto. Um exemplo do que pode ser alcançado por meio deste evento é fornecido no exemplo 106

(.. \RBuilder \ Demos \ Reports \ dm0106.pas).

Arquivo de texto de emulação de relatório

Visão geral

O dispositivo TextFile discutido no tópico anterior é usado para saída de arquivo de texto ASCII. Esse tipo de dispositivo é excelente para criar dados portáteis a partir da saída de relatórios. No entanto, a formatação do relatório é perdida em uma vírgula bem definida ou formato de dados delimitado por tabulação. Quando a saída de arquivo de texto que retém o espaçamento do relatório original é necessária, o dispositivo ReportTextFile pode ser usado. Os arquivos de texto gerados com este dispositivo são úteis para enviar como anexos de e-mail ou diretamente para impressoras matriciais para saída de alta velocidade.

O dispositivo ReportTextFile não requer nenhuma configuração especial com antecedência. Todos os componentes textuais (exceto memorandos e richtext) serão impressos no arquivo de texto.

Imprima diretamente para o arquivo de texto de emulação de relatório

O código a seguir envia um relatório para um arquivo de texto de emulação de relatório:

usa

ppTypes;

```
ppReport1.ShowPrintDialog: = False;  
ppReport1.DeviceType: = dtReportTextFile;  
ppReport1.TextFileName: = 'c: \ test.txt';  
ppReport1.Print;
```

Permitir que o usuário final imprima em um arquivo de texto de emulação de relatório

O código a seguir daria ao usuário final a opção de imprimir em um arquivo de texto de emulação de relatório quando o método de impressão for chamado:

usa

ppTypes;

```
ppReport1.AllowPrintToFile: = True;  
ppReport1.TextFileName: = 'c: \ test.txt';  
ppReport1.DeviceType: = dtPrinter;  
ppReport1.ShowPrintDialog: = True; ppReport1.Print;
```

Para que o texto de formato livre da saída do relatório seja mapeado corretamente para a grade de caracteres de um arquivo de texto, às vezes é necessário ajustar as posições dos componentes no layout do relatório. Um exemplo de um relatório que foi configurado para uso com o dispositivo Report Emulation Text File é fornecido no exemplo 107

(.. \ RBuilder \ Demos \ Reports \ dm0107.pas).

RTF, HTML e outros formatos

Visão geral

Utilizando a arquitetura de dispositivo altamente extensível do ReportBuilder, você pode criar seus próprios descendentes de classe de dispositivo para converter a saída nativa do ReportBuilder em qualquer formato de arquivo. Alguns desenvolvedores já fizeram exatamente isso e estão oferecendo classes de dispositivos para alguns dos formatos de arquivo mais populares. Esses dispositivos adicionais se integram perfeitamente ao ReportBuilder.

TExtraDevices

por James Waler

Fornece suporte de dispositivo de arquivo para RTF, HTML, Excel, Lotus e Quattro Pro. Uma demonstração está disponível para download. Contate James Waler diretamente em www.waler.com para mais detalhes.

PsRBExportDevices

por Pragnaan Software

Fornece suporte de dispositivo de arquivo para PDF, HTML, RTF, Excel, JPEG, GIF, BMP, EMF e WMF. Uma demonstração está disponível para download. Contate Pragnaan Software em www.pragnaan.com para mais detalhes.

IIPDFLib

por Ilionsoft

Fornece suporte de dispositivo de arquivo para o formato PDF.

Gnostice eDoc Engine

por Gnostice Information Technologies

Fornece suporte para dispositivo de arquivo para mais de 20 formatos, incluindo PDF, RTF, XHTML, HTML, Excel, TIFF, SVG, PNG, JPEG, Metafile e outros. Os recursos avançados para o formato PDF incluem compressão, criptografia, fontes TrueType, incorporação / sub-configuração de fontes, caracteres internacionais (europeus, chineses, japoneses, ...), gráficos vetoriais e renderização de metarquivos, renderização de rich text.

Relatórios de e-mail

Visão geral

ReportBuilder inclui suporte integrado para envio de relatórios por e-mail como anexos. Use as propriedades Report.EmailSettings para configurar as opções de e-mail. Por padrão, os relatórios são enviados por e-mail como anexos em PDF.

Habilite a opção de e-mail para o visualizador

Quando Report.EmailSettings.Enabled é definido como True, um botão E-mail aparecerá no visualizador de relatório, logo à direita do botão Imprimir. Pressionar o botão E-mail iniciará o cliente de e-mail padrão do usuário final com um anexo em PDF do relatório. O usuário final pode então especificar o endereço do destinatário, o assunto e a composição da mensagem, assim como o faz para todos os seus e-mails.

```
ppReport1.EmailSettings.Enabled: = True;  
ppReport1.Print;
```

Abra o cliente de e-mail com PDF anexado

```
ppReport1.EmailSettings.PreviewInEmail Cliente: =  
True;  
ppReport1.SendMail;
```

Enviar e-mail sem interação do usuário

```
ppReport1.EmailSettings.PreviewInEmail Cliente: =  
False;  
ppReport1.EmailSettings.Subject: = 'Testar e-mail  
ReportBuilder';  
ppReport1.EmailSettings.Recipients.Text: = '  
somebody@somecompany.com';  
ppReport1.EmailSettings.Body.Text: = 'Ver relatório em  
anexo ...';  
ppReport1.SendMail;
```

Envie e-mail diretamente usando Indy

O recurso de e-mail do ReportBuilder tem uma arquitetura de plug-in. ReportBuilder inclui plug-ins para usar Mapi e Indy. Por padrão, o ReportBuilder usa Mapi para enviar e-mail. No entanto, para enviar e-mail diretamente sem interação do usuário, você pode querer usar Indy, porque a segurança do Windows geralmente exibe uma caixa de diálogo de confirmação ao usar Mapi para enviar e-mail diretamente.

usa

```
ppSMTPIndy10; // ou use ppSMTPIndy9 ou  
ppSMTPIndy
```

```
TppSMTPPlugIn.RegisterClass (TppSMTPIndy);  
ppReport1.EmailSettings.PreviewInEmail Cliente: =  
False;  
ppReport1.EmailSettings.Subject: = 'Testar e-mail  
ReportBuilder';  
ppReport1.EmailSettings.Recipients.Text: = '  
somebody@somecompany.com';  
ppReport1.EmailSettings.Body.Text: = 'Ver relatório em  
anexo ...';  
ppReport1.SendMail;
```


DEPLOY

Introdução 135

Modelos de relatório 137

Como um EXE 139

Como Pacotes 141

Suporte a idiomas internacionais 143

DEPLOY

Introdução

Visão geral

Existem duas questões totalmente diferentes, mas vitais, envolvidas na decisão de como você implantará seu aplicativo:

1 Como implantar o ReportBuilder como parte de seu aplicativo

2 Como implantar os relatórios criados com ReportBuilder como parte de seu aplicativo

Implementação de aplicativos

Ao criar o exe do seu aplicativo, você pode optar por compilar tudo em um executável ou pode escolher distribuir os pacotes fornecidos com Delphi, ReportBuilder ou qualquer outro componente que esteja utilizando em seu aplicativo como arquivos separados. A última opção é conhecida como 'distribuição com pacotes' (pacotes aqui se referem a uma DLL compatível com Delphi) e resulta em um arquivo executável menor e mais eficiente.

Essas duas abordagens para implantar seu aplicativo são discutidas posteriormente nesta seção.

Implementação de relatórios

A segunda parte da implantação envolve determinar como implantar os relatórios criados com ReportBuilder como parte de seu aplicativo. ReportBuilder fornece as opções de implantação mais flexíveis de qualquer produto de relatório no mercado. As quatro estratégias de implantação mais comumente usadas são as seguintes:

1 Compile os relatórios no executável.

Permita que o componente de relatório resida em um formulário invisível junto com os objetos de acesso a dados necessários. Na hora de imprimir o relatório, instancie o formulário e chame o método Print. Uma ligeira variação dessa arquitetura seria usar um módulo de dados como um contêiner para os objetos de acesso a dados. Este método requer que o formulário contenha o componente ppReport.

2 Compile os relatórios em um pacote.

Essa estratégia é semelhante à opção 1, mas os formulários de relatório são compilados em um pacote em vez de em um executável. A vantagem dessa abordagem é que ela otimiza o tamanho do executável e carrega dinamicamente o pacote que contém os relatórios conforme necessário.

3 Distribua os arquivos de modelo de relatório.

Coloque todos os componentes de acesso a dados para os relatórios em um módulo de dados. Coloque um único componente de relatório em um formulário que 'usa' o módulo de dados. Crie todos os layouts de relatório por meio deste componente de relatório, salvando cada um em um arquivo de layout de relatório separado. Na hora de imprimir o relatório, carregue o relatório do arquivo e chame o método Print.

4 Distribua um banco de dados de relatórios.

Use todas as mesmas etapas da opção 3, mas salve as definições de relatório em um campo BLOB do banco de dados em vez de em arquivos separados. Com essa abordagem, você pode salvar todas as suas definições de relatório em uma tabela de banco de dados.

Uma implementação completa da Opção 1 é fornecida no tutorial 'Construindo um Aplicativo de Relatório'. A opção 1 tem a vantagem de exigir o mínimo de código. A principal vantagem das opções 3 e 4 é que você pode alterar os layouts de relatório e reimplantá-los para os usuários finais sem compilar o executável. Como a Opção 1 compila os layouts de relatório no executável, ela requer uma recompilação sempre que um relatório deve ser alterado. Em termos de tempo de carregamento do layout do relatório ou velocidade de impressão, todas as opções são essencialmente iguais.

Modelos de relatório

Visão geral

'Modelos de relatório' refere-se à definição de relatório criada quando um layout de relatório é salvo em um arquivo ou banco de dados. Você pode usar modelos de relatório para liberar suas definições de relatório dos limites do executável no qual estão sendo utilizados. Portanto, se você precisar fazer modificações em um relatório, poderá fazê-lo sem recompilar e reimplantar todo o seu aplicativo. Basta enviar o novo arquivo de layout de relatório ou tabela de banco de dados contendo os layouts de relatório para seus usuários.

Se você salvar layouts de relatório em arquivos ou banco de dados, poderá usá-los para criar um sistema de controle de versão para relatórios. Dessa forma, os relatórios originais fornecidos com seu aplicativo nunca são modificados, mas os usuários finais podem criar versões desses relatórios. Para permitir que os usuários finais modifiquem os relatórios por meio do Report Designer, é necessário o ReportBuilder Enterprise ou Pro.

Os modelos de relatório alavancam a tecnologia já presente no Delphi. A mesma lógica usada para salvar o estado dos objetos conforme configurado em um formulário Delphi (e armazenado em um arquivo dfm) é usada para salvar os componentes de um layout de relatório (em um arquivo rtm). Para visualizar o conteúdo de um arquivo rtm, defina `Report.Template.Format` como `ftASCII` e salve o relatório. Em seguida, abra o arquivo rtm resultante no editor de código Delphi. Você poderá ver a estrutura deste formato de arquivo.

Modelos baseados em arquivo

Você pode carregar e salvar layouts de relatório em arquivos usando o Report Designer. Você também pode carregar e salvar layouts de maneira programática.

1 Usando o Report Designer

Defina a propriedade `Report.Template.SaveTo` como `stFile` (este é o valor padrão).

UMA Salvar um layout de relatório

Selecione o arquivo | Opção de menu Salvar como ... de dentro do Report Designer. A caixa de diálogo padrão de salvamento de arquivo do Windows será exibida, permitindo que você salve o modelo de relatório em um arquivo .rtm.

B Carregando um modelo de relatório

Selecione o arquivo | Abrir ... opção de menu de dentro do Report Designer. A caixa de diálogo padrão de abertura de arquivo do Windows será exibida, permitindo que você selecione e abra um arquivo .rtm.

2 programaticamente

UMA Salvar um layout de relatório

O código a seguir salva um relatório no arquivo 'test.rtm':

```
ppReport1.Template.FileName = 'c: \ test.rtm';  
ppReport1.Template.SaveToFile;
```

B Carregando um modelo de relatório

O código a seguir abre um layout de relatório e imprime o relatório:

```
ppReport1.Template.FileName = 'c: \ test.rtm';  
ppReport1.Template.LoadFromFile;  
ppReport.Print;
```

Modelos de banco de dados

Você pode salvar e carregar modelos de relatórios baseados em dados usando o Report Designer ou programaticamente.

Definindo a Tabela do Banco de Dados

As definições de relatório podem ser armazenadas em qualquer banco de dados mesa com o seguinte estrutura abaixo:

Nome Char (40)

O nome do relatório é armazenado neste campo.

Modelo BLOB ou Memo

A definição do relatório é armazenada neste campo. Se o formato for binário, o campo deve ser um BLOB; se o formato for ASCII, o campo deve ser um memorando.

Conectando o objeto Relatório ao banco de dados

Depois de definir a tabela do banco de dados, a próxima etapa é conectar o relatório à tabela do banco de dados. Essa tarefa é realizada configurando as seguintes propriedades do objeto Report.Template. Você pode fazer isso por meio do Delphi Object Inspector ou programaticamente:

```
Report.Template.DatabaseSettings.NameField: =  
'Nome';  
Report.Template.DatabaseSettings.TemplateField: =  
'Modelo';  
Report.Template.DatabaseSettings.DataPipeline: =  
plReports;
```

Carregando e salvando relatórios

1 Usando o Report Designer

Defina a propriedade Report.Template.SaveTo como stDatabase.

UMA Salvar um layout de relatório

Selecione o arquivo | Opção de menu Salvar como ... de dentro do Report Designer. Uma caixa de diálogo especial Salvar será exibida, mostrando todos os relatórios na tabela de banco de dados fornecida. Você pode nomear o relatório e salvá-lo na tabela.

B Carregando um modelo de relatório

Selecione o arquivo | Opção de menu Salvar como ... de dentro do Report Designer. Depois que um relatório é salvo, você pode carregá-lo acessando a guia Arquivo | Abra a caixa de diálogo.

2 programaticamente

UMA Salvar um layout de relatório

O código a seguir salva um relatório no arquivo 'test.rtm':

```
ppReport1.Template.FileName: = 'Resumo do pedido';  
ppReport1.Template.SaveToDatabase;
```

B Carregando um modelo de relatório

O código a seguir abre um layout de relatório e imprime o relatório:

```
ppReport1.Template.DatabaseSettings.Name: =  
'Resumo do pedido';  
ppReport1.Template.LoadFromDatabase;  
ppReport.Print;
```

Como um EXE

Visão geral

O ReportBuilder pode ser implantado como parte de seu executável. Simplesmente inclua o caminho correto para as unidades compiladas (arquivos dcu) do ReportBuilder em seu caminho de biblioteca Delphi. Por exemplo:

```
'$ (BDS) \ RBuilder \ Lib'
```

O diretório 'lib' contém as unidades compiladas do ReportBuilder. Ao compilar seu aplicativo, os arquivos dcu necessários desse diretório serão vinculados ao arquivo executável resultante. Os arquivos executáveis criados dessa maneira são independentes, pois podem ser distribuídos com êxito sem nenhum arquivo de suporte.

Os aplicativos que contêm ReportBuilder não serão compilados corretamente se o caminho da biblioteca não estiver configurado corretamente. Se houver diretórios no caminho da Biblioteca apontando para versões antigas do ReportBuilder ou no diretório de origem do ReportBuilder, ocorrerão problemas.

Como pacotes

Visão geral

Você pode otimizar o tamanho do seu executável compilando seu projeto com pacotes de tempo de execução. Use a caixa de diálogo Opções do projeto para selecionar a opção 'Construir com pacotes de tempo de execução'. Isso habilitará uma caixa de edição que pode ser usada para especificar a lista de pacotes de tempo de execução usados pelo código do seu aplicativo. A ajuda online do ReportBuilder contém uma lista detalhada dos nomes dos pacotes de tempo de execução incluídos em cada edição do ReportBuilder. Acesse a ajuda online do ReportBuilder, selecione o índice e a seguir selecione a seção Instalação. Observe que os nomes dos pacotes ReportBuilder têm sufixos que se referem à versão ReportBuilder e à versão Delphi.

Suporte de idioma internacional

Visão geral

ReportBuilder inclui suporte para 12 idiomas integrados.

As cadeias de caracteres para cada idioma residem em arquivos de recursos separados. Esses arquivos estão localizados no diretório ..RBuilder \ Languages, em um subdiretório nomeado para o idioma fornecido. Códigos de três letras são usados para diferenciar entre as DLLs de idioma, que são instaladas no Win-diretório dows \ System32.

línguas

Língua	Tipo Enumerado	Pasta / extensão de arquivo
Padrão	IgDefault	dft
inglês	Inglês	eng
Personalizado	IgCustom	cst
dinamarquês	IgDanish	dan
holandês	IgDutch	nld
francês	IgFrench	fra
alemão	IgGerman	deu
italiano	italiano	ita
Portugues (brasil)	IgPortugueseBrazil	ptb
português	Ig português	ptg
espanhol	IgSpanish	esp
Espanhol (México)	IgSpanishMexico	ESM
sueco	Ig sueco	sve
norueguês	Ig norueguês	nem

Apenas os idiomas que são explicitamente selecionados durante a instalação aparecerão na pasta Idiomas. A pasta 'cst' sempre aparecerá na pasta Idiomas porque representa o tipo de idioma personalizado. A pasta cst contém os arquivos de recursos correlacionados ao tipo enumerado IgCustom. O idioma personalizado pode ser modificado para conter sua própria tradução, que aparecerá na interface do usuário ReportBuilder quando a propriedade Report.Language for definida como IgCustom.

O idioma padrão

Quando o ReportBuilder é instalado, o programa de instalação permite que você especifique um idioma padrão e quaisquer idiomas adicionais que deseja usar. A instalação irá então instalar os arquivos apropriados em seu sistema. O padrão da propriedade Report.Language é o tipo de idioma IgDefault. Este tipo está correlacionado aos arquivos de recursos de idioma localizados no diretório RBuilder \ Lib e às DLLs de idioma localizados no diretório Windows \ System32 (as DLLs com a extensão .dft). Os arquivos .res são usados ao compilar executáveis autônomos no Delphi. As DLLs são usadas para fornecer suporte de linguagem em tempo de design do Delphi. Quando o ReportBuilder é instalado, ele copia esses arquivos para os diretórios apropriados.

As DLLs de idioma que aparecem em Windows \ System32 estão listadas abaixo:

Nome do arquivo	Descrição
rbPrint.dft	Strings usadas por ReportBuilder
rbIDE.dft	Strings usadas pelo Report Designer

Os arquivos de recursos que aparecem em RBuilder \ Lib são listados abaixo:

Nome do arquivo	Descrição
rbPrint.res	Strings usadas por ReportBuilder
rbIDE.res	Strings usadas pelo Report Designer

Você pode alterar o idioma padrão usado pelo ReportBuilder copiando os arquivos de idioma apropriados do diretório RBuilder \ Idiomas para os diretórios RBuilder \ Lib e Windows \ System32, respectivamente. Para os arquivos DLL, certifique-se de alterar a extensão para .dft. Os seguintes comandos de cópia mudarão o idioma padrão para sueco quando emitidos no prompt de comando (de dentro do diretório RBuilder \ Languages \ sve):

```
copiar rbPrint.res
{InstallPath} \ RBuilder \ Lib \ RBuilder \ Lib copiar
rbIDE.res
{InstallPath} \ RBuilder \ Lib \ RBuilder \ Lib cópia
rbPrint.sve
c: \ windows \ system \ rbPrint.dft copy
rbIDE.sve
c: \ windows \ system \ rbIDE.dft
```

Traduções Customizadas

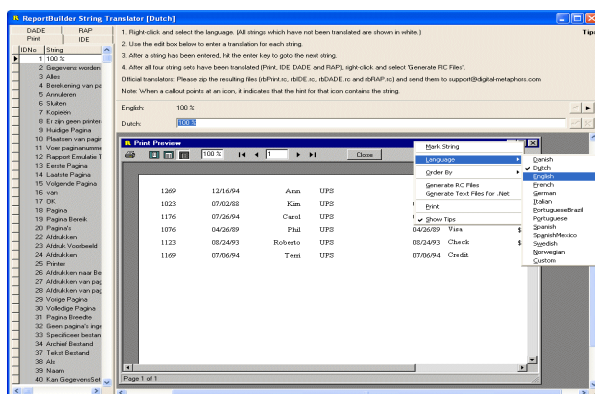
Você pode modificar os arquivos de origem .rc na pasta de idioma personalizado para criar sua própria tradução personalizada. Os arquivos rc são nomeados da mesma forma que suas contrapartes de arquivo de recursos:

Nome do arquivo	Descrição
rbPrint.rc	Strings usadas por ReportBuilder
rbIDE.rc	Strings usadas pelo Report Designer
rbDADE	Strings usadas pelo ambiente de desenvolvimento de acesso a dados (a guia 'Dados')

Você pode criar uma tradução personalizada concluindo as seguintes etapas:

1 Execute o aplicativo de tradução de idiomas.

Crie novos arquivos rc usando o aplicativo ReportBuilder Language Translation. Este aplicativo exibe as strings usadas no ReportBuilder junto com uma captura de tela associada. Ele fornece uma caixa de edição na qual a tradução de cada string pode ser inserida. Ao concluir a tradução, clique no botão 'Gerar arquivos RC' e gere cada um dos três arquivos rc. O aplicativo de tradução de idiomas é mostrado abaixo.



2 Crie novos arquivos de recursos.

Compile os arquivos rc em arquivos de recursos emitindo os seguintes comandos no prompt de comando (na pasta 'cst'):

```
{InstallPath} \ Bin \ brc32 -r rbPrint.rc
{InstallPath} \ Bin \ brc32 -r rbIDE.rc
{InstallPath} \ Bin \ brc32 -r rbDADE.rc
```

Após a execução desses comandos, você deve ter três arquivos de recursos na pasta cst: rbPrint.res, rbIDE.res e rbDADE.res.

3 Crie novos módulos de recursos.

Compile os arquivos de recursos em DLLs. Use os seguintes comandos para fazer esta etapa:

```
{InstallPath} \ Bin \ dcc32 rbPrint.dpr
{InstallPath} \ Bin \ dcc32 rbIDE.dpr
{InstallPath} \ Bin \ dcc32 rbDADE.dpr
```

Após a execução desses comandos, você deve ter três arquivos cst na pasta cst: rbPrint.cst, rbIDE.cst e rbDADE.cst.

4 Instale novos módulos de recursos.

A etapa final é copiar os novos arquivos .res e DLLs para os diretórios ..RBuilder \ Lib e Windows \ System32 respectivamente. Use os seguintes comandos para fazer esta etapa:

```
copiar rbPrint.cst c: \ windows \ system32 copiar
rbIDE.cst c: \ windows \ system32 copiar
rbDADE.cst c: \ windows \ system32
```

5 Use a tradução personalizada.

Crie um relatório no Delphi; selecione o tipo de idioma IgCustom e clique duas vezes no relatório para exibir o Report Designer. Sua tradução será exibida.

REPORTBUILDER

ENTERPRISE EDIÇÃO

FSEM DANOS

Principal 151

Dados 171

Código 195

Design 221

Imprimir 225

Implantar 229

MAIN

Introdução 151

O componente Delphi ts 155

Componente de Designer 157

Report Explorer 161

Dicionário de Dados 163

Juntando Tudo 165 Ajuda

On-Line 167

MAIN

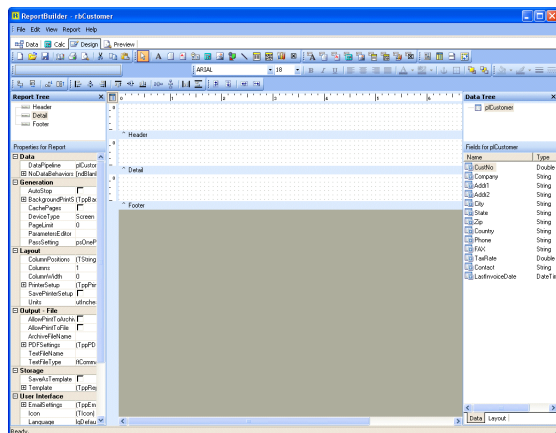
Introdução

Visão geral

O ReportBuilder Enterprise Edition inclui tudo no ReportBuilder Standard, além de um conjunto completo de componentes necessários para fornecer uma solução completa de geração de relatórios para o usuário final. Na introdução ao ReportBuilder, a equação de relatório é descrita. A equação do relatório divide o relatório em quatro atividades principais:



No ReportBuilder Enterprise, o objetivo é fornecer uma solução de relatórios completa para os usuários finais. Esse objetivo é alcançado com a entrega de soluções visuais e fáceis de usar em cada uma dessas quatro áreas. Esta captura de tela do ReportBuilder Enterprise Report Designer mostra o design ergonômico da interface do usuário.



Cada uma das quatro áreas de relatório tem uma guia de bloco de notas representativa contendo um ambiente visual para a criação e configuração de componentes dentro dessa área. Os resultados de cada área tornam-se dados de entrada para a próxima área: os dados são inseridos nos cálculos, os cálculos são inseridos nos componentes do layout do relatório e o layout do relatório é renderizado em uma visualização do relatório. A implementação usada pelo ReportBuilder Enterprise para cada área de relatório é descrita abaixo.

Dados

No ambiente de trabalho da guia Dados, os usuários finais podem criar visualizações de dados rapidamente, que podem ser usadas para fornecer dados aos relatórios. As visualizações de dados geralmente são criadas por meio do Query Wizard ou Query Designer. Ambas as ferramentas são visuais; eles também permitem que o usuário final selecione as tabelas, campos, critérios de pesquisa e ordem de classificação necessários para o relatório. Nos bastidores, uma instrução SQL é gerada e usada para recuperar os dados do banco de dados. Uma captura de tela de uma visualização de dados concluída é mostrada abaixo.

Name	Type	Size
Addr1	String	30
Addr2	String	30
City	String	15
Company	String	30
Contact	String	20
Country	String	20
Custno	Double	
Fax	String	15
Lastinvoicedate	DateTime	
Phone	String	15
State	String	20
Taxrate	Double	
Zip	String	10

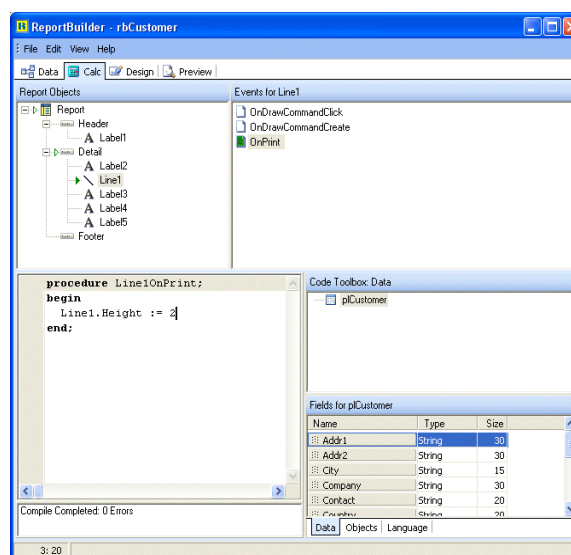
A solução descrita na página anterior é o comportamento padrão no espaço de trabalho de dados. No entanto, o desenvolvedor pode personalizar essa interface do usuário executando uma destas três tarefas:

- Registre um assistente de consulta ou designer de consulta substituto.
- Remova o assistente de consulta ou designer de consulta.
- Criar novas classes de modelo de visualização de dados que podem simplificar ainda mais o processo de seleção de dados, estabelecendo a relação entre as tabelas no banco de dados e apresentando uma interface de usuário alternativa (como um único formulário que permite a inserção de critérios de pesquisa / classificação).

O resultado final é que a área de Dados contém uma solução pronta para uso que pode ser usada imediatamente, mas se personalizações forem necessárias, uma arquitetura foi fornecida para que essas personalizações sejam possíveis.

Calc

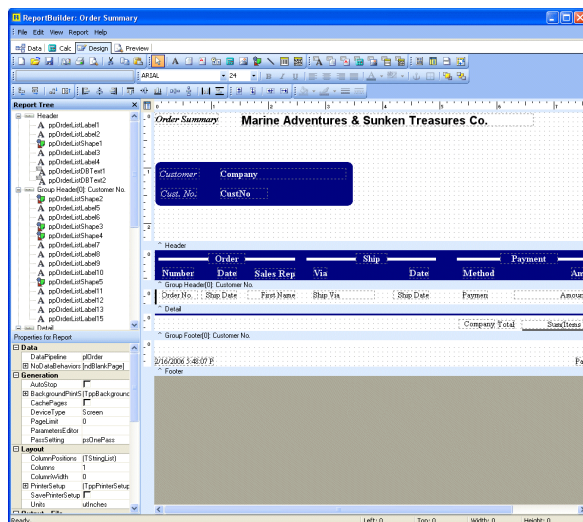
Este espaço de trabalho contém uma visualização em árvore do relatório, todas as bandas dentro do relatório e todos os objetos dentro de cada banda. Quando uma banda ou componente é selecionado, todos os eventos para aquele componente são mostrados em uma lista. O usuário pode então selecionar um evento e codificar o manipulador de eventos no editor de código sensível à sintaxe na parte inferior. A captura de tela a seguir mostra um evento OnPrint codificado para um componente Line.



Esta captura de tela mostra o espaço de trabalho do Calc em sua configuração mais complexa e rica em recursos. O trabalho de desenvolvimento concluído aqui pode ser repassado aos usuários finais para que eles possam modificá-lo, bloqueado para que os usuários finais possam apenas visualizá-lo ou ocultá-lo completamente para que os usuários finais não saibam que está lá. A versão mais reduzida da guia Calc é fornecida por uma interface somente de diálogo, onde nenhuma guia Calc existe e uma única caixa de diálogo do editor de código sensível à sintaxe é acessível a partir da opção de menu Cálculos ... da variável componente. Mais uma vez, a interface do usuário e a arquitetura dos bastidores tornaram-se altamente escaláveis para atender às várias necessidades dos desenvolvedores.

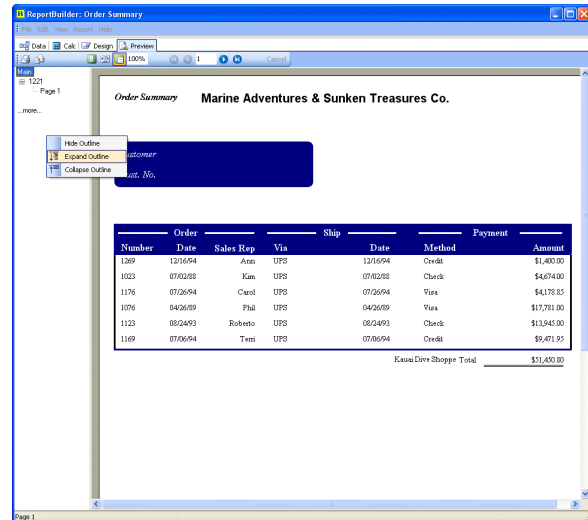
Projeto

A área de trabalho Design contém o layout real do relatório. A interface do usuário é idêntica àquela apresentada aos desenvolvedores que usam o ReportBuilder no tempo de design do Delphi; em outras palavras, é completo e profissional. A interface de estilo Office torna a área de trabalho Design especialmente fácil de aprender para os usuários finais. Um Assistente de Relatório está disponível para criar relatórios rapidamente. Você pode personalizar essa interface substituindo qualquer uma das caixas de diálogo que ela usa e registrando seus próprios assistentes de relatório.



Antevisão

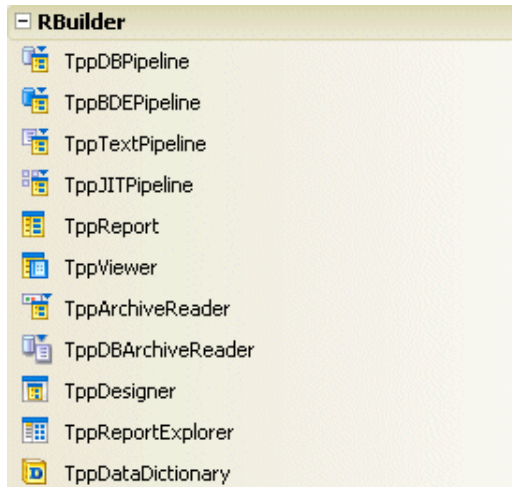
A área de trabalho Visualização contém o relatório renderizado. O relatório pode ser impresso na impressora ou em vários formatos de arquivo neste espaço de trabalho.



Os Componentes Delphi

Visão geral

Os seguintes componentes aparecem na guia RBuilder da paleta de componentes Delphi quando você instala o ReportBuilder Enterprise:



DBPipeline

Usado para acessar dados por meio de descendentes de TDataSet. O DBPipeline é conectado por meio da propriedade DataSource.



BDEPipeline

Fornecido para compatibilidade com versões anteriores. Use DBPipeline para todos os novos desenvolvimentos.



TextPipeline

Usado para acessar vírgula, guia e arquivos de texto de registro de comprimento fixo. Defina a propriedade FileName para especificar o arquivo. Clique duas vezes no componente para definir a estrutura do campo.



JITPipeline

Usado para acessar dados armazenados em estruturas de dados proprietárias, matrizes, objetos, etc. Defina as propriedades InitialIndex e RecordCount. Clique duas vezes no componente para iniciar o Editor de campos e definir os campos. Implemente o evento OnGetFieldValue para retornar o valor do campo correspondente a CurrentIndex e FieldName.



Relatório

O principal componente. Clique duas vezes para invocar o Report Designer. Atribua a propriedade DataPipeline para que o relatório possa percorrer os dados. Atribua a propriedade DeviceType para controlar para onde a saída do relatório é direcionada. Chame Report.Print de Object Pascal para imprimir o relatório ou inicie o formulário de visualização de impressão.



Visualizador

Este objeto raramente é usado porque você pode substituir o formulário de visualização de impressão integrado do ReportBuilder por sua própria versão personalizada com muita facilidade (verifique o tutorial Construindo um Aplicativo de Relatório). Se você deve usar este componente, um exemplo é fornecido em \RBuilder\Demos\Reports.



ArchiveReader

Depois de imprimir um relatório em um arquivo compactado (extensão .raf), você pode ler e visualizar o arquivo por meio deste componente. Basta atribuir o ArchiveFileName ao arquivo e chamar o método Print. Em termos de exibição de um relatório, este componente funciona da mesma forma que o componente Relatório.



Dicionário de dados

Usado pelo componente QueryWizard para converter nomes brutos de tabelas e campos em aliases mais utilizáveis. Este componente é geralmente usado apenas quando você está criando uma solução de relatório de usuário final.



Designer

Usado quando você deseja implantar o Report Designer para seus usuários finais. Consulte a demonstração do usuário final (diretório Demos \ EndUser).



ReportExplorer

Usado para fornecer uma interface no estilo do Windows Explorer para soluções de Relatórios do Usuário Final desenvolvidas com o ReportBuilder Enterprise. Consulte a demonstração do usuário final (Demos \ EndUser \ 1 Report Explorer \ diretório).



Componente de Designer

Visão geral

O Designer é um componente não visual que atua como um 'invólucro' em torno da janela Report Designer exibida na solução de relatório do usuário final. Ao definir as várias propriedades e eventos do componente designer, você personaliza o conteúdo e o comportamento do Report Designer. As propriedades mais comumente usadas estão listadas abaixo.

AllowDataSettingsChange

Usado para controlar se a opção de menu DataSettings aparece no menu Arquivo da área de trabalho de dados. Normalmente desabilitado para evitar que os usuários finais alterem as configurações do banco de dados.

AllowSaveToFile

Quando definido como True, o menu Arquivo da área de trabalho de design tem as opções de menu 'Carregar do arquivo' e 'Salvar no arquivo'.

DataSettings

DADE apenas. Usado para configurar DADE (espaço de trabalho de dados) para um determinado banco de dados.

Ícone

O ícone a ser exibido na janela Report Designer.

InitStorageType e IniStorageName

Usado para controlar como e onde as preferências são armazenadas. As preferências podem ser armazenadas no Registro ou em um arquivo .Ini. Consulte a ajuda online para obter mais detalhes.

RAPInterface

Apenas RAP. Definições de configuração RAP.

RAPOptions

Apenas RAP. Outras definições de configuração RAP.

Relatório

O objeto de relatório que será usado para carregar e salvar relatórios. A configuração das propriedades do modelo de relatório é usada pelo Report Designer para determinar como os relatórios serão carregados e salvos.

Mostrar componentes

Determina quais componentes serão exibidos nas paletas de componentes do ReportBuilder. Às vezes usado para desativar componentes avançados, como regiões ou sub-relatórios.

TabsVisible

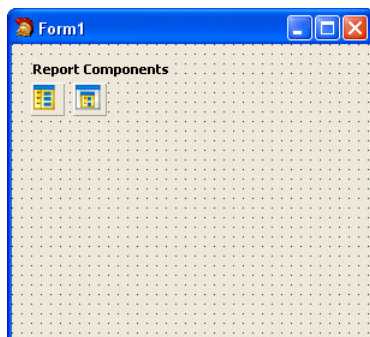
Determina se as guias Design e Visualização serão exibidas. Quando definido como falso, o Report Designer se torna uma janela 'somente design', onde o usuário final pode trabalhar no layout, mas não pode visualizar. Se DADE ou RAP estiver instalado, esta propriedade não pode ser definida como False.

Uma solução simples de relatório de usuário final

Podemos criar uma solução simples de relatório para o usuário final usando apenas um componente de designer e um componente de relatório.

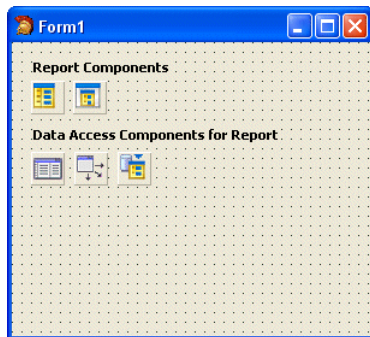
1 Crie componentes de Relatório e Designer.

Coloque um componente de designer e um componente de relatório em um formulário e, a seguir, conecte o relatório ao designer atribuindo a propriedade Designer.Report.

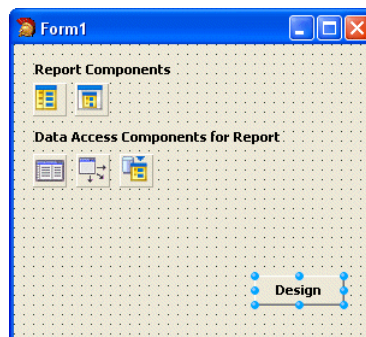


2 Crie componentes de acesso a dados.

Crie componentes de acesso a dados para fornecer dados ao relatório. Aqui criamos os componentes TTable, TDataSource e TDBPipeline. O TableName do componente TTable foi definido como 'customer.db' para que o usuário possa criar um relatório com base nos dados desta tabela.



3 Crie um botão para iniciar o Report Designer.



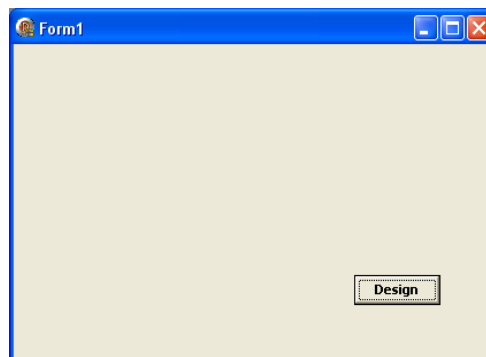
4 Codifique o evento OnClick do botão para iniciar o designer.

```

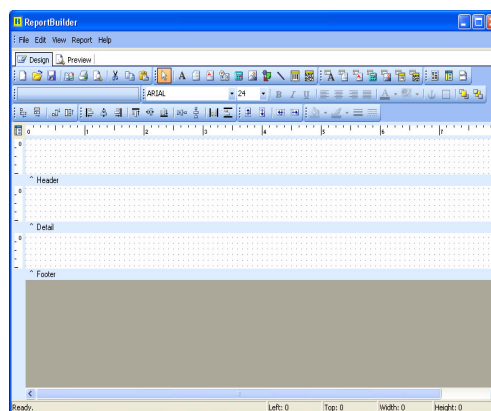
Welcome Page Unit1
procedure TForm1.Button1Click
begin
    ppDesigner1.ShowModal;
end;
39 end.

```

5 Execute o aplicativo.



6 Clique no botão 'Design' para iniciar o designer.



O usuário agora pode criar um relatório com base na tabela do cliente. O layout do relatório pode ser salvo em um arquivo ou os layouts pré-existentes podem ser carregados do arquivo. Os relatórios só podem ser criados com base na tabela do cliente, uma vez que é a única tabela disponível para o relatório.

Embora essa solução para o usuário final seja simples e fácil, ela tem algumas limitações:

1 Essa abordagem não fornece um espaço de trabalho de dados onde dados diferentes, além da tabela do cliente, podem ser acessados.

2 Essa abordagem não fornece um espaço de trabalho de cálculo onde cálculos personalizados ou manipuladores de eventos podem ser atribuídos ao relatório.

3 A capacidade de salvar ou abrir layouts de relatório em arquivo é baseada em diálogo e, portanto, o usuário não tem uma maneira de organizar e controlar os muitos relatórios que pode criar.

Essas três limitações podem ser tratadas com funcionalidades adicionais disponíveis no ReportBuilder Enterprise, ou seja, DADE, RAP e Report Explorer. Essas soluções são discutidas em detalhes ao longo desta seção do guia.



Report Explorer

Visão geral

O Report Explorer permite que seus usuários finais usem uma interface de usuário do Windows Explorer para gerenciar relatórios armazenados em um banco de dados. O Report Explorer é um componente não visual que atua como um 'wrapper' em torno do formulário do explorer, que é exibido em tempo de execução. Você pode definir o valor de várias propriedades em tempo de design por meio deste componente, o que afetará o comportamento do formulário explorador de relatório em tempo de execução. As propriedades mais comumente usadas estão listadas abaixo.

Designer

O componente de designer que será usado para exibir a janela Report Designer quando os layouts de relatório forem criados ou editados.

FolderFieldNames

Os nomes dos campos a serem usados ao armazenar uma pasta.

FolderPipeline

O pipeline de dados a ser usado ao armazenar uma pasta.

ItemFieldNames

Os nomes dos campos a serem usados ao armazenar um item (ou seja, relatório, módulo de código ou módulo de dados).

ItemPipeline

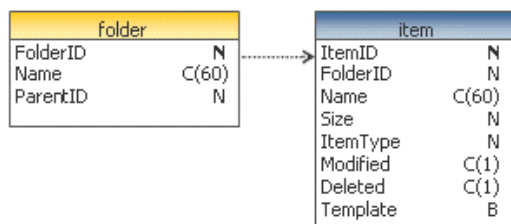
O pipeline de dados a ser usado ao armazenar um item.

ModalForm

Se o formulário do explorador de relatórios deve ou não ser exibido modalmente. O padrão é True.

O explorador de relatórios é projetado em torno do conceito de pastas e itens. 'Itens' incluem layouts de relatório, módulos de dados ou módulos de código. Os layouts de relatório consistem em tudo o que o usuário final vê ao projetar um relatório, incluindo as visualizações de dados no espaço de trabalho de dados e os manipuladores de eventos codificados no espaço de trabalho calc. Módulos de dados consistem em visualizações de dados que foram exportadas do espaço de trabalho de dados para que possam ser importados para uso na criação de novos relatórios. Módulos de código consistem em manipuladores de eventos que foram exportados do espaço de trabalho calc para que possam ser importados para outros relatórios.

Os itens exibidos no explorer devem pertencer a uma determinada pasta. As pastas podem ser aninhadas em outras pastas ou podem ser atribuídas à pasta pai global (intitulada 'Todas as pastas'), que é sempre exibida. Tanto as pastas quanto os itens são armazenados por meio de pipelines de dados.



O campo FolderId da tabela Folder é o campo-chave e deve ser sempre exclusivo. No Paradox, este campo é geralmente um AutoIncrement. Em tabelas de banco de dados SQL, esse campo geralmente é um número de sequência gerado por um gatilho. Ambas as técnicas são documentadas nos tutoriais do usuário final. A relação entre a pasta e a tabela de itens é de um para muitos. Isso não quer dizer que a tabela de itens não use o campo Nome como uma chave: ela usa um campo ItemId. Este campo também é um número de sequência de geração automática. Embora seja geralmente verdade que dois itens com o mesmo nome não podem residir na mesma pasta (mesmo nome e id de pasta), isso não é verdade para a lixeira, que pode conter vários itens com o mesmo nome. Assim, os campos FolderId-ItemType-Name não podem formar uma chave única (o campo de tipo de item indica se o item é um layout de relatório, módulo de dados,

Todos os itens atribuídos à lixeira têm uma id de pasta de -2. Todos os itens atribuídos à pasta global 'Todas as pastas' têm um id de pasta de -1. Os itens atribuídos a qualquer outra pasta têm o valor de id dessa pasta no campo id da pasta. Nem a pasta 'Todas as pastas' nem a lixeira realmente aparecem na tabela de pastas: elas são criadas automaticamente pelo explorador de relatórios.

 Dicionário de dados

Visão geral

O dicionário de dados é um componente não visual projetado para fornecer ao espaço de trabalho de dados a capacidade de substituir os nomes de tabelas e nomes de campos recuperados do banco de dados por aliases mais legíveis fornecidos por você. As propriedades mais comumente usadas do dicionário de dados são descritas abaixo.

FieldFieldNames

Os nomes dos campos a serem usados para acessar os aliases de campo.

FieldPipeline

O nome do pipeline de dados que tem acesso aos aliases de campo.

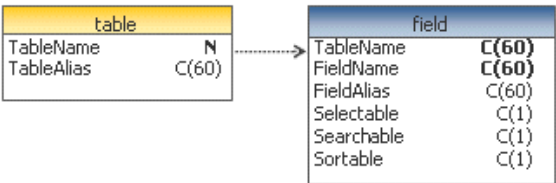
TableFieldNames

Os nomes dos campos a serem usados para acessar os aliases da tabela.

TablePipeline

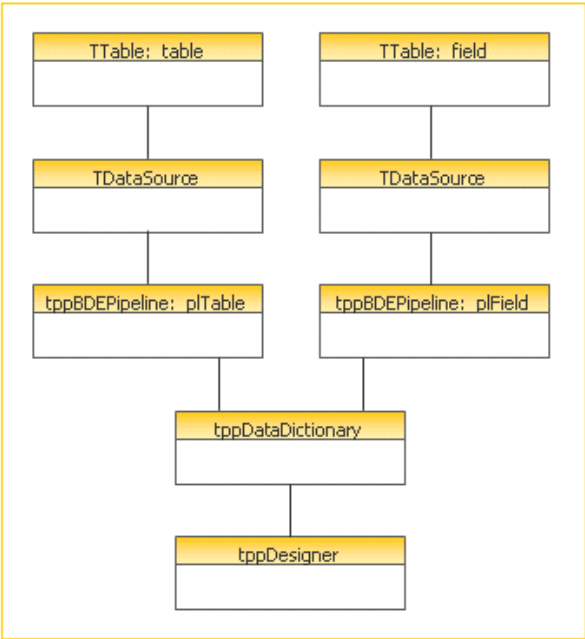
O nome do pipeline de dados que tem acesso aos aliases da tabela.

O módulo de dados para as tabelas de banco de dados subjacentes ao dicionário de dados é mostrado abaixo.



O TableName é o campo-chave da tabela da tabela. O TableName e FieldName são os campos-chave da tabela de campos. O campo Seleccionável determina se o campo pode ser selecionado no Assistente de Consulta e no Designer de Consulta. O campo Pesquisável determina se o campo pode ser usado nos critérios de pesquisa. O campo Classificável determina se o campo pode ser usado para ordenar os dados.

Uma vez que as tabelas de dicionário de dados foram criadas e preenchidas, elas podem ser atribuídas a um componente Delphi TTable ou TQuery padrão, que pode então ser atribuído a um pipeline de dados por meio de um TDataSource. Os pipelines são então atribuídos às propriedades TablePipeline e FieldPipeline do dicionário de dados. O diagrama abaixo mostra um componente de dicionário de dados totalmente configurado. O componente de dicionário de dados é atribuído ao espaço de trabalho de dados por meio da propriedade DataDictionary do componente designer.

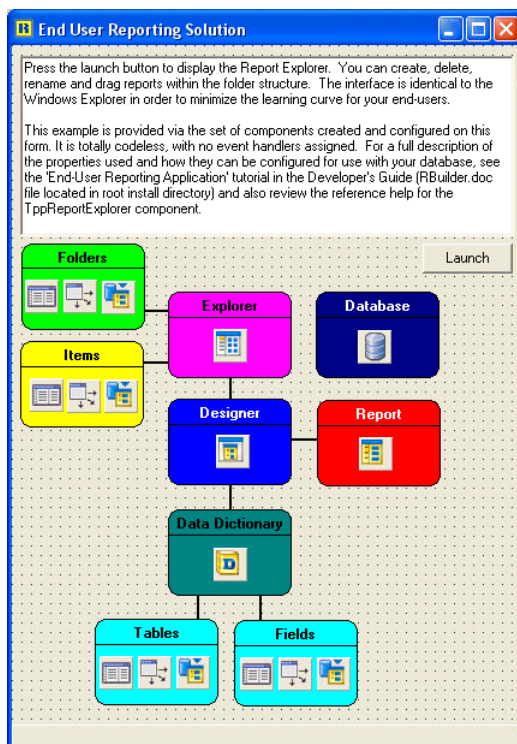


Juntando tudo

Resumo

Para criar a solução de relatório de usuário final mais sofisticada e completa possível, precisamos conectar e configurar todos os componentes disponíveis no ReportBuilder Enterprise. A captura de tela abaixo mostra a configuração de uma solução de relatório de usuário final com recursos completos (as formas e rótulos coloridos foram adicionados apenas para fins informativos). Uma versão completa desta solução está disponível no

\ RBuilder \ Demos \ EndUser \ 1. Diretório do Report Explorer. Instruções passo a passo para criar este aplicativo são fornecidas no tutorial 'Construindo um aplicativo de relatório do usuário final'.



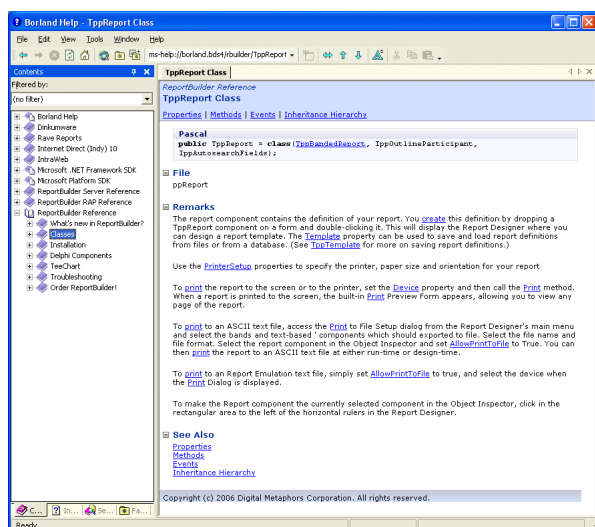
Ajuda online

Visão geral

A Ajuda On-Line do ReportBuilder contém mais 1.500 tópicos que documentam as propriedades, eventos e métodos dos vários componentes que constituem o produto.

Para máxima legibilidade, a ajuda segue o estilo exato da ajuda do Delphi. Ele é projetado para ser integrado com a ajuda do Delphi de forma que quando você acessar a ajuda do Delphi (através da opção de menu Ajuda | Tópicos), 'Referência do ReportBuilder' apareça como uma única entrada na página de conteúdo. Os tópicos da ajuda do ReportBuilder aparecerão quando você pesquisar a ajuda na página 'Índice'. Além disso, ao criar um componente em um formulário Delphi, você pode acessar o tópico de ajuda principal para aquele componente clicando em F1. Se você selecionar uma propriedade de componente no Inspetor de objetos e clicar em F1, a ajuda para aquela propriedade individual será exibida.

A ajuda foi projetada como uma referência para aqueles que já dominam os fundamentos do ReportBuilder. Para começar a usar o produto, estude este guia e conclua os tutoriais.



DATA

Introdução 171

Assistente de Consulta 173

Designer de Consultas 177

Configurando DADE 185

Arquitetura DADE 187

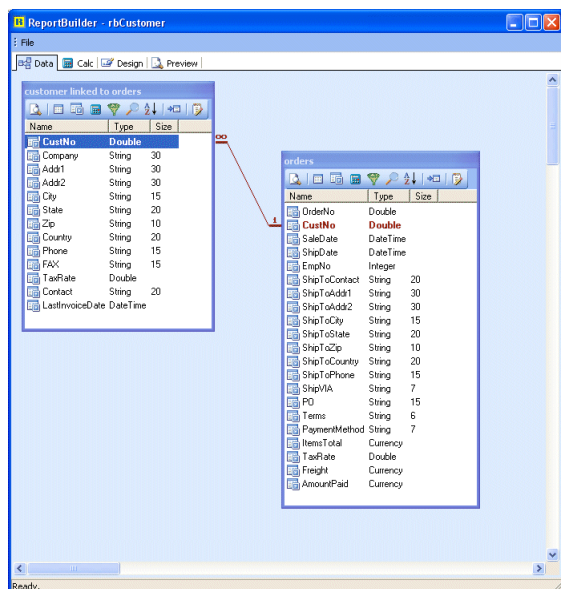
Estendendo DADE 189

DATA

Introdução

Visão geral

Os desenvolvedores do ReportBuilder descobriram que uma solução visual para acesso a dados era necessária para os usuários finais. Essa constatação levou ao desenvolvimento do DADE, o Ambiente de Desenvolvimento de Acesso a Dados. Embora o DADE consista em uma extensa arquitetura orientada a objetos "nos bastidores", ele aparece como um espaço de trabalho de dados simples e fácil de usar dentro do Report Designer. A área de trabalho de dados do Report Designer, que contém uma visão de dados completa, é mostrada abaixo.



Exibição de dados

Uma visualização de dados apresenta o que parece ser um único conjunto de dados para o usuário final. Na realidade, a visualização de dados pode ser composta de um ou mais componentes de acesso a dados Delphi. A implementação de uma visualização de dados pode ser tão simples quanto uma única consulta SQL ou tão complexa quanto um conjunto de tabelas vinculadas. Qualquer que seja

implementação, a visualização de dados sempre parece a mesma para o usuário final, e a implementação sempre depende das escolhas que você faz como desenvolvedor.

Os Dataviews fazem interface com os outros espaços de trabalho dentro do Report Designer por meio dos pipeline (s) de dados que eles contêm. A visualização de dados à esquerda contém um conjunto de tabelas mestre / detalhes. Essas tabelas alimentam dados por meio de componentes Delphi TDataSource padrão, que por sua vez alimentam os dados por meio de componentes de pipeline de dados. Quando essa visualização de dados é criada pela primeira vez, ela atribui o pipeline de dados do cliente ao relatório. Os pipelines de pedidos e itens estão disponíveis para o usuário final para uso em sub-relatórios.

Os componentes do pipeline de dados em uma visualização de dados criam a interface entre a visualização de dados e as outras áreas de trabalho no Report Designer. No espaço de trabalho de design, os pipelines de dados para todas as visualizações de dados são listados na árvore de dados, na lista suspensa da barra de ferramentas de edição e na caixa de diálogo de dados, que pode ser acessada no menu principal. No espaço de trabalho calc, os pipelines de dados aparecem na lista de objetos e na caixa de ferramentas de código, tornando mais fácil codificar cálculos envolvendo valores de campo.

Discutiremos DADE, visualizações de dados e como usar esta parte do ReportBuilder para maximizar a produtividade de seus usuários finais na seção seguinte.

Assistente de Consulta

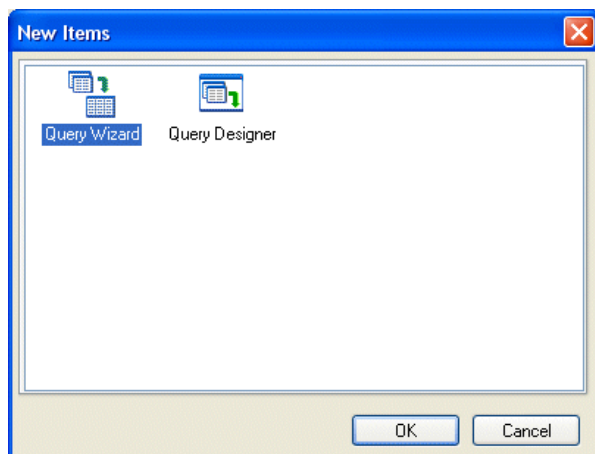
Visão geral

Dentro da área de trabalho de dados, você pode selecionar dados de seu banco de dados usando uma consulta SQL. Essa funcionalidade é fornecida por meio de visualizações de dados baseadas em consulta, que podem ser visualmente criadas usando o Query Wizard e mantidas visualmente usando o Query Designer.

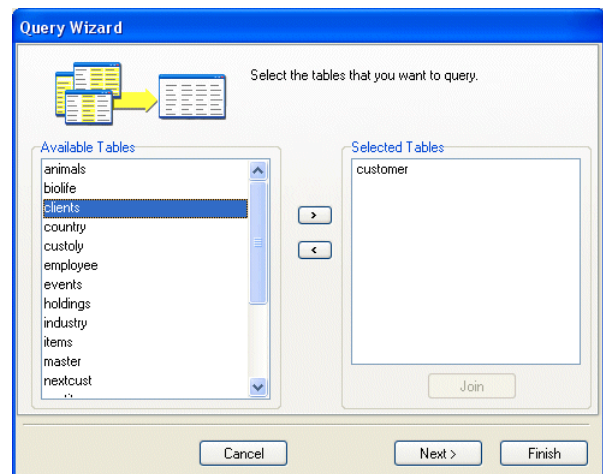
Assistente de consulta: crie um Dataview baseado em consulta simples

A seguinte série de capturas de tela mostra como criar uma visualização de dados simples baseada em consulta por meio do Query Wizard.

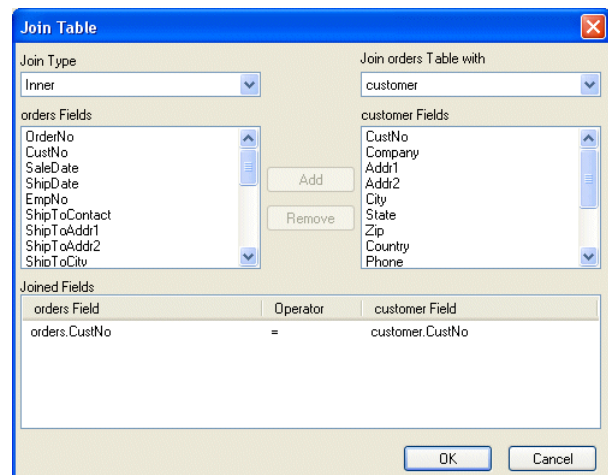
1 Selecione Arquivo | Novo no espaço de trabalho de dados. A nova caixa de diálogo será exibida. Clique duas vezes no ícone do Assistente de consulta.



2 Selecione a primeira tabela para a consulta. A tabela do cliente foi selecionada.

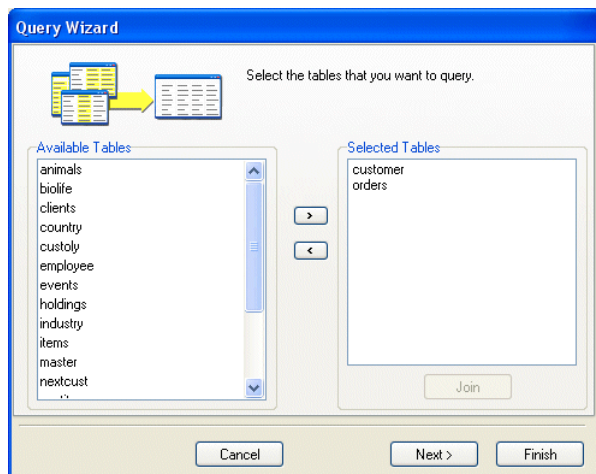


3 Selecione a segunda tabela para a consulta. A tabela de pedidos foi selecionada.

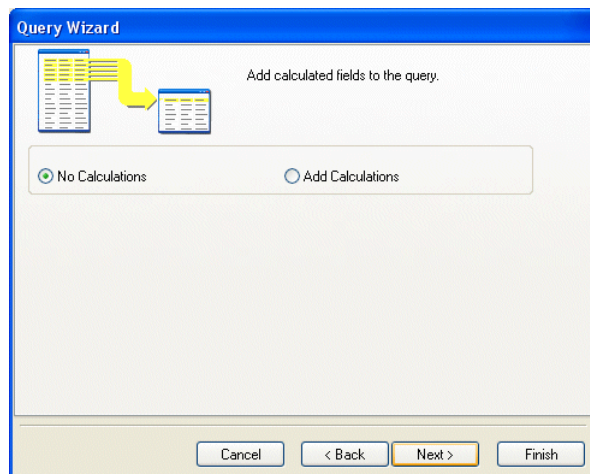


Assistente de consulta: crie um Dataview simples baseado em consulta - cont.

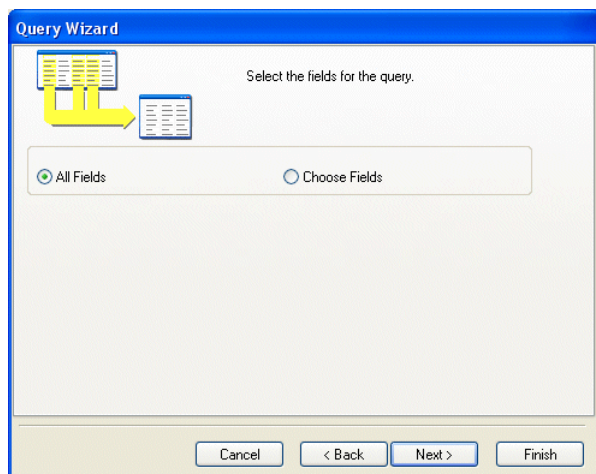
4 Quando retornamos ao assistente de consulta, ambas as tabelas são mostradas conforme selecionadas.



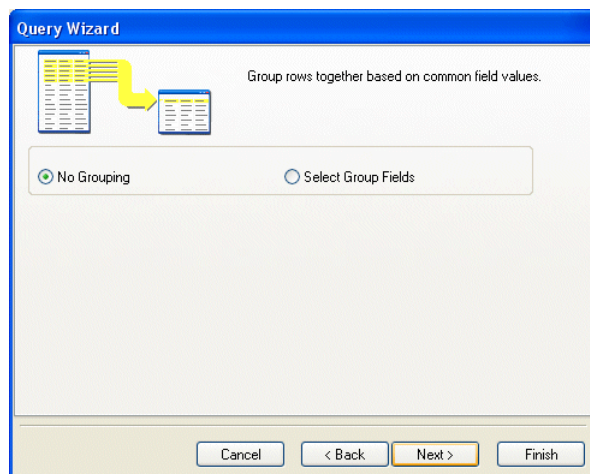
6 Pule a página de cálculos, pois esta consulta não conterá cálculos.



5 Pule a página de campos, pois queremos selecionar todos os campos.

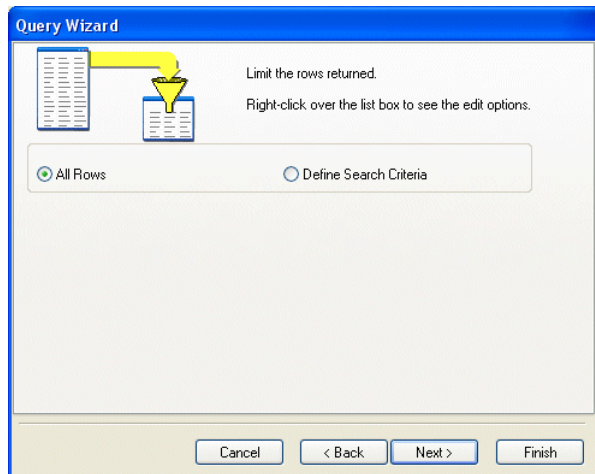


7 Pule a página de grupos, pois esta consulta não será agrupada.

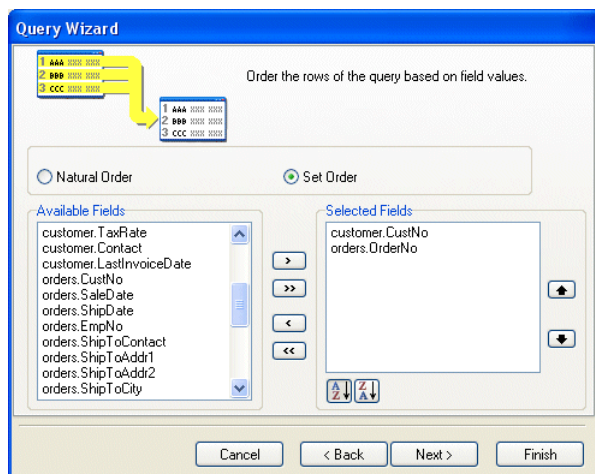


Assistente de consulta: crie um Dataview simples baseado em consulta - cont.

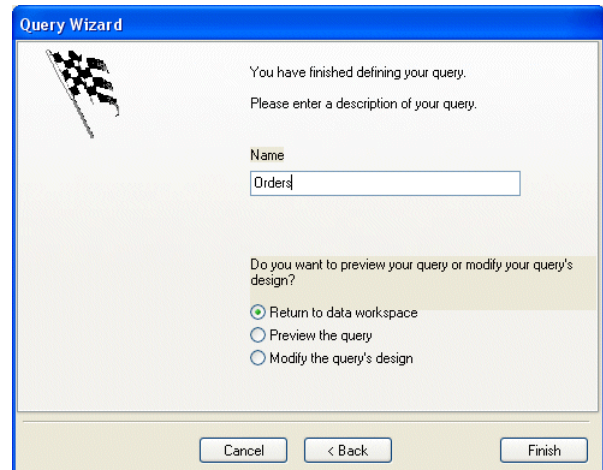
8 Pule a página de critérios de pesquisa; todos os registros serão selecionados.



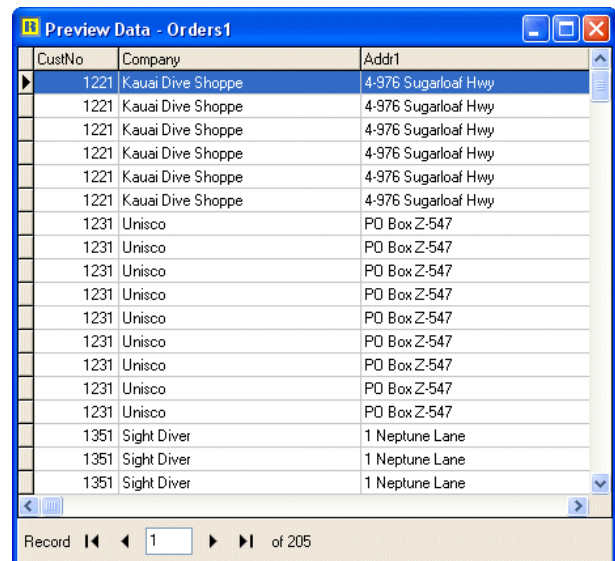
9 Defina o pedido como o número do cliente e, em seguida, o número do pedido.



10 Nomeie a visualização de dados. O nome do pipeline de dados é gerado automaticamente quando nomeamos a visualização de dados. A próxima ação será visualizar os dados.

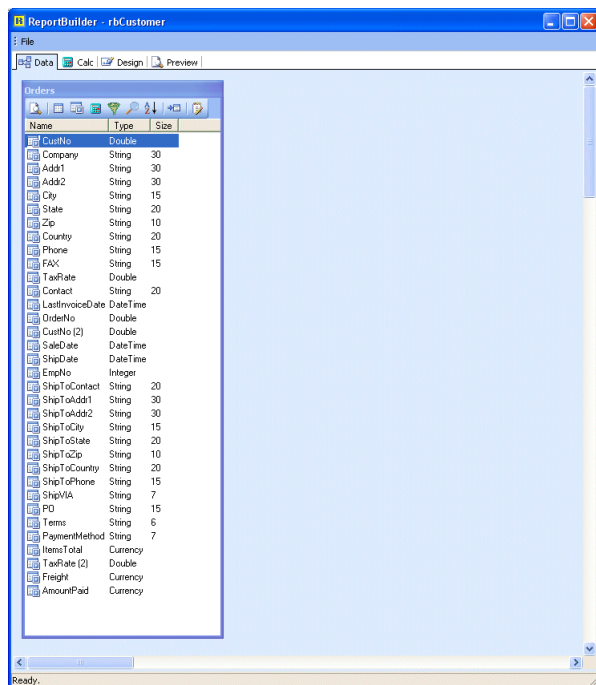


11 Visualize os dados para certificar-se de que os registros corretos foram selecionados.



Assistente de consulta: crie um Dataview simples baseado em consulta - cont.

12 Feche a janela de visualização. A visualização de dados é então criada e exibida na área de trabalho.

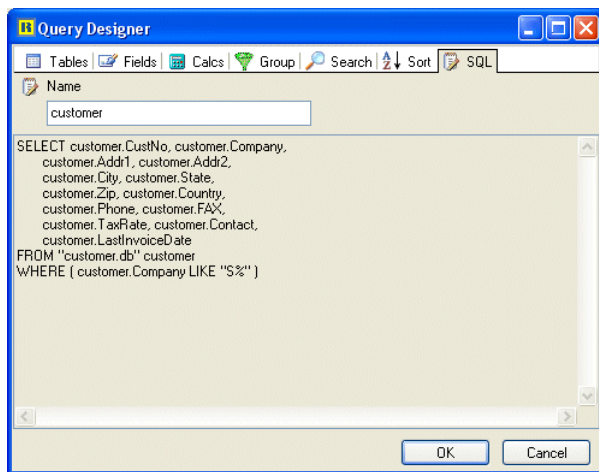


A partir daqui, poderíamos prosseguir para a área de trabalho de design, onde poderíamos criar um novo layout de relatório com base nessa visualização de dados manualmente ou por meio do uso do Assistente de Relatório.

Designer de Consulta

Visão geral

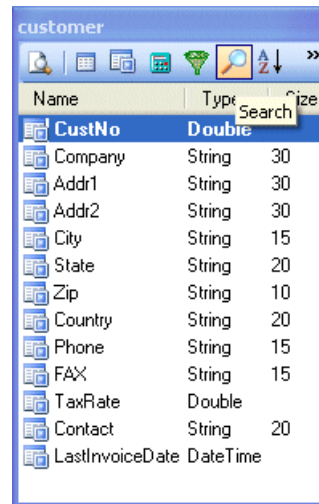
O Query Designer é usado para modificar visualizações de dados baseadas em consulta. O designer de consulta apresenta uma série de guias do bloco de notas; cada guia representa uma parte diferente da consulta. A última guia do bloco de notas no Query Designer mostra o SQL gerado e permite que o nome da visualização de dados e do pipeline de dados sejam alterados. O Query Designer é mostrado abaixo.



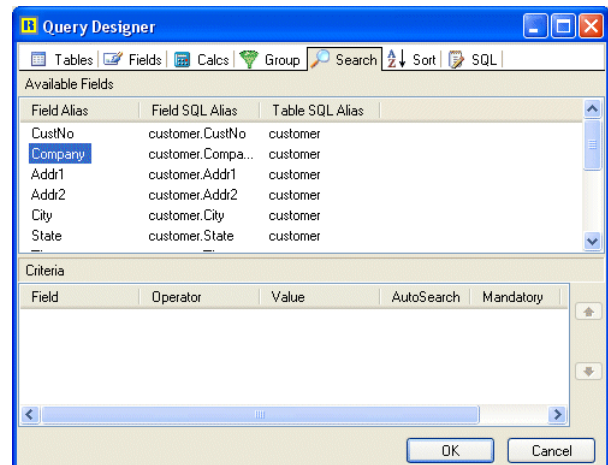
Designer de Consulta: Adicionando Critérios de Pesquisa

Você pode usar o Query Designer para adicionar ou remover critérios de pesquisa de sua consulta. Execute estas etapas para adicionar critérios de pesquisa:

- 1 Clique no ícone Pesquisar da visualização de dados para iniciar o Query Designer.

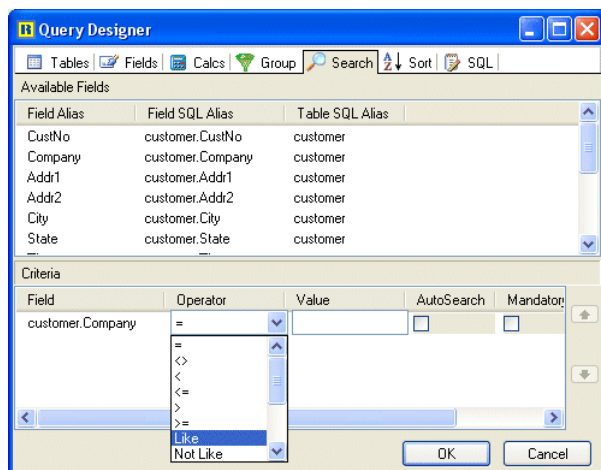


- 2 Na lista de campos na parte superior da página de pesquisa, clique duas vezes no campo para o qual os critérios precisam ser inseridos.

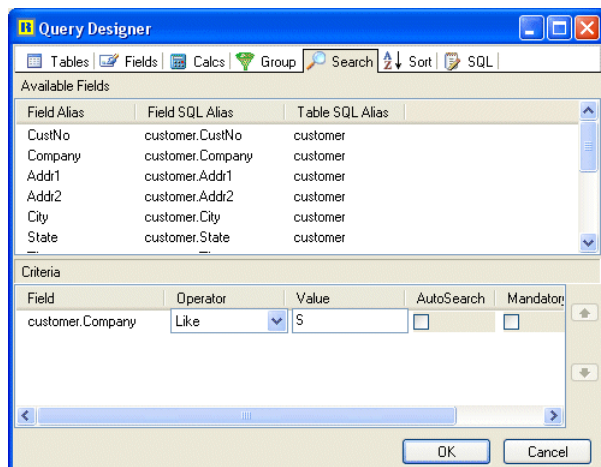


Designer de Consultas: Adicionando Critérios de Pesquisa - cont.

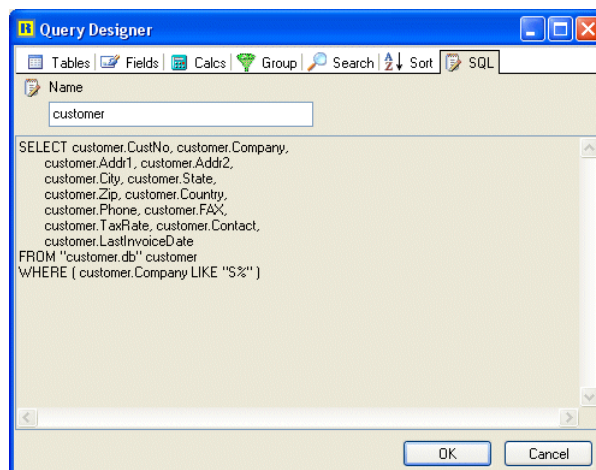
3 Clique no campo que foi adicionado à lista de critérios na parte inferior e selecione o operador.



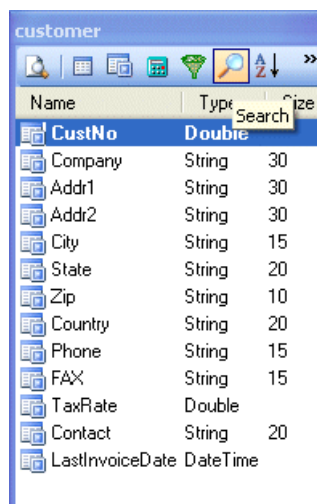
4 Clique na caixa de edição e insira o valor do critério de pesquisa. Este critério encontrará todos os nomes de empresas que começam com a letra 'S'.



5 Clique na guia SQL para certificar-se de que o valor dos critérios é válido.

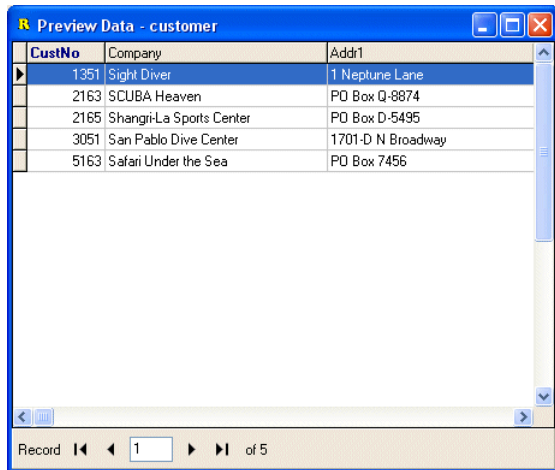


6 Feche o Query Designer e clique no ícone de visualização.



Designer de Consultas: Adicionando Critérios de Pesquisa - cont.

7 Visualize os dados e certifique-se de que os registros pretendidos sejam selecionados.



Criar uma soma de grupo

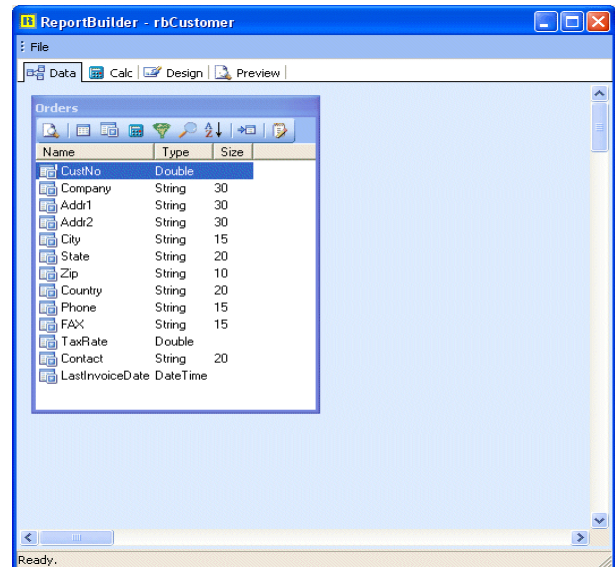
A cláusula SQL 'GROUP BY' permite que você elimine linhas em sua consulta onde os valores do campo se repetem. Por exemplo, vamos supor que temos uma tabela de banco de dados que contém registros de pedidos. Cada registro de pedido contém o número do cliente e o valor pago. Se visualizarmos os dados nesta tabela, veremos que o valor no campo de número do cliente se repete quando há vários pedidos para um cliente.

Podemos usar o SQL para selecionar dados da tabela de pedidos e calcular o valor total pago para cada cliente.

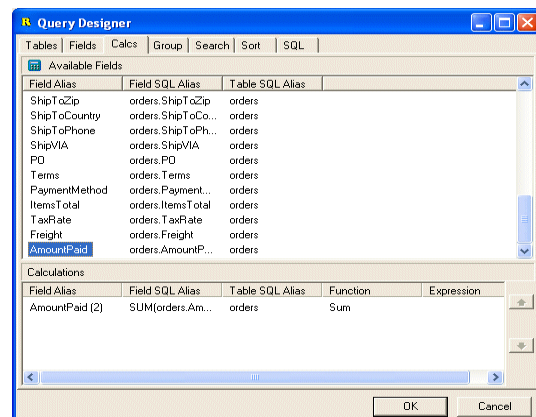
Podemos fazer isso especificando um grupo no campo de número do cliente. Ao especificar o grupo, estamos dizendo ao mecanismo SQL: crie uma linha no conjunto de resultados para cada número de cliente encontrado. Quando o mecanismo SQL executa a consulta, ele encontrará vários registros para alguns clientes; esses registros serão eliminados do conjunto de resultados. O SQL nos permite realizar cálculos nesses registros repetidos e armazenar o resultado em um novo campo do conjunto de resultados.

Esses tipos de cálculos podem ser criados na guia Calc do Query Designer. Execute estas etapas para somar o valor pago por todos os clientes na tabela de pedidos:

1 Clique no ícone 'Calc' para iniciar o Query Designer.

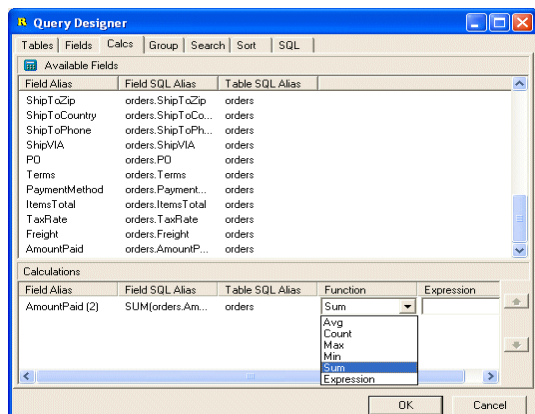


2 Clique duas vezes no campo 'Valor pago' na lista de seleção na parte superior da página. O valor pago será adicionado à lista de cálculos.

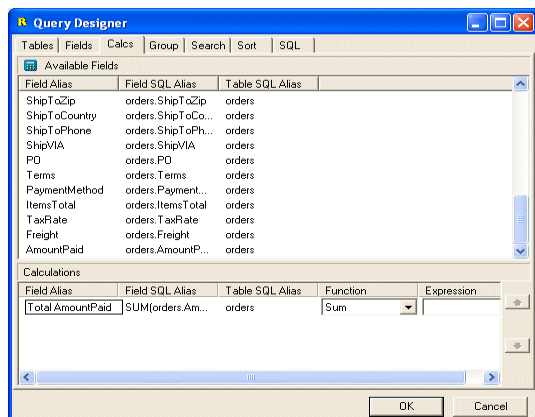


Criar uma soma de grupo - cont.

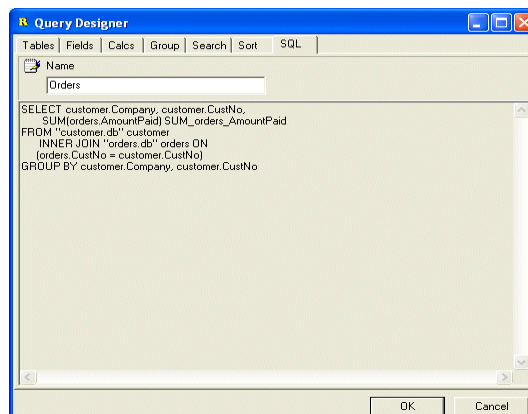
3 Selecione 'Soma' como o tipo de função para o cálculo.



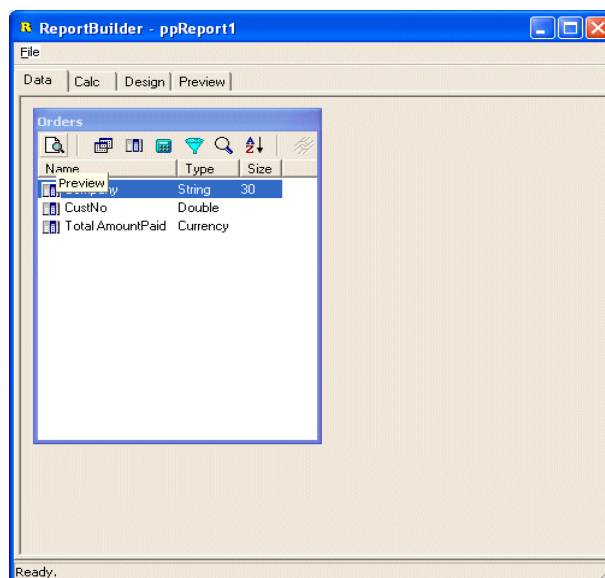
4 Insira o Alias do campo que deseja usar para este campo calculado.



5 Clique na guia SQL para certificar-se de que o SQL gerado é válido.

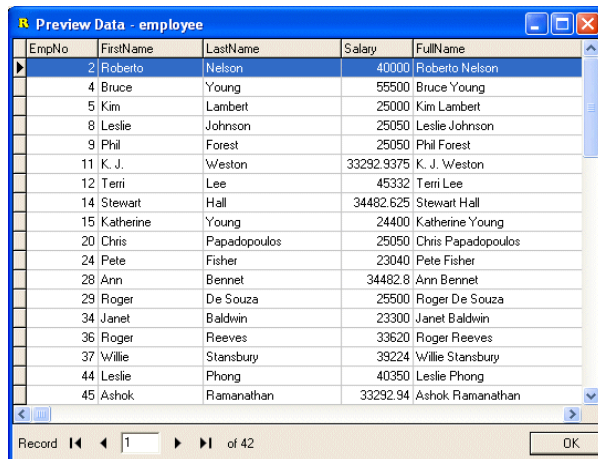


6 Feche o Query Designer e clique no ícone Visualizar para visualizar os dados.



Criar uma soma de grupo - cont.

7 Verifique os dados para garantir que a soma seja calculada conforme o esperado.

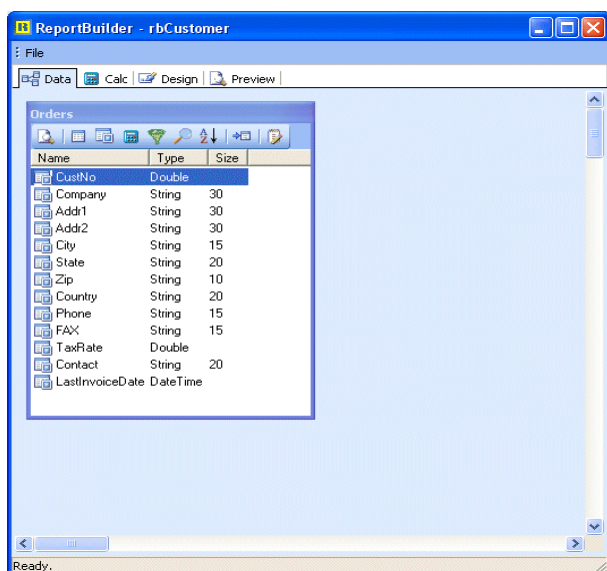


EmpNo	FirstName	LastName	Salary	FullName
2	Roberto	Nelson	40000	Roberto Nelson
4	Bruce	Young	55500	Bruce Young
5	Kim	Lambert	25000	Kim Lambert
8	Leslie	Johnson	25050	Leslie Johnson
9	Phil	Forest	25050	Phil Forest
11	K. J.	Weston	33292.9375	K. J. Weston
12	Terri	Lee	45332	Terri Lee
14	Stewart	Hall	34482.625	Stewart Hall
15	Katherine	Young	24400	Katherine Young
20	Chris	Papadopoulos	25050	Chris Papadopoulos
24	Pete	Fisher	23040	Pete Fisher
28	Ann	Bennet	34482.8	Ann Bennet
29	Roger	De Souza	25500	Roger De Souza
34	Janet	Baldwin	23300	Janet Baldwin
36	Roger	Reeves	33620	Roger Reeves
37	Willie	Stansbury	39224	Willie Stansbury
44	Leslie	Phong	40350	Leslie Phong
45	Ashok	Ramanathan	33292.94	Ashok Ramanathan

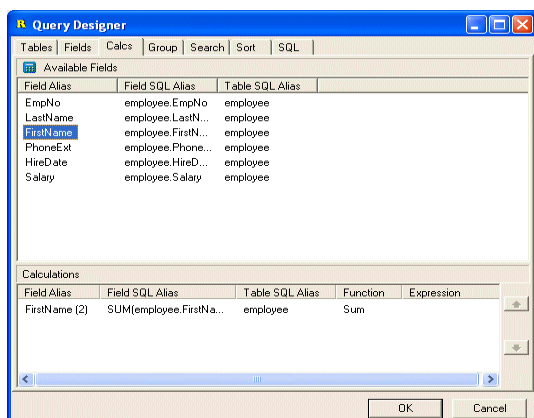
Campos Concatenados

Você pode inserir expressões SQL na guia Calc do Query Designer. A consulta a seguir seleciona dados de uma tabela de funcionários. A tabela possui um campo de nome e sobrenome. Execute essas etapas para concatenar esses dois campos usando o Query Designer:

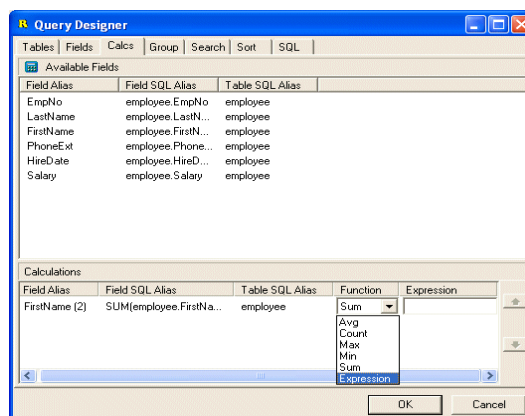
1 Clique no ícone 'Calcs' para iniciar o Query Designer.



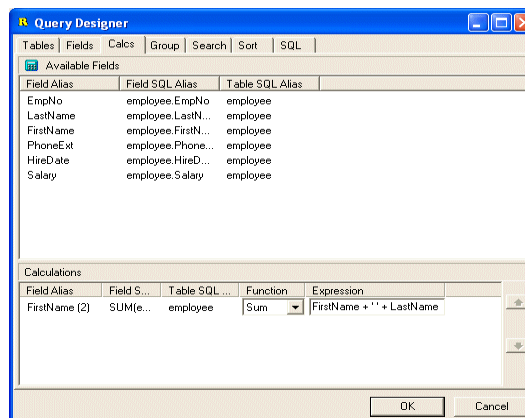
2 Clique duas vezes no campo 'Nome' da lista de seleção no topo da página. 'Nome' será adicionado à lista de cálculos.



3 Selecione 'Expressão' como o tipo de função para o cálculo.

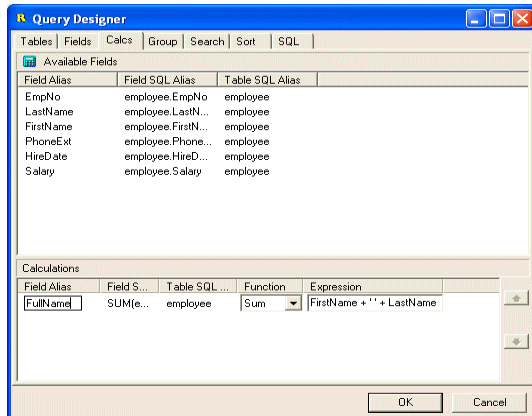


4 Modifique as larguras das colunas 'Field SQL Alias' e 'Table SQL Alias' e 'Expressão' na lista de cálculos na parte inferior da página para que haja espaço suficiente para inserir a expressão. A figura abaixo ilustra um exemplo de expressão.

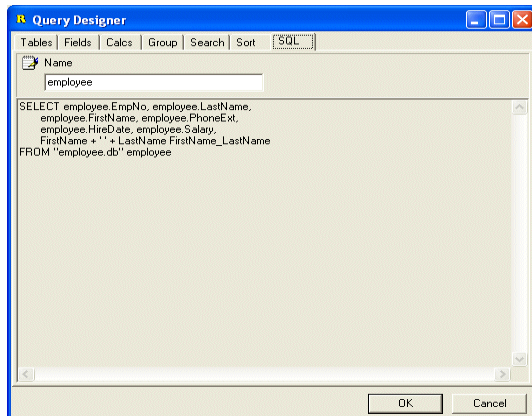


Campos concatenados - cont.

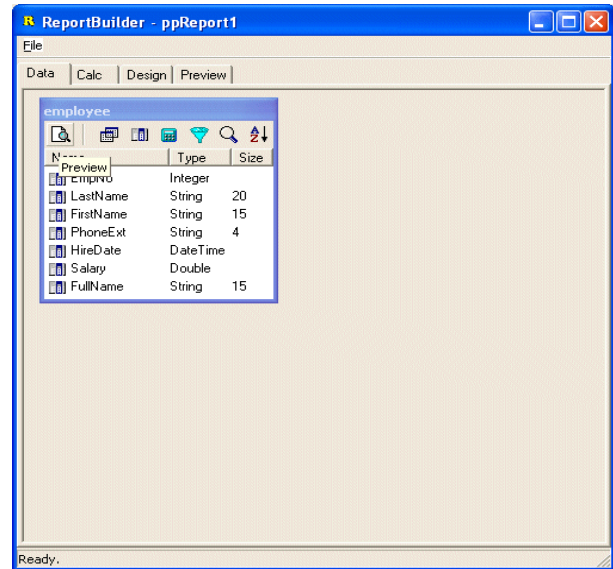
5 Insira o Alias do campo que deseja usar para este campo calculado.



6 Clique na guia SQL para certificar-se de que o SQL gerado é válido.



7 Feche o Query Designer e clique no ícone Visualizar para visualizar os dados.



8 Verifique os dados para ter certeza de que o campo foi calculado conforme o esperado.

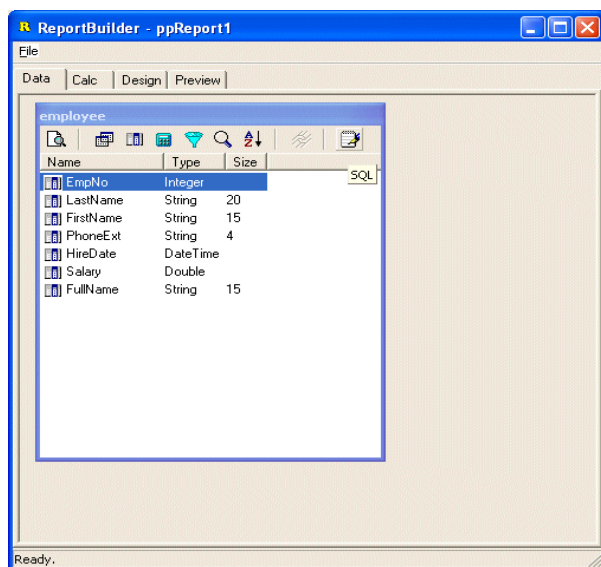


Editar SQL

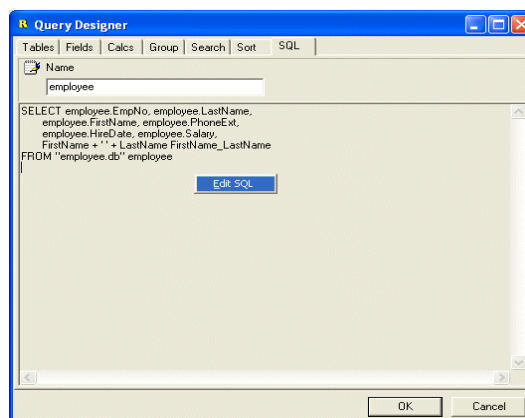
Pode haver momentos em que você precise utilizar recursos avançados de SQL que não podem ser acessados por meio da interface visual do Query Designer. Nesses casos, você pode editar o SQL manualmente no Query Designer. Depois de editar o SQL manualmente, você deve sempre usar a guia SQL do Query Designer para fazer modificações futuras.

Execute estas etapas para editar o SQL gerado pelo Query Designer:

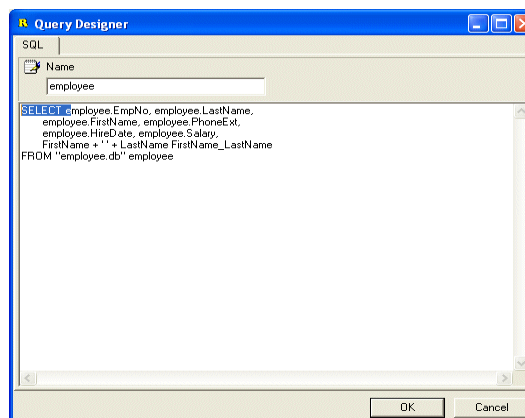
1 Clique no ícone SQL para iniciar o Query Designer.



2 Clique com o botão direito sobre o texto SQL para exibir o menu pop-up.



3 Selecione o item de menu. Clique em Sim na caixa de diálogo da mensagem. Agora você pode editar o SQL.



Configurando DADE

Visão geral

A primeira etapa na configuração do DADE para seus usuários finais envolve a definição das propriedades do objeto `DataSettings` do componente `Designer`. Este objeto contém as seguintes propriedades:

SessionType

Use para especificar o plug-in DADE que está sendo usado. Cada plug-in DADE registra um tipo de sessão correspondente. Por exemplo, se você estiver usando o acesso a dados ADO, escolha a opção `ADOSession`. ReportBuilder instala plug-ins para suportar os componentes de acesso a dados incluídos no Delphi: ADO, BDE, IBExpress, dbExpress. Muitos plug-ins adicionais estão disponíveis. Alguns são instalados em bancos de dados RBuilder \ Demos \ EndUser e outros estão disponíveis em Download | Seção de plug-ins DADE de nosso site.

AllowEditSQL

Determina se o usuário final pode ou não editar o SQL no Query Designer.

Nome do banco de dados

O nome do banco de dados do qual os dados serão recuperados quando a consulta for executada.

Dicionário de dados

O objeto de dicionário de dados que converterá nomes brutos de tabelas e campos em aliases.

UseDataDictionary

Determina se o objeto de dicionário de dados atribuído à propriedade `DataDictionary` será usado.

SQLType

Especifica o tipo de sintaxe SQL gerada: `BDELocal`, `SQL1`, `SQL2`. A opção `BDELocal` pode ser usada para o Paradox. Para a maioria dos produtos de banco de dados modernos, use a configuração `SQL2`.

DatabaseType

Use o tipo de banco de dados para selecionar o produto de banco de dados específico que está sendo usado. Isso refina ainda mais a sintaxe SQL para aderir às variações específicas do produto.

CollationType

Usado para informar ao ReportBuilder que tipo de agrupamento é usado por seu banco de dados: `Ascii`, `Ansi`, `Variant`. Isso é importante para oferecer suporte à passagem de dados mestre / detalhes pelo mecanismo de relatório.

GuidCollationType

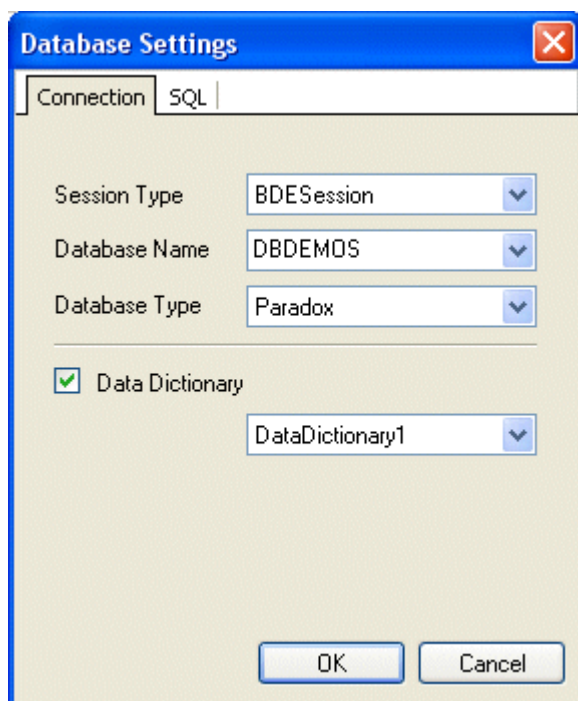
Usado para informar ao ReportBuilder que tipo de agrupamento `Guid` é usado por seu banco de dados: `MSSQLServer`, `MSAccess`, `Guid`, `String`. Isso é importante para suportar a passagem de dados mestre / detalhes quando o relacionamento de vinculação é definido por um campo de dados do tipo `Guid`.

IsCaseSensitive

Usado para informar ao ReportBuilder se seu banco de dados diferencia maiúsculas de minúsculas. Isso é importante para oferecer suporte à passagem de dados mestre / detalhes pelo mecanismo de relatório.

Configurações de dados

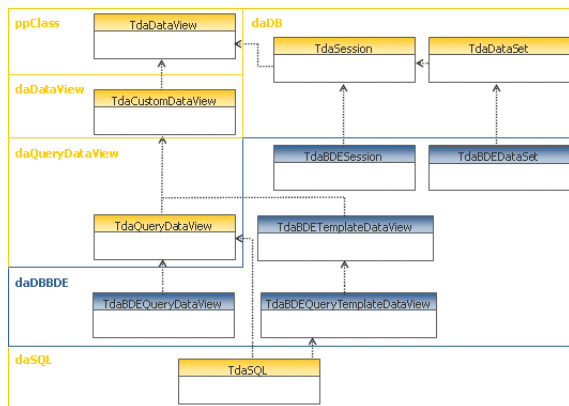
Depois de definir as configurações de dados, você pode notar que uma opção de menu Configurações de dados aparece no menu Arquivo da área de trabalho de dados quando você executa seu aplicativo. Esta opção exibe uma caixa de diálogo que permite ao usuário final modificar as configurações de dados. Se você não quiser que o usuário final altere essas configurações, defina a propriedade AllowDataSettingsChange do componente Designer como False.



Arquitetura DADE

Visão geral

O modelo de objeto contendo as classes mais importantes de DADE é ilustrado abaixo:



Os relacionamentos de herança são indicados por triângulos no diagrama acima. Onde uma classe está associada a outra classe, uma linha é usada. Se a linha terminar em forma de losango, isso indica que a classe realmente faz parte da implementação interna do componente. Por exemplo, a classe **TdaSQL** faz parte da implementação de **TdaQueryDataView**. A unidade em que cada classe é declarada é indicada pelo texto no canto superior esquerdo de cada forma contida.

Classes básicas

O DADE tem uma arquitetura de plug-in que permite que as Ferramentas de Consulta sejam usadas com qualquer descendente de **TDataSet** baseado em SQL. A seguir está uma visão geral das classes básicas que compõem a arquitetura de plug-in DADE. Ao criar descendentes dessas classes, o suporte nativo para produtos de banco de dados SQL pode ser implementado.

TdaDataView

Dataviews representam um conjunto de dados e são visualmente criados e mantidos na área de trabalho de dados do Report Designer. Internamente, uma visualização de dados cria os objetos de acesso a dados necessários para conectar um relatório aos dados. A interface da visualização de dados para as outras áreas de trabalho do Report Designer é composta de pipelines de dados, que estão contidos na visualização de dados. Uma visualização de dados deve ter pelo menos um pipeline de dados. Uma visualização de dados típica conterá um ou mais pipelines de dados, fontes de dados e conjuntos de dados.

TdaDataView contém as declarações dos métodos que os descendentes devem implementar. Está na unidade **ppClass** porque declara uma propriedade **Report** que se refere a **TppCustomReport**. Se você precisa saber o relatório principal ao qual uma visualização de dados está associada, você sempre pode usar esta propriedade.

TdaSession

Contém os métodos necessários para a classe TdaSQL ao editar (por meio do Query Wizard e do Query Designer) e gerar SQL. Fornece os nomes das tabelas para um determinado nome de banco de dados e o nome do driver do banco de dados (ou seja, 'Sybase').

TdaDataSet

Contém métodos necessários para a classe TdaSQL para gerar SQL. É usado para obter os nomes dos campos para um determinado nome de tabela e para verificar a validade das instruções SQL.

TdaSQL

Esta é a principal classe de edição e geração de SQL. O Query Wizard, Query Designer e DataViews contam com este componente. Como o SQL é projetado visualmente, o componente SQL mantém uma descrição da consulta SQL baseada em objeto. Esta descrição é usada para gerar o SQL. O objeto SQL é salvo como parte de visualizações de dados baseadas em consulta.

TdaCustomDataView

Essa classe define ainda mais a API de visualização de dados e implementa a capacidade de salvar a visualização de dados como um conjunto de objetos.

TdaQueryDataView

Esta classe contém a associação à classe TdaSQL.

DADE Plug-in Classes

O DADE tem uma arquitetura de plug-in que permite que as Ferramentas de Consulta sejam usadas com qualquer descendente de TDataSet baseado em SQL. A seguir está uma visão geral das classes que compõem o plug-in ADO. Todos os plug-ins seguem esta mesma implementação de classe.

Sessão TdaADOS

Registra o tipo de sessão ADOSession com DADE. Responsável por recuperar a lista de ADOConnections disponíveis e a lista de nomes de tabela disponíveis para uma conexão especificada.

TdaADODataset

Responsável por recuperar metadados de campo para uma tabela ou consulta SQL especificada.

TdaADOQueryDataView

Gerencia um objeto ADOQuery e TdaSQL. As ferramentas de consulta se comunicam com o TdaSQLObject. O ADOQueryDataView atribuiu a instrução SQL gerada pelo TdaSQLObject ao ADOQuery.

TdaADOTemplateDataView

Esta classe pode ser usada quando modelos de visualização de dados personalizados são necessários. Os modelos de visualização de dados utilizam uma interface de usuário fornecida pelo desenvolvedor para a criação e modificação da visualização de dados. Isso geralmente é feito para substituir o Query Wizard e o Query Designer por uma interface de usuário totalmente personalizável.

TdaADOQueryTemplateDataView

Esta classe pode ser usada quando modelos de visualização de dados personalizados que contêm uma consulta são necessários.

Estendendo DADE

Visão geral

DADE é uma arquitetura flexível e extensível que permite aos desenvolvedores personalizar a funcionalidade entregue para atender aos requisitos de um aplicativo específico. As formas mais comuns em que o DADE é estendido são as seguintes:

1 Modelos de Dataview

Crie modelos de visualização de dados para personalizar o processo de criação de visualização de dados ou para fornecer configurações de pipeline de dados que não podem ser criadas por meio da visualização de dados baseada em consulta integrada.

2 Suporte para produtos de banco de dados

Crie plug-ins DADE que suportem nativamente conjuntos de componentes de acesso a dados adicionais.

3 Novo Assistente de Dados

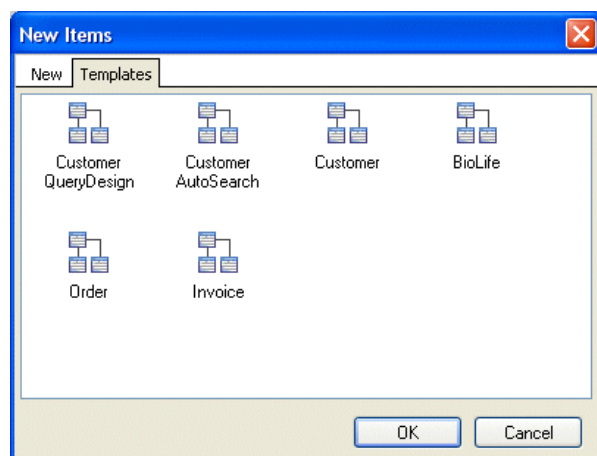
Crie um novo assistente de dados.

Modelos de Dataview

Um modelo de visualização de dados é um tipo especial de classe de visualização de dados que foi declarado por você, o desenvolvedor. E embora a representação visual da visualização de dados no espaço de trabalho de dados permaneça a mesma de qualquer visualização de dados, a interface do usuário usada para criar e modificar a visualização de dados é totalmente dependente do que você deseja fornecer aos usuários finais. Os modelos do Dataview podem ser usados para qualquer coisa, desde a criação de relações predefinidas entre um conjunto de tabelas até a integração de um construtor de consultas completo. Nesta seção, mostraremos como você pode criar um modelo de visualização de dados. Mas antes de fazermos isso, vamos dar uma olhada no que o usuário final verá quando usar um modelo de visualização de dados.

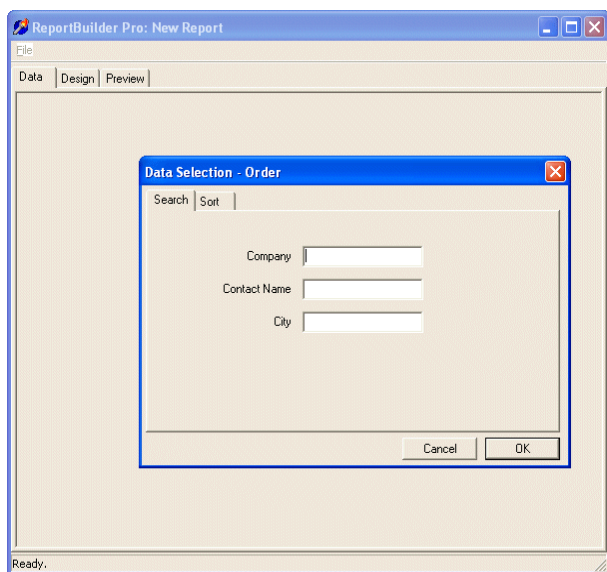
Modelo Dataview: A visão do usuário final

1 Quando os modelos de visualização de dados são registrados no DADE, uma guia adicional é adicionada à caixa de diálogo Novos itens. A figura abaixo mostra o que veríamos se acessássemos o arquivo | Nova opção de menu no espaço de trabalho de dados:

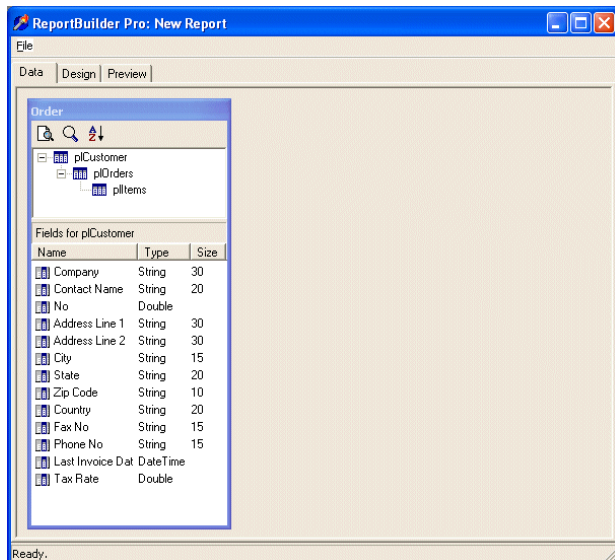


Modelo de Dataview: A visão do usuário final - cont.

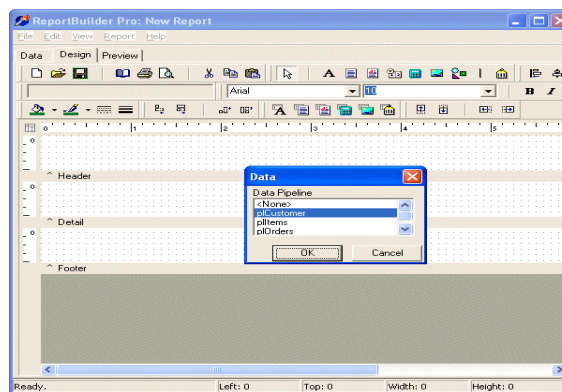
2 Aqui podemos ver que seis modelos de visualização de dados diferentes foram registrados. Após clicar duas vezes no modelo 'Pedido', a seguinte tela iria aparecer:



3 Esta é uma caixa de diálogo personalizada criada pelo desenvolvedor do modelo de visualização de dados do pedido. Ele permite que o usuário final insira critérios de pesquisa e classificação simples. Depois de fazer isso e clicar em OK, a visualização de dados concluída será exibida:



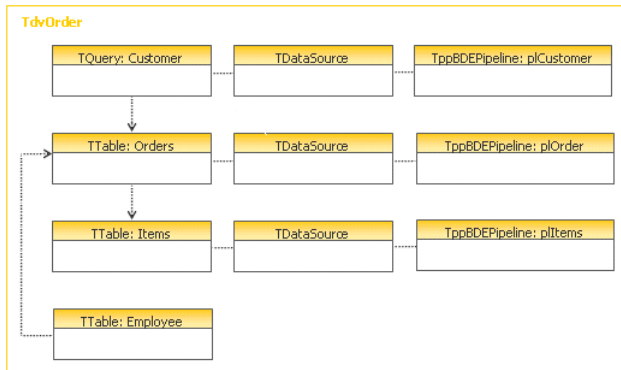
4 Como você pode ver na lista de pipeline de dados desta visualização de dados, este modelo declarou relacionamentos entre as tabelas de cliente, pedido e itens de pedido. Cada tabela é independente e tem seu próprio pipeline de dados. Clicando na aba Design e acessando o Relatório | Na opção de menu de dados, podemos ver como esta visualização de dados atribuiu os pipelines de dados ao relatório:



O pipeline de dados do cliente é atribuído ao relatório. Os pipelines de dados de pedidos e itens estão disponíveis e podemos criar sub-relatórios e atribuí-los a esses pipelines de dados. Esse tipo de configuração de pipeline de dados não está disponível em visualizações de dados baseadas em consulta, e é por isso que o desenvolvedor escolheu criar este modelo de visualização de dados.

Modelo Dataview: A implementação

Este modelo de visualização de dados é implementado como uma consulta SQL na tabela do cliente com tabelas de detalhes aninhadas nas tabelas de pedidos e itens. Alguns dos campos da tabela de detalhes do pedido são, na verdade, fornecidos pela tabela de funcionários, que é configurada como uma tabela de consulta:



Uma versão completa e funcional do modelo de visualização de dados do pedido está disponível no

\ Demos \ EndUser \ 2. Custom DataView Templates \
diretório dentro do diretório ReportBuilder principal.

Suporte para produtos de banco de dados

O DADE é construído com base no conceito de visualização de dados; portanto, é importante conhecer as seguintes características da visualização de dados:

- Uma visualização de dados é responsável por criar os objetos de acesso a dados necessários para conectar um relatório aos dados.
- Uma visualização de dados deve conter pelo menos um pipeline de dados.
- Uma visualização de dados típica conterá um ou mais pipelines de dados, fontes de dados e conjuntos de dados.
- Dataviews são criados e mantidos visualmente no espaço de trabalho de dados do designer de relatórios.

Implementando novos descendentes de várias classes dentro do DADE, podemos integrar o suporte nativo para produtos de banco de dados adicionais no DADE para que as ferramentas visuais fornecidas no espaço de trabalho de dados (ou seja, o Query Wizard, Query Designer e interfaces de usuário de visualização de dados personalizadas) possam ser usadas com esse banco de dados.

Durante a operação normal, o Query Wizard e o Query Designer requerem certas informações do banco de dados, como uma lista de tabelas a partir das quais os dados podem ser selecionados ou uma lista de campos para uma determinada tabela. Essas ferramentas visuais nunca fazem referência direta a nenhum objeto de banco de dados na obtenção dessas informações, mas, em vez disso, contam com a funcionalidade fornecida pela visualização de dados.

Aulas

A implementação de um descendente QueryDataView exige que definamos o seguinte conjunto de classes:

TdaChildADOQuery

- Descendente de TADOQuery que pode ser filho de um DataView
- Substitua o método HasParent de Tcomponent para retornar True
- Deve ser registrado com Delphi IDE via RegisterNoIcon

TdaChildADOTable

- Descendente de TADOTable que pode ser filho de um DataView
- Substitua o método HasParent de Tcomponent para retornar True
- Deve ser registrado com Delphi IDE via RegisterNoIcon

Sessão TdaADOS

- Descendente de TppSession
- Implementa GetDatabaseNames
- GetTableNames, etc.

TdaADODataset

- Descendente de TppDataSet
- Implementa GetFieldNames para SQL

TdaADOQueryDataView

- Descendente de TppQueryDataView
- Usa as classes acima para criar o Query DataSource Pipeline Report necessário conexão
- Usa o objeto TdaSQL construído pelo QueryWizard para atribuir SQL ao ADOQuery, etc.

Plug-ins DADE:

Se você tiver o Delphi Enterprise, poderá ver essas classes em ação instalando o demo ADO. Os seguintes plug-ins DADE são fornecidos com o ReportBuilder Enterprise. Os exemplos podem ser encontrados em \RBuilder \ Demos \ EndUser Databases.

• ALVOROÇO

- IBX
- IBO
- Vantagem
- DBISAM

CTRIBUTO

Introdução 195

O Calc Workspace 197

Escrevendo Código RAP

205 Configurando RAP 209

Estendendo RAP 213

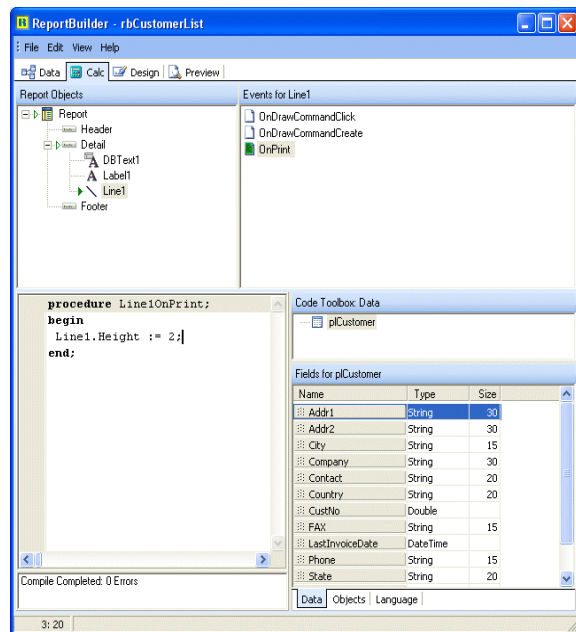
Depurando Código RAP 215

CTRIBUTO

Introdução

Visão geral

RAP (Report Application Pascal) e o ambiente de desenvolvimento integrado que pode ser usado para criar programas RAP são o resultado da descoberta de que uma solução visual para processamento e cálculo de dados era necessária para os usuários finais. Mesmo que o RAP consista em um compilador orientado a objetos e arquitetura extensível "nos bastidores", ele aparece como um espaço de trabalho calc simples e fácil de usar dentro do Report Designer. O espaço de trabalho calc é ilustrado abaixo. A janela do editor de código na parte inferior da área de trabalho mostra um manipulador de eventos simples atribuído a um componente de linha.



A linguagem Report Application Programming (ou RAP para abreviar) foi projetada para permitir que desenvolvedores e usuários finais codifiquem cálculos, manipuladores de eventos e procedimentos autônomos para uso com ReportBuilder Enterprise. Os programas RAP podem ser criados, modificados, compilados e executados em tempo de execução.

Sobre RAP

RAP é fácil de aprender.

A linguagem RAP é idêntica ao Object Pascal. Se você sabe como codificar manipuladores de eventos Delphi, sabe como codificar RAP; portanto, a curva de aprendizado para desenvolvedores Delphi é mínima.

O RAP permite trabalhar com objetos e propriedades de objetos.

O RAP fornece acesso total ao relatório, bandas, grupos, componentes de relatório, pipelines de dados e quaisquer outros objetos que você deseja repassar ao usuário final. Novos objetos podem ser criados. As propriedades publicadas e públicas dos objetos são acessíveis, de modo que os usuários finais podem configurar os objetos imediatamente.

RAP é portátil.

Depois de criar um conjunto de manipuladores de eventos ou procedimentos autônomos, você pode salvá-los como parte da definição do relatório. Você pode exportar e importar cálculos de um relatório para outro. Você pode até salvar cálculos em um arquivo ou campo BLOB de banco de dados. E tudo pode ser feito em tempo de execução, sem recompilar seu aplicativo.

O RAP possui uma interface de usuário escalonável.

- Para usuários finais casuais, uma caixa de diálogo Cálculos simples pode ser acessada no menu de velocidade do novo componente TppVariable. Esta caixa de diálogo fornece um local para os cálculos serem inseridos e retornados por meio de um único parâmetro Result. O resultado do cálculo é então exibido na variável quando o relatório é impresso.
- Para usuários mais sofisticados, o RAP pode ser configurado para ser exibido como uma guia adicional no Report Designer. A nova guia 'Calc' mostra as bandas do relatório em uma visualização em árvore, junto com as variáveis associadas. O usuário final pode clicar nas variáveis e codificar o cálculo em uma janela de edição sensível à sintaxe.
- O nível mais alto de funcionalidade é fornecido pela visualização 'Evento'. Este recurso mostra todos os componentes do relatório em uma exibição em árvore. Quando um componente é selecionado, todos os eventos do componente são exibidos. O usuário pode então selecionar um evento e codificar um manipulador de eventos.

RAP é extensível.

O RAP é fornecido como padrão com a capacidade de compilar os objetos ReportBuilder e Delphi usados com mais frequência. Se quiser que os usuários finais tenham acesso a mais do que isso, você pode registrar informações RTTI adicionais com o RAP. O RAP poderá então compilar seus componentes personalizados. Você também pode estender a linguagem adicionando novas funções e procedimentos integrados que os usuários podem chamar de seus programas RAP. Esses procedimentos são escritos e compilados em Object Pascal e então registrados com RAP.

RAP é otimizado.

RAP é uma linguagem complementar para ReportBuilder Enterprise. No entanto, o ReportBuilder Enterprise não requer RAP para compilar. Você pode instalar o RAP e ele aparecerá automaticamente no Report Designer. As unidades Delphi que compõem o RAP somente serão adicionadas ao seu executável se você usar o RAP em seu relatório.

RAP é legal.

O amplo conjunto de recursos, a funcionalidade baseada em objetos e a interface de usuário profissional tornam o RAP a linguagem a ser batida para aplicativos de relatório.

O Calc Workspace

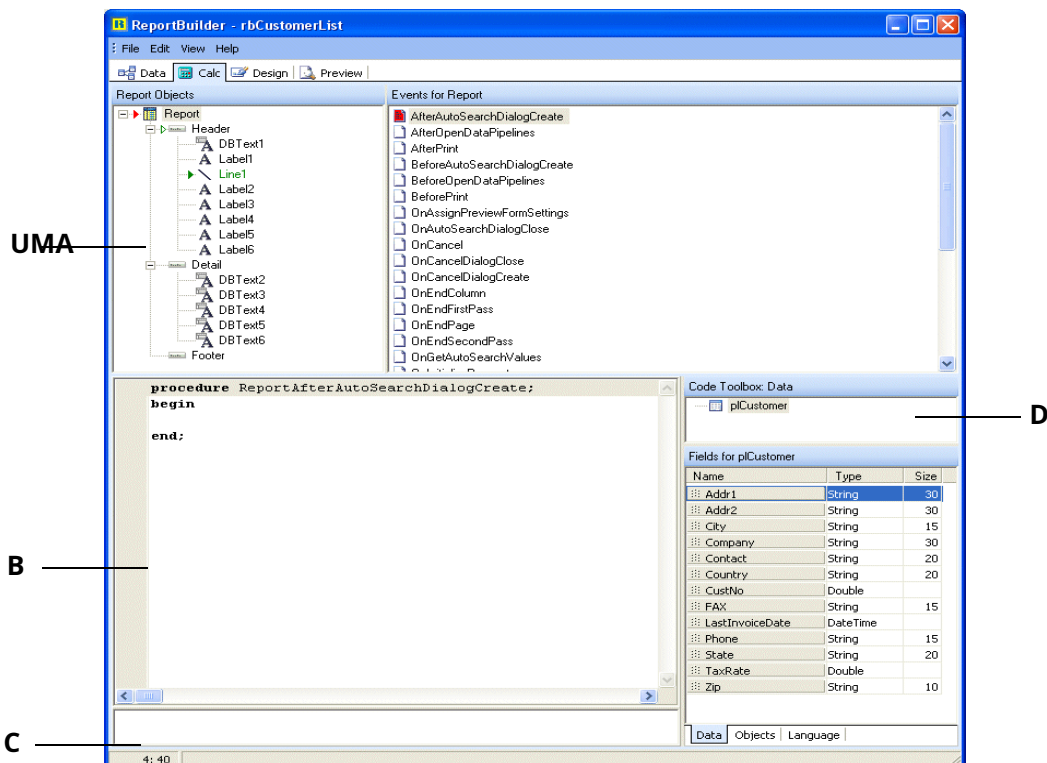
Visão geral

O espaço de trabalho do Calc é o ambiente de desenvolvimento para RAP. Com todas as opções RAP disponíveis, o espaço de trabalho Calc é um ambiente de desenvolvimento Pascal poderoso. Para a maioria dos usuários finais, isso pode ser mais do que precisam. Julgando as necessidades de seus usuários, você deve definir quanto do RAP IDE disponibilizar para eles.

Você pode incluir o espaço de trabalho Calc em seus projetos de usuário final adicionando raIDE à cláusula de uso de uma de suas unidades. Quando seus usuários selecionarem a guia Calc, eles verão o IDE RAP. O RAP IDE consiste em:

- O Code Explorer
- O Editor de Código
- A janela de mensagem
- A caixa de ferramentas de código

O IDE RAP



A Code Explorer: contém a visualização em árvore (esquerda) e a visualização de lista (direita)

B Editor de código: o espaço onde você codifica e compila o relatório

C Janela de mensagem: as mensagens do compilador são apresentadas aqui

D A caixa de ferramentas de código: contém trechos de código que são arrastados para o Editor de código

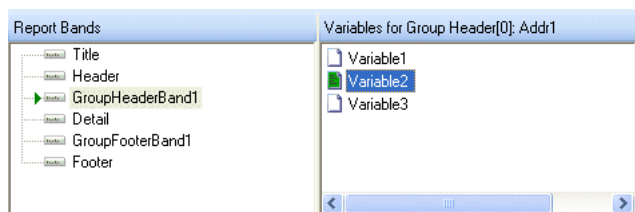
O Code Explorer

O Code Explorer está contido nos painéis superiores esquerdo e direito da área de trabalho do Calc.

O painel esquerdo contém uma visualização em árvore - use-a para navegar pelo código do seu relatório. O painel direito contém uma exibição de lista - ele exibirá uma variedade de itens dependendo do que for selecionado na exibição em árvore. Ao clicar com o botão direito na árvore, você pode exibir um menu de contexto que permite controlar o comportamento do Code Explorer.

Visualização de Variáveis

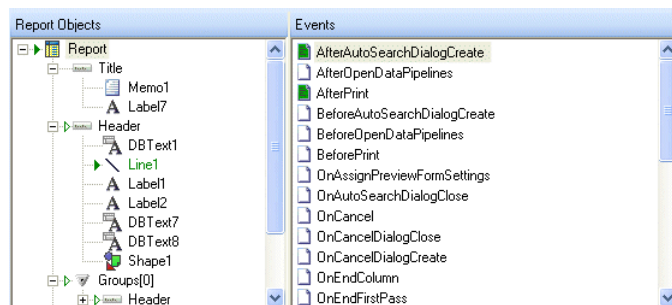
A visualização Variáveis do Code Explorer é exibida clicando com o botão direito do mouse no painel esquerdo e selecionando Variáveis no menu de contexto.



A visualização Variáveis do Code Explorer é exibida clicando com o botão direito do mouse no painel esquerdo e selecionando Variáveis no menu de contexto.

Visualização de eventos

A visualização Eventos do Code Explorer é exibida clicando com o botão direito do mouse no painel esquerdo e selecionando Eventos no menu de contexto.



Esta visualização exibe uma lista de todos os componentes contidos no relatório. O painel direito exibe todos os eventos associados ao componente atualmente selecionado. Selecionar um evento exibirá o manipulador de eventos, se houver.

Esta visão é boa para ver todos os objetos de relatório e seus eventos.

Na visualização em árvore, você notará pequenas imagens em forma de seta à esquerda de alguns nós. Estes são indicadores de estado de compilação. Eles são usados para dizer onde seu código está e em que estado ele está. Existem cinco indicadores possíveis:

Sem símbolo. Indica que este componente não possui manipuladores de eventos atribuídos, nem contém componentes que possuam manipuladores de eventos.

▶ Vermelho, não preenchido. Indica que este componente não possui manipuladores de eventos, mas contém componentes que os possuem. A cor vermelha indica que um ou mais dos componentes aninhados possuem manipuladores de eventos que não compilam.

▶ Vermelho, cheio. Indica que este componente contém manipuladores de eventos e que em algum lugar nesta ramificação há código que não é compilado. Se não houver código contido nos componentes abaixo deste, o código do problema está neste componente. No entanto, se houver código abaixo desse componente, o problema pode estar nos manipuladores de eventos de um componente filho, nos manipuladores de eventos deste componente ou em ambos.

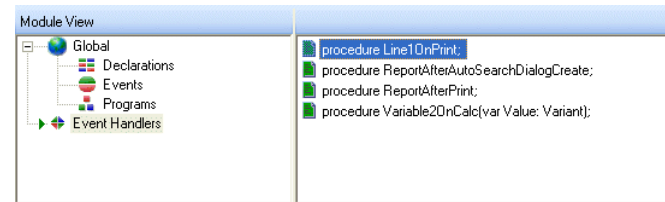
Se o código de um componente filho não compilar, o componente pai ainda exibirá uma seta vermelha, embora seu próprio código possa ser compilado.

▶ Verde, não preenchido. Indica que este componente não possui manipuladores de eventos atribuídos, mas contém outros componentes que os possuem. A cor verde indica que os manipuladores de eventos dos componentes contidos foram compilados com êxito.

▶ Verde, cheio. Indica que este componente possui manipuladores de eventos atribuídos. A cor verde indica que esses manipuladores de eventos, e quaisquer outros atribuídos a componentes aninhados neste, foram compilados com êxito.

Visão do Módulo

A visualização do Módulo do Code Explorer é exibida clicando com o botão direito do mouse no painel esquerdo e selecionando Módulo no menu de contexto.



Esta visualização exibe itens que são visíveis a todos os manipuladores de eventos do relatório:

- **Declarações** - Essas são variáveis e constantes que são visíveis globalmente em todo o relatório.
- **Eventos** - São, em essência, os eventos do relatório. OnCreate e OnDestroy são bons lugares para código de inicialização e finalização, como criar e liberar objetos e inicializar variáveis.
- **Programas** - São procedimentos e funções visíveis globalmente em todo o relatório e, portanto, podem ser chamados de qualquer manipulador de eventos.
- **Manipuladores de eventos** - Esses são todos os manipuladores de eventos que foram implementados no relatório.

Escopo Global

Os itens declarados na seção Global são efetivamente visíveis em todo o módulo e para todos os sub-relatórios filhos. Ao compilar, o compilador tenta resolver todas as referências no módulo atual. Se não conseguir encontrar um identificador, ele começa a procurar na próxima seção global do relatório pai e continua até o relatório principal.

Considere o seguinte cenário envolvendo sub-relatórios de vários níveis:

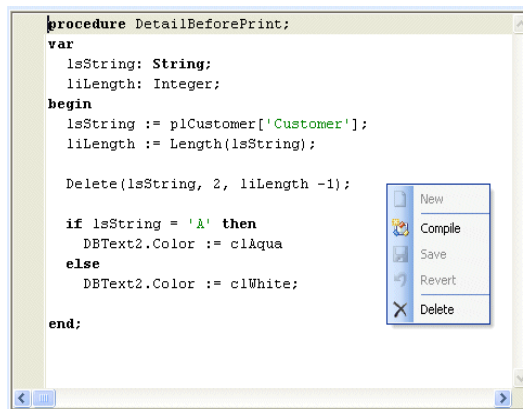
Suponha que declaremos, na seção Global do relatório principal, uma variável, `MyGlobalString: String`; Essa variável seria visível em todo o relatório principal e em todos os sub-relatórios, pois o compilador acabará por encontrar a declaração no pai de qualquer sub-relatório.

No entanto, suponha, novamente, que declaremos `MyGlobalString` na seção Global de `sbrOrderDetail`. Agora a variável está visível para `sbrOrderDetail` e `SubReport1`, mas não visível para `sbrVendorSummary`, `SubReport2` ou mesmo o relatório principal.

Lembre-se: as declarações são visíveis para qualquer sub-relatório filho do relatório atual.

O Editor de Código

O Editor de código é o lugar onde você realmente escreve e modifica o código RAP.



Este editor sensível à sintaxe é semelhante ao encontrado no Delphi, com a exceção de que exibe apenas um procedimento por vez. Qualquer que seja o item selecionado atualmente (manipulador de eventos, procedimento, função, declarações de variáveis, etc.), apenas o código para aquele item é exibido no Editor de Código. Quando um item é selecionado, o editor conterá a implementação do código ou ficará em branco se não houver implementação. Se não houver implementação, você pode criar uma clicando no editor. Você pode inserir o código digitando ou arrastando itens da Caixa de ferramentas de código e soltando-os no editor. Se um item for colocado no editor sobre uma seção do código selecionado, o código selecionado será substituído.

O menu de contexto do Editor de código contém os seguintes itens:

Novo

Novo tem o mesmo efeito que clicar no Editor de código. Só é habilitado se não houver implementação para o item atualmente selecionado no Code Explorer.

Compilar

Compile ativa o compilador RAP para tentar compilar o procedimento atual e quaisquer procedimentos dos quais o atual dependa.

Salve

O espaço de trabalho do Calc mantém um buffer intermediário para o Editor de código. Selecionar Salvar irá comprometer o conteúdo atual do Editor de Código para o buffer; ele não salvará o relatório inteiro. Selecionar Salvar tem o mesmo efeito que navegar para longe e retornar ao procedimento atual.

Reverter

Use Reverter para substituir o conteúdo do Editor de código pelo que está atualmente contido no buffer de código. Isso tem o efeito de remover todas as alterações desde o último salvamento.

Excluir

Selecione Excluir para remover todo o procedimento atual.

Janela de Mensagem

A janela de mensagens funciona essencialmente da mesma maneira que a janela de mensagens do Delphi. As mensagens do compilador são apresentadas aqui. Você pode navegar até o local dos erros do compilador clicando duas vezes na mensagem de erro.

A caixa de ferramentas de código

O Code Toolbox é um repositório de código visual. Ele contém a maioria dos identificadores e elementos de código que o compilador RAP reconhece.

A caixa de ferramentas de código permite que você:

- Exibir identificadores agrupados por dados disponíveis, propriedades de objetos visíveis ou recursos de linguagem
- Gerar código arrastando identificadores para o Editor de código. As funções arrastadas para o editor irão gerar uma lista de parâmetros de espaço reservado. Por exemplo, o código a seguir é gerado quando você arrasta a função Copiar para o Editor de Código:

Cópia (S, Índice, Contagem);

Cada guia da Caixa de ferramentas de código consiste em uma visualização em árvore e uma lista de identificadores. A visualização em árvore permite que você navegue pelos grupos de identificadores listados em cada guia.

Guia de Dados

A guia Dados da Caixa de ferramentas de código exibe pipelines e campos de dados, permitindo que você arraste e solte referências de campo no Editor de código.

Fields for plCustomer		
Name	Type	Size
Addr1	String	30
Addr2	String	30
City	String	15
Company	String	30
Contact	String	20
Country	String	20
CustNo	Double	
FAX	String	15
LastInvoiceDate	DateTime	
Phone	String	15
State	String	20
TaxRate	Double	

Selecionar um pipeline da lista exibirá todos os campos nesse pipeline, bem como o tipo de dados e as informações de tamanho dos campos.

Para inserir um valor de campo na janela de edição de código, selecione o campo e arraste-o para o Editor de Código. O código necessário para recuperar o valor do campo do pipeline será gerado. Por exemplo, arrastar o campo 'Cidade' da caixa de ferramentas de código ilustrada acima resultaria neste código:

Valor: = Clientes ['Cidade']

Guia de Objetos

A guia Objetos da Caixa de ferramentas de código exibe objetos de relatório e suas propriedades, permitindo que você arraste e solte propriedades no Editor de código.

Properties for Label1		
Name	Type	Value
AnchorBottom	Boolean	False
AnchorLeft	Boolean	False
AnchorLeftBottom	Boolean	False
AnchorLeftTop	Boolean	True
AnchorNone	Boolean	False
AnchorRight	Boolean	False
AnchorRightBottom	Boolean	False
AnchorRightTop	Boolean	False
AnchorTop	Boolean	False
Angle	Integer	0
Assign	Method	procedure Assign
AutoSize	Boolean	True

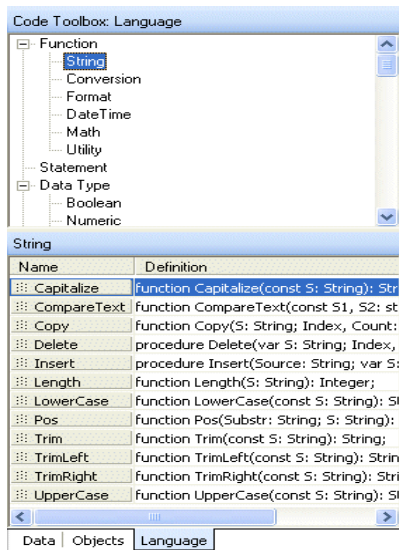
Selecionar um objeto da árvore exibirá uma lista das propriedades desse objeto.

Para inserir uma propriedade no Editor de Código, selecione a propriedade e arraste-a para o Editor de Código. Será gerado o código necessário para recuperar o valor da propriedade ou chamar o método. Por exemplo, arrastar a propriedade 'AutoSize' da caixa de ferramentas de código ilustrada acima resultaria no seguinte código:

Label1.AutoSize

Guia de idioma

A guia Idioma da caixa de ferramentas de código exibe elementos de linguagem RAP, permitindo que você arraste e solte elementos no Editor de código.



Selecionar uma categoria da árvore exibirá uma lista de elementos para essa categoria.

Para inserir um elemento no Editor de código, selecione o elemento e arraste-o para o Editor de código. O código necessário para fazer referência ou usar o elemento será gerado. Observe que, quando você descarta um item, como uma chamada de função, a lista de parâmetros da função é fornecida. Por exemplo, se você arrastar Copiar para o Editor de código, ele se expandirá como:

Cópia (S, Índice, Contagem);

Quando você registra as funções de passagem com o RAP, elas aparecem automaticamente na seção Função da guia Idioma. Os identificadores listados na caixa de ferramentas também exibirão informações relevantes, como tipo, tamanho, assinatura, etc. Os identificadores que aparecem na caixa de ferramentas de código são registrados com RAP por meio da arquitetura extensível de RAP. Você pode usar essa arquitetura para registrar suas próprias funções de passagem e objetos com RAP - isso permite que suas novas funções apareçam na caixa de ferramentas de código e sejam reconhecidas pelo compilador.

Escrevendo código RAP

Programas em RAP

Todo código em um relatório está contido no Módulo de Código. O Módulo de Código contém todos os manipuladores de eventos para objetos contidos no Relatório. Há também uma seção global que pode conter eventos, declarações, procedimentos e funções de nível de módulo. Esta parte do Módulo de Código é visível na visualização Módulo do Explorador de Código.

Os programas e variáveis declarados na seção global de um módulo são visíveis para todos os módulos de código de sub-relatório no relatório. Os manipuladores de eventos aparecem da mesma forma que no Delphi, com a exceção de que o parâmetro Sender é omitido da assinatura - isso se deve ao fato de que os manipuladores de eventos só podem ser atribuídos a um objeto no RAP, enquanto o Delphi permite que o mesmo manipulador de eventos ser atribuído a vários eventos.

Codificando um manipulador de eventos

Para codificar um novo manipulador de eventos, basta selecionar um evento no Code Explorer e clicar no espaço em branco do Editor de Código. Tanto a assinatura quanto o par início / fim para o manipulador de eventos serão gerados automaticamente. Você pode então começar a inserir seu código. Um manipulador de eventos típico apareceria como:

```
procedimento Variable1OnCalc (var Valor: Variante);  
  começar  
  fim;
```

Nota: Quando você está trabalhando na visualização Variáveis do Code Explorer, a assinatura do manipulador de eventos e o par de início / fim não aparecem no Editor de Código. Você verá apenas uma única linha para atribuir o valor:

Valor: =

Neste modo, o Code Explorer exibe apenas variáveis e está, de certo modo, igualando as variáveis ao valor retornado pelo evento OnCalc. Isso é valioso para limitar a quantidade de funcionalidade que você deseja revelar aos usuários.

Compilando manipuladores de eventos

O compilador RAP tenta a compilação de todo o módulo automaticamente quando ocorre uma das seguintes situações:

- Você carrega um relatório.
- Você seleciona a guia Calc no Report Designer.
- Você alterna as visualizações no Code Explorer.

Também é possível compilar o manipulador de eventos atualmente selecionado. Para fazer isso, clique com o botão direito do mouse no Editor de código e selecione Compilar no menu pop-up. O compilador verificará as seções globais e quaisquer outros programas necessários para compilar o manipulador de eventos atual.

Declarando Variáveis Locais

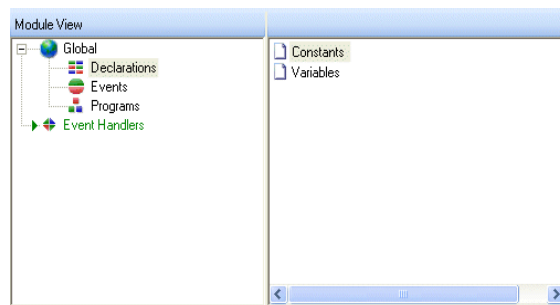
As declarações de variáveis locais em RAP são exatamente as mesmas que em Object Pascal. Para adicionar uma declaração de variável local a uma função, ative o Editor de Código para o item atual e coloque o cursor entre a declaração da função e o início. Digite var e declare suas variáveis exatamente como faria no Delphi.

Declaração de constantes locais

As declarações de constantes locais em RAP são iguais às de Object Pascal. Para adicionar uma declaração de constante local a uma função, ative o Editor de Código para o item atual e coloque o cursor entre a declaração da função e o início. Digite const e declare suas constantes da mesma forma que faria no Delphi.

Declarando Variáveis Globais

Para declarar uma variável global, clique com o botão direito no Code Explorer e selecione Módulo. A visualização em árvore mudará para exibir a seção global. Clique no nó Declarações - isso exibirá dois itens na lista: Constantes e Variáveis.



Selecione o item Variáveis. Se você já declarou algumas variáveis, elas serão exibidas no Editor de Código. Se o Editor de Código estiver em branco, clique com o botão direito do mouse no item Variáveis e selecione Novo - isso ativará o Editor de Código e adicionará **var** para a primeira linha.

Declare suas variáveis usando a sintaxe Object Pascal padrão.

var

```
myGlobalString: String;
myGlobalInteger: Integer;
myStringList: TStringList;
```

Declaração de constantes globais

Para declarar uma constante global, clique com o botão direito no Code Explorer e selecione Módulo. A visualização em árvore mudará para exibir a seção global. Clique no nó Declarações - isso exibirá dois itens na lista: Constantes e Variáveis.

Selecione o item Constantes. Se você já declarou algumas constantes, elas serão exibidas no Editor de Código. Se o Editor de código estiver em branco, clique com o botão direito do mouse no item Constantes e selecione Novo - isso ativará o Editor de código e adicionará `const` à primeira linha.

Declare suas constantes usando a sintaxe Object Pascal padrão.

const

```
csMyConstString = 'O que é moderno?';  
ciMyBigAnswer = 42;
```

Declaração de procedimentos e funções globais

A seção global do Módulo de Código pode conter funções visíveis em todo o módulo e para quaisquer sub-relatórios abaixo do relatório atual. Para declarar tais funções, clique com o botão direito no Code Explorer e selecione Módulo. A visualização em árvore mudará para exibir a seção global. Clique no nó Programas - a visualização em lista exibirá todas as funções existentes. Se você não declarou nenhum, o listview estará vazio.

Clique com o botão direito em um espaço vazio na lista - observe que os dois primeiros itens do menu são Nova Função e Novo Procedimento. A seleção de qualquer um desses itens criará uma declaração e um esboço de implementação.

Se você selecionar Nova Função, uma nova função chamada `GlobalFunction1` será adicionada à visualização da lista e a seguinte implementação será adicionada ao Editor de Código:

```
função GlobalFunction1: Variant;  
começar
```

```
Resultado: =
```

```
fim;
```

Da mesma forma, se você selecionar Novo Procedimento, um novo procedimento denominado GlobalProcedure1 será adicionado à visualização da lista e a seguinte implementação será adicionada ao Editor de Código:

```
procedimento GlobalProcedure1;  
  começar  
  
  fim;
```

Parâmetros de procedimento e função

As listas de parâmetros de função são as mesmas no RAP e no Object Pascal.

Como os manipuladores de eventos em RAP não podem se aplicar a mais de um evento, os manipuladores de eventos em RAP não têm um parâmetro Sender em sua assinatura.

Procedimentos e funções de chamada

Chamar uma função em RAP não é diferente de Object Pascal. Contanto que a função esteja no escopo, você pode chamá-la.

Ao chamar a função global de um relatório pai de dentro de um sub-relatório, você não precisa qualificar o identificador com o nome do relatório pai.

Em outras palavras, se seu relatório principal tem um procedimento global, GlobalDoSomething, e você deseja chamá-lo de dentro do evento OnCreate de SubReportA, não é necessário dizer Main.GlobalDoSomething. Basta chamar GlobalDoSomething.

Configurando RAP

Configurações de usuário final

Ao construir uma solução para o usuário final, você deve decidir quanto da funcionalidade do RAP expor aos usuários. O RAP permite um continuum de um diálogo de cálculos a um ambiente de desenvolvimento completo. Da mesma forma, você pode fornecer a seus usuários apenas os recursos de idioma básicos incluídos no RAP ou pode expor novas funções de passagem personalizadas, tornar o RAP ciente de suas próprias classes e até mesmo permitir que os usuários exibam os formulários personalizados que você criou.

As duas propriedades principais usadas para dimensionar o ambiente RAP são TppDesigner.RAPOptions e TppDesigner.RAPInterface.

A propriedade RAPInterface controla como o espaço de trabalho do Calc será apresentado ao usuário - como um diálogo, uma guia ou ambos.

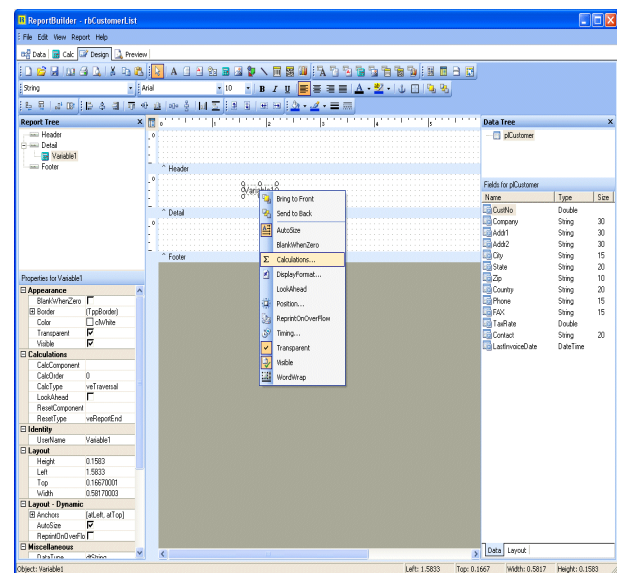
Quando RAP é apresentado como uma guia Calc no Report Designer, RAPOptions permite que você especifique o quanto o usuário pode visualizar e editar.

A interface de usuário RAP pode ser configurada para atender às necessidades de seus usuários finais. Existem duas configurações básicas disponíveis:

1 Diálogo de Cálculos

Para usuários finais casuais, uma caixa de diálogo Cálculos simples pode ser acessada no menu de velocidade do componente TppVariable. Esta caixa de diálogo fornece um local para os cálculos serem inseridos e retornados por meio de um único parâmetro Result. O resultado do cálculo é então exibido na variável quando o relatório é impresso.

Para configurar a interface do usuário para fornecer este nível de funcionalidade, defina a propriedade RAPInterface riNotebookTab como False e a propriedade riDialog como True. A figura abaixo ilustra a opção Cálculos ... disponível quando o usuário final acessa o menu rápido de uma variável.

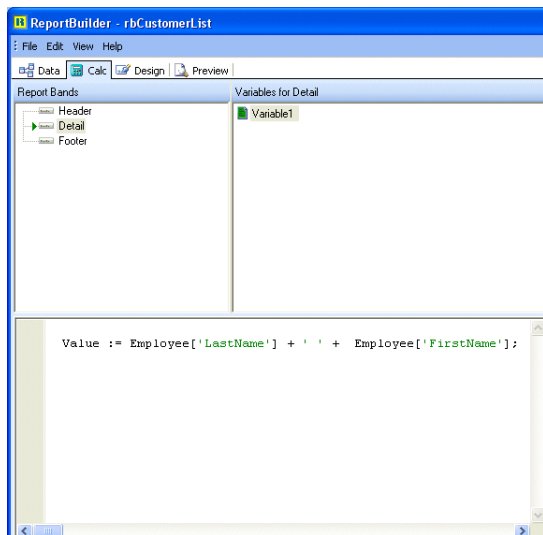


2 Calc Tab

Nesta configuração, seus usuários têm a mesma funcionalidade do Diálogo do Calc, mas está disponível na guia Calc. Na caixa de diálogo Calc, o Code Explorer não é mostrado, pois, nessa caixa de diálogo, lidamos com apenas uma variável de cada vez. No entanto, na guia Calc, o explorador é mostrado. Para configurar a interface do usuário para fornecer este nível de funcionalidade, defina a propriedade `RAPInterface riNotebookTab` como `True` e defina todas as `RAPOptions` como `True`. Quando o Report Designer é exibido, a guia Calc estará visível. Se você clicar com o botão direito na visualização em árvore à esquerda, um menu rápido que mostra as diferentes visualizações de cálculos disponíveis será exibido:

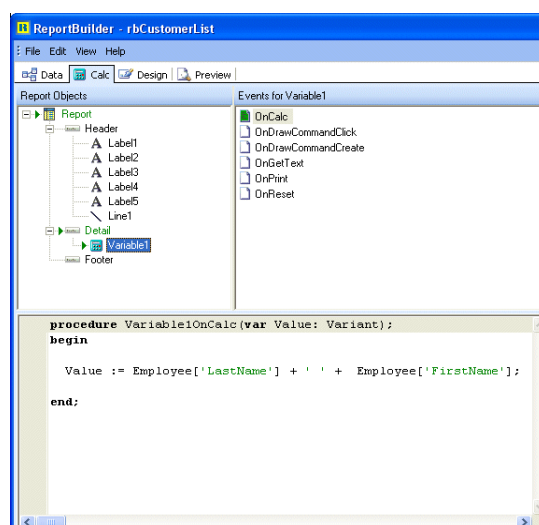
UMA Variáveis

Esta visualização mostra as variáveis em cada banda do relatório e quaisquer variáveis contidas na banda atualmente selecionada. A guia Calc torna mais fácil ver o código associado a quaisquer variáveis no relatório, mas ainda mantém o conceito de eventos oculto do usuário. Selecionar uma variável exibe o manipulador de eventos `OnCalc`, se houver.



B Eventos

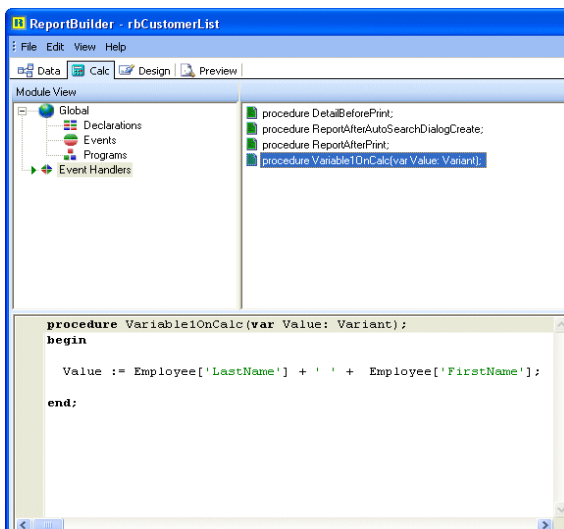
Esta visualização mostra o relatório e todos os objetos dentro dele. O usuário tem acesso a todos os eventos de objetos de relatório e pode codificar manipuladores de eventos para eles. O painel direito exibe todos os eventos associados ao componente atualmente selecionado. Selecionar um evento exibirá o manipulador de eventos, se houver um. Um manipulador de eventos pode ser codificado para qualquer objeto. Esta visão é boa para ver todos os objetos de relatório e seus eventos.



C Módulo

Finalmente, para expor todo o poder do espaço de trabalho do Calc, você pode adicionar as configurações globais a `RAPOptions`. Isso permite que o usuário veja e edite tudo o que está disponível no RAP. Esta visualização exibe itens que são visíveis a todos os manipuladores de eventos do relatório: declarações globais, funções, programas e manipuladores de eventos.

- **Declarações** - As variáveis e constantes que são globalmente visíveis em todo o relatório.
- **Eventos** - Esses são, em sequência, os eventos do relatório. No caso em que a janela de visualização é exibida, OnCreate e OnDestroy disparam quando a janela é aberta e fechada, respectivamente. Isso é diferente de Report.BeforePrint e AfterPrint porque esses métodos serão acionados sempre que Report.Print for chamado. OnCreate e OnDestroy são bons lugares para código de inicialização e finalização, como criar e liberar objetos e inicializar variáveis.
- **Programas** - Os procedimentos e funções que são globalmente visíveis em todo o relatório e, portanto, podem ser chamados de qualquer manipulador de eventos.
- **Manipuladores de eventos** - Todos os manipuladores de eventos que foram implementados no relatório.



Estendendo RAP

Funções de passagem RAP

No coração do RAP está um compilador de tempo de execução que executa o código dos usuários independentemente do compilador do Delphi. Portanto, o compilador do RAP tem seu próprio conjunto independente de tokens reconhecíveis. Em outras palavras, RAP é um subconjunto do Object Pascal.

O método para tornar o compilador RAP ciente das funções é declarar descendentes de `TraSystemFunction` - um para cada função - e registrá-los com o compilador usando `raRegisterFunction`. Cada classe descendente substitui `ExecuteFunction` - este método diz ao compilador RAP o que fazer quando um nome de função é encontrado no código do usuário, passando assim a verdadeira funcionalidade Delphi para o usuário final.

Por exemplo, quando o compilador RAP encontra a função `Copy`, ele executa o método `ExecuteFunction` que, em essência, diz ao RAP como atribuir a lista de parâmetros da função, então chama a função `Copy` do Delphi.

Para exemplos de registro de novas funções de passagem, consulte a demonstração RAP principal: `RBuilder \ Demos \ 0. RAP \ 1. Main \ RAP.dpr` - relatórios nº 31-33. Veja também o Tutorial de Adição de Funções.

Adicionando funções à caixa de ferramentas de código

Embora a guia Idioma da Caixa de ferramentas de código contenha muitas funções úteis para você e seu usuário, às vezes é necessário adicionar funcionalidade. Você pode registrar uma nova função de passagem com RAP com bastante facilidade. As etapas são:

- 1** Identifique uma função Delphi a ser chamada. Pode ser uma função que você escreveu, uma chamada DLL, uma chamada Win API ou uma função VCL padrão.
- 2** Crie um descendente `TraSystemFunction`.
- 3** Retorne a assinatura do seu procedimento no método `GetSignature`.
- 4** Indique se a chamada `HasParams` e `IsFunction` por meio desses dois métodos booleanos.
- 5** Faça a chamada real para a função ou procedimento Delphi no método `ExecuteFunction`.
- 6** Registre sua nova função com RAP por meio de uma chamada para `raRegisterFunction`. Coloque esta chamada na seção de inicialização da unidade que contém a classe descendente.
- 7** Adicione esta unidade ao seu projeto.

Ao executar o aplicativo, você verá sua nova função na guia de idioma da Caixa de ferramentas de código. Além disso, você poderá chamar a função a partir de qualquer programa ou manipulador de eventos RAP que você criar. Para um tutorial detalhado, consulte Adicionando novas funções ao RAP.

Adicionando classes e propriedades via RTTI

Os componentes do relatório no RCL são todos apresentados no RAP. Além disso, você notará que o compilador conhece TStrings, TStringList e TList. É possível tornar o RAP ciente de outros objetos que você pode desejar expor no espaço de trabalho do Calc. Por exemplo, você pode querer que seus usuários possam consultar o TmyUniversalFinancialComponent.

Para fazer isso, consulte Estendendo o RAP RTTI. Este tutorial mostrará não apenas como expor sua nova classe, mas também como disponibilizar as propriedades e métodos públicos dessa classe.

Depuração de código RAP

Suporte CodeSite

Não há depuração integrada para RAP - isso está um pouco além do escopo do produto. No entanto, há uma opção para fornecer suporte para a excelente ferramenta de depuração CodeSite - Raize Software Solutions.

Para que você, como desenvolvedor, use esta opção, você deve ter uma cópia licenciada do CodeSite instalada em seu computador. Se você deseja fornecer este suporte para seus usuários, eles devem ter cópias licenciadas do CodeSite instaladas.

O suporte da RAP à CodeSite não altera de forma alguma o acordo de licenciamento da Raize Software Solutions; qualquer pessoa que utilize o suporte do CodeSite da RAP deve ter uma versão devidamente registrada do CodeSite.

Agora, dito isso, a fim de habilitar o suporte a CodeSite em tempo de execução, você deve adicionar a unidade raCSFuncs à sua cláusula de usos. Isso registrará as funções de passagem do CodeSite com RAP, disponibilizando-as assim na caixa de ferramentas de código.

Para habilitar o suporte a CodeSite em tempo de design, você precisará compilar e instalar um pacote no Delphi. Veja o arquivo ReadMe.doc em seu RBuilder \ Demos \ 0. RAP \ 2. Diretório CodeSite para obter instruções. A maioria das chamadas CodeSite são disponibilizadas, algumas com algumas modificações devido à arquitetura do RAP.

Alguns itens notáveis:

- A propriedade Enabled do CodeSite está disponível como o procedimento csEnable no RAP. Passe um valor booleano para habilitar ou desabilitar CodeSite.
- Como o RAP ainda não suporta os tipos de registro, as funções SendPoint e SendPointEx foram alteradas para aceitar inteiros, X e Y em vez de um TPoint e as funções SendRect e SendRectEx foram alteradas para aceitar inteiros, Left, Top, Right e Bottom em vez de um TRect. As funções de passagem irão, por sua vez, mapear esses inteiros nos valores apropriados ao chamar os métodos CodeSite correspondentes.
- Há um projeto de demonstração que mostra o uso das funções de passagem do CodeSite em RBuilder \ Demos \ 0. RAP \ 1. CodeSite.

Usando as funções CodeSite

Para aqueles que não estão familiarizados com CodeSite, aqui estão algumas idéias de como usá-lo com RAP. Observe que este não é um manual sobre como usar o CodeSite; consulte a documentação do CodeSite para obter essas informações.

Digamos, por exemplo, que uma TppVariable não esteja exibindo um valor que você acha que deveria e você deseja ter certeza de que uma seção do código está realmente sendo executada. Você pode adicionar o seguinte código ao evento OnCalc da variável:

Este é um exemplo bastante simples, mas você pode ver que o uso liberal de chamadas CodeSite pode ser essencialmente como percorrer o código, ver os valores.

Há um projeto de demonstração que mostra o uso das funções de passagem do CodeSite em RBuilder \ Demos \ 0. RAP \ 2. CodeSite.

Código	Evento Variable1OnCalc
	<pre>procedimento Variable1OnCalc (var Valor: Variante); var IsLabelText: String; começar csEnterMethod ('Variable1OnCalc'); csSendString ('TaxRate', plCustomer ['TaxRate']); se plCustomer ['TaxRate']> 8, então começar csSendBoolean ('Detail.Overflow', Detail.Overflow); se Detail.Overflow então começar IsLabelText: = 'Continuação ...'; Valor: = ''; fim senão começar IsLabelText: = plCustomer ['Company']; Valor: = GetMyValue; fim; csSendString ('Valor', Valor); fim senão começar csSendWarning ('Invalid TaxRate'); Valor: = 'Inválido'; fim; csExitMethod ('Variable1OnCalc'); fim;</pre>

Compilando condicionalmente o suporte CodeSite

Adicionar e remover repetidamente chamadas CodeSite pode ser problemático, especialmente se você tiver muitas delas enigmáticas em seu código. Por esse motivo, as chamadas CodeSite podem ser compiladas condicionalmente. Todas as funções CodeSite na unidade raCSFuncs têm instruções condicionais em torno de suas implementações ExecuteFunction e no início da unidade está a linha:

```
{x $ DEFINE CODESITE}
```

Para habilitar as implementações da função, remova o "x" do início da linha. Para desativá-los, adicione o "x". Isso permitirá que você deixe todas as chamadas CodeSite no código sem ativar o objeto CodeSite na máquina do usuário.

Em outras palavras, para habilitar o suporte CodeSite para o seu relatório, faça o seguinte:

1 Abra a unidade raCSFuncs.

2 Role para o topo da unidade, logo abaixo da cláusula de interface.

3 Remova o 'x' da linha:

```
{x $ DEFINE CODESITE}
```

Isso habilitará o suporte a CodeSite.

4 Salve a unidade.

5 Reconstrua seu projeto selecionando Build All no Delphi.

Para desativar o suporte do CodeSite para o seu relatório sem remover as próprias chamadas do CodeSite, faça o seguinte:

1 Abra a unidade raCSFuncs.

2 Role para o topo da unidade, logo abaixo da cláusula de interface.

3 Para habilitar o suporte a CodeSite, adicione um 'x' no início da linha entre '{' e '\$'.

```
{ $ DEFINE CODESITE }
```

4 Adicione um 'x'

5 Salve a unidade.

6 Reconstrua seu projeto selecionando Build All no Delphi.

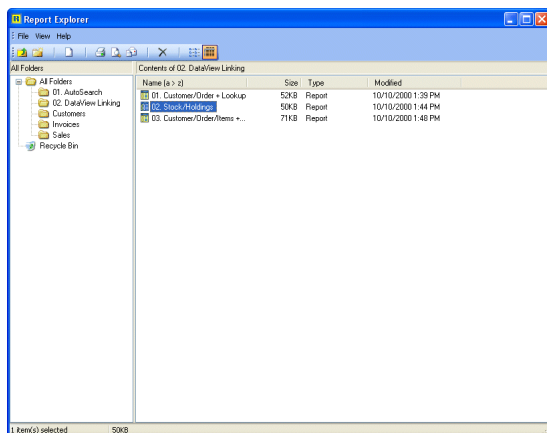
DESIGN

O Report Explorer 221

DESIGN

O Report Explorer

O componente Report Explorer permite que você implante uma interface do Windows Explorer que seus usuários finais podem usar para organizar seus relatórios. O Report Explorer lida com todas as operações por meio de pipelines de dados; portanto, a estrutura da pasta e todos os itens dentro dela podem ser salvos nas tabelas do banco de dados. Esta interface apresenta uma curva de aprendizado mínima, pois a maioria dos usuários está familiarizada com o funcionamento do Explorer. A interface do usuário do Report Explorer é ilustrada abaixo:



Barra de ferramentas do Report Explorer

A ação que cada botão da barra de ferramentas do Report Explorer executa é descrita a seguir.



Subir um nível

Quando uma pasta é selecionada, use este botão para selecionar a pasta pai.

Nova pasta

Cria uma nova pasta dentro da pasta atualmente selecionada.

Novo Relatório

Cria um novo relatório e exibe o Report Designer.

Relatório Aberto

Abre o relatório atualmente selecionado e o exibe no designer.

Impressão

Imprime o relatório atualmente selecionado.

Antevisão da Impressão

Abre o relatório atualmente selecionado e o exibe na janela de visualização de impressão.

E-mail

Gera o relatório selecionado em PDF e, em seguida, inicia o cliente de e-mail padrão do usuário final com o relatório PDF anexado.

PRINT

Opções do usuário final 225

PRINT

Opções do usuário final

Visão geral

Você pode controlar se os usuários finais podem arquivar ou imprimir em qualquer um dos formatos de arquivo suportados usando as seguintes propriedades do componente Relatório:

- AllowPrintToArchive
- AllowPrintToFile

Definir AllowPrintToArchive como True permitirá que os usuários finais arquivem relatórios. Em seguida, você precisaria fornecer uma interface de usuário para visualizar e imprimir os arquivos arquivados usando o componente ArchiveReader. Para obter um exemplo de como usar o ArchiveReader, consulte o exemplo 151 (`..\RBuilder\Demos\Reports\dm0151.pas`).

Definir AllowPrintToFile como True permitirá que os usuários finais imprimam em qualquer um dos arquivos disponíveis

formatos. Você pode controlar quais formatos de arquivo estão disponíveis para usuários finais por meio dos procedimentos de registro de classe de dispositivo. Por exemplo, o código a seguir removeria ReportTextFileDevice da lista de formatos de arquivo disponíveis:

```
usa  
ppDevice, ppFilDev;  
ppUnRegisterDevice (TppReportTextFileDevice);
```

Aplicativos de usuário final

Para definir as propriedades AllowPrintToArchive e AllowPrintToFile dentro do contexto de um aplicativo de usuário final, os eventos OnNew e OnLoad do objeto Report.Template devem ser usados para definir as propriedades padrão do relatório. Por exemplo, você adicionaria o código abaixo ao formulário principal para criar o aplicativo do usuário final.

Código	Codificar a forma principal de um aplicativo do usuário final
	<pre>procedimento TmyEndUserSolution.FormCreate (Sender: TObject); começar {atribuir manipulador de eventos para eventos de modelo} ppReport1.Template.OnNew: = ReportTemplateEvent; ppReport1.Template.OnLoadEnd: = ReportTemplateEvent; {remover o dispositivo de arquivo de texto de relatório} ppUnRegisterDevice (TppReportTextFileDevice); fim; procedimento TmyEndUserSolution.ReportTemplateEvent (Sender: TObject); começar {definir propriedades de relatório padrão} ppReport1.AllowPrintToArchive: = True; ppReport1.AllowPrintToFile: = True; fim;</pre>

DEPLOY

Resumo 229

DEPLOY

Resumo

Visão geral

Existem duas questões totalmente diferentes, mas vitais, envolvidas na decisão de como você implantará seu aplicativo:

1 Como implantar o ReportBuilder como parte de seu aplicativo

2 Como implantar os relatórios criados com ReportBuilder como parte de seu aplicativo

Implementação de aplicativos

Ao criar o exe do seu aplicativo, você pode optar por compilar tudo em um executável ou pode escolher distribuir os pacotes fornecidos com Delphi, ReportBuilder ou qualquer outro componente que esteja utilizando em seu aplicativo como arquivos separados. A última opção é conhecida como 'distribuição com pacotes' (pacotes aqui se referem a uma DLL compatível com Delphi) e resulta em um arquivo executável menor e mais eficiente.

Essas duas abordagens para implantar seu aplicativo são discutidas posteriormente nesta seção.

Implementação de relatórios

A segunda parte da implantação envolve determinar como implantar os relatórios criados com ReportBuilder como parte de seu aplicativo. ReportBuilder fornece as opções de implantação mais flexíveis de qualquer produto de relatório no mercado. As estratégias de implantação mais comumente usadas são as seguintes:

1 Soluções de relatórios do usuário final

A solução de relatórios do usuário final usa o Report Explorer para gerenciar relatórios armazenados em um banco de dados.

Existem duas opções de implantação para distribuir as tabelas de banco de dados do usuário final.

UMA Banco de dados local do usuário final

Nesse cenário, cada usuário final tem uma cópia local das tabelas do usuário final que são usadas como um banco de dados pessoal para armazenar relatórios.

B Banco de dados de usuário final compartilhado

Nesse cenário, há uma única cópia das tabelas do usuário final que são usadas como um banco de dados comum para armazenar todos os relatórios.

2 Soluções de relatórios padrão

Uma solução de relatório padrão refere-se àquela em que o desenvolvedor cria relatórios que podem ser executados pelo usuário final.

UMA Compile relatórios no executável.

Permita que o componente de relatório resida em um formulário invisível junto com os objetos de acesso a dados necessários. Na hora de imprimir o relatório, instancie o formulário e chame o método Print. Você pode variar ligeiramente essa arquitetura usando um módulo de dados como um contêiner para os objetos de acesso a dados. Este método requer que o formulário contenha o componente ppReport.

B Compile relatórios em um pacote.

Semelhante à opção A, mas em vez de compilar os formulários que contêm os relatórios no executável, os formulários de relatório são compilados em um pacote. A vantagem dessa abordagem é que ela otimiza o tamanho do executável e carrega dinamicamente o pacote que contém os relatórios conforme necessário.

C Distribuir arquivos de modelo de relatório.

Coloque todos os componentes de acesso a dados para os relatórios em um módulo de dados. Coloque um único componente de relatório que usa o módulo de dados no formulário. Crie todos os layouts de relatório por meio deste componente de relatório, salvando cada um em um arquivo de layout de relatório separado. Na hora de imprimir o relatório, carregue o relatório do arquivo e chame o método Print.

D Distribua um banco de dados de relatórios.

Use todas as mesmas etapas da opção C, mas salve as definições de relatório em um campo BLOB do banco de dados em vez de em arquivos separados. Com essa abordagem, você pode salvar todas as suas definições de relatório em uma tabela de banco de dados.

Uma implementação completa da Opção A acima é fornecida no tutorial 'Construindo um Aplicativo de Relatório' no Guia do Desenvolvedor. A opção A tem a vantagem de exigir o mínimo de código. A principal vantagem das opções C e D é que você pode alterar layouts de relatório e reimplantá-los para seus usuários finais com a regeneração de seu executável. Como a Opção A compila os layouts de relatório no executável, ela requer uma recompilação sempre que um relatório deve ser alterado. Em termos de tempo de carregamento do layout do relatório ou velocidade de impressão, todas as opções são essencialmente iguais.

REPORT TUTORIAIS

Criando um relatório por meio da árvore de dados	233
Criando um relatório por meio do assistente de relatório	241
Um relatório simples da maneira mais difícil	251
Grupos, cálculos e a faixa de resumo	259
Usando regiões para agrupar logicamente componentes dinâmicos	275
Emulação de formulários com um relatório	285
detalhado de imagem WMF	293
Relatório de Detalhe Mestre	307
Visualização interativa com sub-relatórios de detalhamento	319
Relatórios de enganchamento junto com sub-relatórios de estilo de seção	323
Usando colunas para criar etiquetas de endereçamento e impressão em um arquivo de texto	333
Imprimindo de um arquivo de texto	341
Usando o JITPipeline para imprimir de um StringGrid	345
Usando o componente Rich Text para correspondência / mesclagem	349
Criando uma tabela cruzada	353

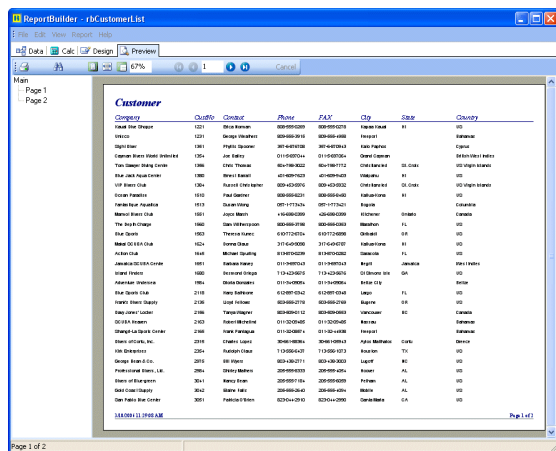
REPORT TUTORIAIS

Criação de um relatório por meio da árvore de dados

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Use a árvore de dados
- Trabalhar com o Report Designer
- Crie e configure os componentes de relatório mais comuns



Crie uma pasta de tutorial

1 Acesse o Windows Explorer.

2 Crie uma pasta na raiz do seu disco rígido.

3 Nomeie a pasta My RB Tutoriais.

Nota: Ao instalar novas versões, o programa de instalação do RB excluirá o diretório de instalação do RB existente. Por este motivo, sugerimos criar o diretório de tutoriais na raiz ou em algum lugar fora do diretório da RB.

Criar uma nova aplicação

1 Execute Delphi.

2 Selecione Arquivo | Novo aplicativo do menu

Delphi. Isso criará um novo projeto e um formulário em branco.

3 Defina o nome do formulário como 'frmViaDataTree'.

4 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbViaDT no diretório My RB Tutoriais.

5 Selecione Exibir | Gerente de projeto no menu principal.

6 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

7 Salve o projeto com o nome `rbDTProj` no diretório `My RB Tutoriais`.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie uma tabela, uma fonte de dados e um componente `DataPipeline`

1 Selecione a guia `BDE` da paleta de componentes Delphi.

2 Adicione um componente `Tabela` ao formulário.

3 Configure o componente `Tabela`:

Nome do banco de dados	<code>DBDEMOS</code>
Nome	<code>tblCustomer</code>
Nome da tabela	<code>customer.db</code>

4 Selecione a guia `Data Access` da paleta Delphi Component.

5 Adicione um componente `DataSource` ao formulário.

6 Configure o componente `DataSource`:

<code>DataSet</code>	<code>tblCustomer</code>
Nome	<code>dsCustomer</code>

7 Selecione a aba `RBuilder` do Delphi compo-paleta atual.

8 Adicionar um componente `DBPipeline`  para o formulário.

9 Configure o componente `DBPipeline`:

Fonte de dados	<code>dsCustomer</code>
Nome	<code>plCustomer</code>

Crie um relatório e conecte-o aos dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente `Relatório`:

`DataPipeline plCustomer`
Nome `rbCustomerList`

Chame o Report Designer e defina a orientação do papel

1 Clique duas vezes no componente `Relatório` para exibir o `Designer de Relatórios`.

2 Dimensione e mova a janela `Report Designer` para que o `Object Inspector` fique visível.

3 Selecione o arquivo | Opção `Configuração de página` no menu principal do `Report Designer`.

4 Clique na guia `Tamanho do papel` e altere a orientação do papel de `Retrato` para `Paisagem` e clique no botão `OK`.

Defina a altura da banda do cabeçalho para 1 polegada

1 Coloque o mouse sobre a divisória cinza intitulada 'Cabeçalho'. O cursor deve mudar para uma seta de dois lados. Mantenha pressionado o botão esquerdo do mouse, arraste para baixo uma curta distância e solte o botão do mouse. Verifique a régua vertical à esquerda e arraste para baixo até ver a marca de 2 polegadas.

2 Arraste o divisor de banda para cima até que a guia superior indique que você atingiu a marca de 1 polegada.

Nota: Conforme você arrasta o divisor de banda para baixo para aumentar a altura, a altura da régua aumenta com ele. Conforme você arrasta o divisor de banda para cima, uma guia é exibida na régua para indicar a nova altura da banda.

3 Clique com o botão direito sobre o espaço em branco da banda do cabeçalho e acesse a caixa de diálogo Posição. O valor da altura deve ser 1.

Crie etiquetas na faixa de cabeçalho

1 Clique no ícone do componente Label  no Paleta do componente Report Designer.

2 Clique no lado esquerdo da faixa do cabeçalho. Um rótulo será criado na banda de cabeçalho. Ela se tornará a seleção atual no Inspetor de objetos.

3 Localize a caixa Editar no canto superior esquerdo do Report Designer. Selecione o texto dentro dele e digite Lista de clientes.

4 Localize os controles de fonte na barra de formatação.

Observação: os controles de fonte aparecem à direita da caixa de edição e fornecem as seguintes funções:

- Lista suspensa de nomes de fontes
- Lista suspensa Tamanho da fonte
- Negrito
- itálico
- Estilo de fonte sublinhado
- Alinhamento de Texto
- Cor da fonte
- Cor de destaque

5 Defina a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito itálico

6 Posicione o rótulo no canto superior esquerdo da faixa de cabeçalho.

7 Coloque outro rótulo próximo ao centro do cabeçalho banda.

8 Configure o rótulo:

Rubrica	Aventuras marinhas e Sunken Treasure Co.
Nome da fonte	Times New Roman
Tamanho da fonte	16
Estilo de fonte	Negrito itálico
Alinhamento de Texto	Centrado

9 Defina a cor da fonte para marrom clicando no lado direito do ícone de cor da fonte  para exibir o Paleta de cores suspensa e, em seguida, clicando no quadrado marrom.

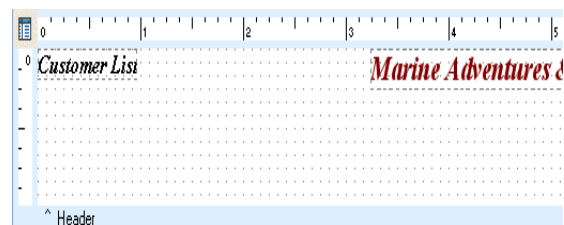
Nota: Quando você seleciona uma cor por meio dos ícones Fonte ou Cor de destaque, a cor selecionada é exibida abaixo do ícone. Você pode clicar no ícone diretamente (sem exibir novamente a paleta de cores suspensa) e o componente ou componentes selecionados atualmente serão definidos para essa cor.

10 Posicione o rótulo Marine Adventures na parte superior da área de trabalho da banda do cabeçalho.

11 Selecione Exibir | Barras de ferramentas | Alinhar ou espaçar no menu principal do Report Designer. A barra de ferramentas deve aparecer como uma janela flutuante.

12 Clique e arraste a janela da barra de ferramentas para o lado esquerdo do Report Designer e posicione-a de forma que ela se encaixe à esquerda da régua vertical.

13 Centralize o rótulo clicando no ícone Centralizar horizontalmente  na barra de ferramentas Alinhar ou Espaço.



Use a árvore de dados para organizar as informações do cliente

1 Selecione Exibir | Barras de ferramentas | Árvore de dados.

Posicione a árvore de dados à esquerda do Report Designer.

2 Clique na guia Layout na parte inferior da Árvore de dados e defina as configurações de arrastar e soltar como segue:

Crio	Tudo
Estilo	Tabular
Grade de etiqueta	Verdadeiro
Nome da fonte do rótulo	Times New Roman
Estilo de fonte da etiqueta	Negrito
Tamanho da fonte da etiqueta	12 pontos
Cor da fonte do rótulo	Marrom
Grade de campo	Verdadeiro
Nome da fonte do campo	Times New Roman
Estilo da fonte do campo	Regular
Tamanho da fonte do campo	10 pontos
Cor da fonte do campo	Preto

Observação: neste tutorial, estamos usando o estilo Tabular para criar uma grade de rótulos e componentes de campo. Depois de concluir este tutorial, você deve tentar novamente usando o estilo Vertical, bem como várias outras combinações das configurações de layout disponíveis.

3 Clique na guia Dados da Árvore de dados e selecione o Customer DataPipeline na lista da árvore no topo. Os campos do cliente serão exibidos na lista na parte inferior.

4 Clique no campo CustNo e, em seguida, mantenha pressionada a tecla Ctrl e clique nos seguintes campos na ordem especificada:

Empresa
Contato
Telefone
FAX

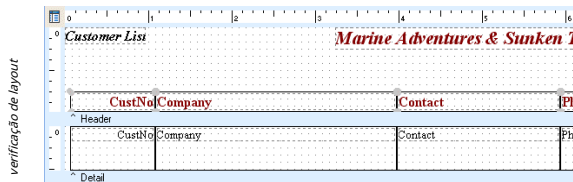
Nota: Este método de seleção de campo é denominado método Ctrl-clique.

5 Arraste os campos selecionados para a área de trabalho, posicionando o mouse no canto inferior esquerdo da faixa do cabeçalho. Solte o botão do mouse. Uma grade de rótulos será criada na parte inferior do componente da banda de cabeçalho e uma grade correspondente de componentes DBText será criada na banda de detalhes.

Observação: a árvore de dados possui um comportamento interno para a criação de rótulos e componentes com reconhecimento de dados. Se os componentes não couberem no espaço vertical da banda abaixo da etiqueta, eles serão empilhados no final da banda.

Nota: A largura atribuída a cada componente DBtext é calculada usando DisplayWidth dos componentes de campo do pipeline de dados. Ao usar o DBPipeline, os componentes do campo são criados automaticamente e usam as informações DisplayWidth dos campos no conjunto de dados ao qual o pipeline está conectado. Você pode substituir esse comportamento definindo a propriedade AutoCreateFields como False e usando o Editor de Campos do DataPipeline para definir o DisplayWidth padrão.

6 Feche a janela da árvore de dados.



Ajuste o Layout

1 Role para a direita até ver a borda direita da banda do cabeçalho.

2 Clique no espaço em branco da banda do cabeçalho para desmarcar os componentes.

3 Role para trás para a esquerda, selecione CustNo Label e, em seguida, mantenha pressionada a tecla Shift e clique no componente DBText correspondente diretamente abaixo dele. Este método de seleção de componentes é denominado método Shift-clique.

4 Clique no ícone Justificar à Esquerda.

Observação: o campo CustNo é numérico; portanto, o rótulo e o componente DBText foram justificados à direita quando criados pela Árvore de Dados.

5 Clique no espaço em branco da faixa de cabeçalho, logo acima do rótulo CustNo. Mantenha pressionado o botão do mouse e mova o cursor do mouse para baixo aproximadamente no meio do rótulo CustNo e, em seguida, mova o mouse para a direita - em todos os rótulos e formas na faixa de cabeçalho.

6 Solte o botão do mouse para selecionar os objetos.

Nota: Uma caixa pontilhada será desenhada na área de trabalho conforme você move o mouse. Ao soltar o botão do mouse, todos os objetos que cruzam os limites da caixa serão selecionados. Este método de seleção de componentes é denominado método de caixa delimitadora.

Nota: Se você não puder selecionar todos os objetos porque precisa rolar para a direita, selecione o máximo que puder e solte o botão do mouse. Agora vá para a direita. Mantenha pressionada a tecla Shift e crie outra caixa delimitadora pressionando o botão esquerdo do mouse e arrastando o mouse novamente. Ao soltar o botão do mouse, os objetos selecionados serão adicionados à lista. Se você selecionar novamente quaisquer objetos existentes, eles serão removidos da seleção.

7 Exiba a barra de ferramentas Desenhar.

8 Clique na seta do lado direito do Preenchimento

Ícone de cor na barra de ferramentas Desenhar para exibir o Paleta de cores suspensa e, em seguida, clique no quadrado colorido cinza claro. Isso preencherá os componentes da forma com cinza claro.

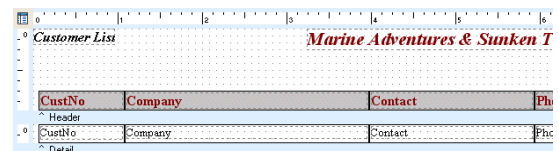
9 Pressione a tecla Ctrl e use a tecla de seta para baixo para mover a seleção para baixo até que esteja alinhada com a parte inferior da faixa de cabeçalho.

10 Coloque o mouse sobre a área cinza intitulada 'Detalhe' e clique com o botão direito do mouse. Selecione a opção de menu Posição Defina a altura para 0,2292.

Nota: As formas na banda de detalhe têm sua propriedade ParentHeight definida como True e, portanto, se ajustam automaticamente.

11 Selecione Arquivo | Salve no menu principal do Delphi.

12 Clique na guia Visualizar no Report Designer.



Adicionar um TimeStamp

1 Volte para a área de trabalho Design.

2 Clique no ícone do componente SystemVariable na  paleta de componentes do Report Designer.

3 Clique no lado esquerdo da faixa de rodapé. Um componente de variável do sistema será criado.

4 Posicione o componente no canto superior esquerdo da faixa de rodapé.

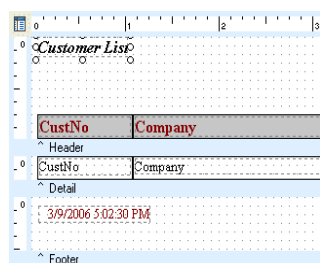
5 Localize a lista suspensa no canto superior esquerdo do Report Designer. Ele deve conter uma lista de tipos de variáveis para o componente.

6 Selecione o item PrintDateTime nesta lista. A data e hora atuais devem ser exibidas em o componente.

7 Defina a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	10
Cor da fonte	Preto
Estilo de fonte	Regular

Nota: O tipo de variável PrintDateTime faz com que SystemVariable exiba a data e a hora em que o relatório começou a ser impresso. A mesma data e hora são impressas em todas as páginas - é essencialmente um carimbo de hora que pode ser usado para rastrear o momento em que o relatório foi impresso.

**Adicionar números de página**

1 Role para a direita até que a borda da faixa de rodapé esteja visível.

2 Clique no ícone do componente SystemVariable na  paleta de componentes do Report Designer.

3 Clique no lado direito da faixa de rodapé. Um componente SystemVariable será criado.

4 Use a lista suspensa no lado esquerdo do Report Designer para selecionar o tipo de variável PageSetDesc. A página 1 de 1 deve ser exibida no componente.

5 Posicione o componente no canto superior direito da faixa de rodapé.

6 Justifique o componente à direita clicando no ícone justificar à direita na barra de ferramentas Formatar.

7 Role para a esquerda, mantenha pressionada a tecla Shift e selecione a variável de sistema PrintDateTime.

8 Clique no ícone Alinhar Superior  ou

Barra de ferramentas do espaço.

9 Clique no ícone Selecionar Relatório , que aparece no canto superior esquerdo do Report Designer, onde as régua horizontal e vertical se encontram. Isso selecionará o componente de relatório no Inspetor de objetos.

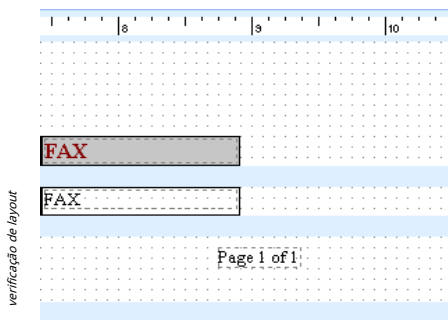
10 Localize a propriedade PassSetting do Relatório no Inspetor de objetos.

11 Certifique-se de que esteja definido como psTwoPass. Esta propriedade é definida automaticamente quando você seleciona um tipo SystemVariable que requer uma contagem de páginas.

Nota: Definir o PassSetting do relatório em psTwoPass fará com que o mecanismo de relatório gere todas as páginas automaticamente. Os relatórios de uma passagem geram apenas a primeira página e, em seguida, quaisquer páginas solicitadas depois disso. Isso não permite que o relatório calcule o número total de páginas. Relatórios de duas passagens são necessários sempre que você estiver usando o PageCount em uma variável do sistema.

12 Selecione Arquivo | Salve no menu principal do Delphi.

13 Clique na guia Visualizar no Report Designer.



Visualize o relatório em tempo de execução

1 Feche a janela Report Designer.

2 Selecione a guia Padrão da paleta de componentes Delphi.

3 Adicione um componente Botão ao formulário.

4 Configure o componente Botão:

Nome btnPreview

Antevisão da legenda

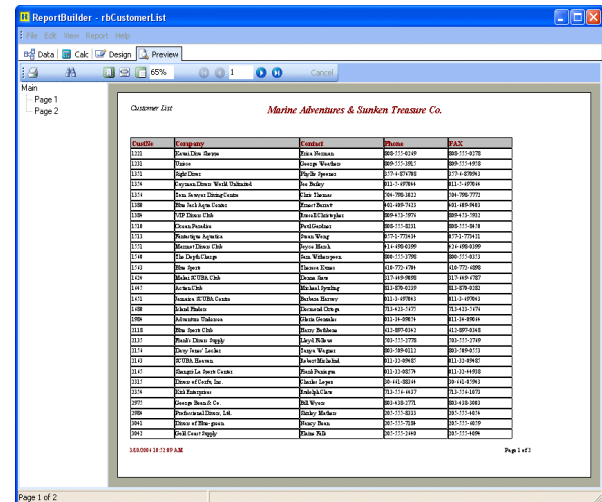
5 Adicione o seguinte código ao manipulador de eventos OnClick do botão:

```
rbCustomerList.Print;
```

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Execute o projeto.

8 Clique no botão Visualizar. A propriedade DeviceType do componente de relatório é padronizada como Tela; assim, o relatório exibe automaticamente o formulário Visualizar impressão quando o método de impressão é chamado. O formulário de visualização deve ser semelhante a este:



Criação de um relatório por meio do assistente de relatório

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configurar objetos de acesso a dados Delphi para uso por relatórios
- Use o Assistente de Relatório para criar layouts de relatório
- Salvar e carregar layouts de relatório para relatar
- arquivos de modelo (.rtm)
- Adicionar código para visualizar e imprimir o relatório em tempo de execução

Relatório Tabular

<i>plCustomer</i>							
Company	CustNo	Contact	Phone	FAX	City	State	Country
Action Club	1645	Michael Spurling	813-870-0238	813-870-0282	Sarasota	FL	US
Action Diver Supply	3158	Marianne Miles	22-44-500211	22-44-500596	St. Thomas		US Virgin Islands
Adventure Undersea	1984	Shonda Underseas	011-34-90054	011-34-90064	Belize City		Belize
American SCUBA Supply	3053	Lynn Cincosapi	213-654-0092	213-654-0095	Lundia	CA	US
Aquatic Drama	6312	Ollian Owen	613-442-7654	613-442-7678	Tampa	FL	US
Blue Glass Happiness	3984	Christine Taylor	213-555-1984	213-555-1995	Santa Monica	CA	US
Blue Jack Aqua Center	1380	Ernest Banath	401-600-7623	401-600-9403	Wapahus	RI	US
Blue Sports	1993	Theresa Klones	610-772-6764	610-772-6668	Oslo-Alt	OR	US
Blue Sports Club	2118	Katy Bathbone	613-887-0340	612-887-0348	Largo	FL	US
Catamenian Dive Club	3054	Nicole Dupont	213-223-0941	213-223-2324	Catalina Island	CA	US
Cayman Diver World Unlimited	1354	Joe Bailey	011-8-897044	011-8-897064	Grand Cayman		British West Indies
Central Underwater Supplies	5151	Maria Bruckhoff	27-11-4802488	27-11-4802058	Johannesburg		Republic So. Africa
Damp Jonaef Locker	2150	Tanya Wagner	803-609-0112	803-609-0553	Vancouver	BC	Canada
Diver's Oretto	3055	Peter Owen	213-432-0093	213-432-4821	Downey	CA	US
Divers of Blue-green	3041	Nancy Dean	205-555-7184	205-555-8050	Pellham	AL	US
Divers of Corfu, Inc.	2315	Charles Lopez	30-661-98364	30-661-05943	Ajaccio Matthaios	Corfu	Greece
Divers of Mexico	4212	Simone Green	915-482-2095	915-482-9842	Vigilias	FL	US
Divershotline	6432	Joe Hutter	878-808076	878-059345	Soua		Fiji
Fantastichic Aquatics	1813	Susan Wong	057-1-773424	057-1-773421	Bogota		Columbia
Fisherman's Eye	3151	Bethan Lewis	800-655-4860	800-655-4860	Grand Cayman		British West Indies
Frodo's Diver Supply	2105	Lloyd Fellows	603-555-2778	603-555-2709	Eugene	OR	US
George Dean & Co.	2876	Bill Wayne	803-438-2771	803-438-3533	Logan	NC	US
Guild Craft Supply	3042	Elaine Falls	205-555-2640	205-555-4504	Mobile	AL	US
Island Fishers	1890	Ernest Odaga	713-423-5078	713-423-5978	St. Vincent Mta	GA	US

Relatório Vertical

<i>plCustomer</i>							
Company	Action Club	Company	Aquatic Drama	Company	Action Diver Supply	Company	Blue Glass Happiness
CustNo	1645	CustNo	6312	CustNo	3158	CustNo	3984
Contact	Michael Spurling	Contact	Ollian Owen	Contact	Marianne Miles	Contact	Christine Taylor
Phone	813-870-0239	Phone	613-442-7654	Phone	22-44-500211	Phone	213-555-1984
FAX	813-870-0282	FAX	613-442-7678	FAX	22-44-500596	FAX	213-555-1995
City	Sarasota	City	Tampa	City	St. Thomas	City	Santa Monica
State	FL	State	FL	State		State	CA
Country	US	Country	US	Country	US Virgin Islands	Country	US
Company	Adventure Undersea	Company	Blue Jack Aqua Center	Company	Adventure Undersea	Company	Blue Jack Aqua Center
CustNo	1984	CustNo	1380	CustNo	1984	CustNo	1380

Criar uma nova aplicação

1 Selecione Arquivo | Novo aplicativo do menu

Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmViaReportWizard'.

3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbViaWiz no diretório My RB Tutoriais (estabelecido na página 181).

Nota: É importante nomear o formulário e salvar a unidade do formulário usando os nomes fornecidos acima porque o relatório será usado em tutoriais posteriores.

4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

5 Clique com o botão direito sobre o nome do projeto no

Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

6 Salve o projeto com o nome rbRWProj no diretório My RB Tutoriais.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie uma tabela, uma fonte de dados e um componente DataPipeline

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblCustomer
Nome da tabela	customer.db
IndexFieldName	Empresa

4 Selecione a guia Data Access da paleta Delphi Component.

5 Adicione um componente DataSource ao formulário.

6 Configure o componente DataSource:

DataSet tblCustomer
Nome dsCustomer

7 Selecione a aba RBuilder do Delphi compo-paleta atual.

8 Adicionar um componente DBPipeline  para o formulário.

9 Configure o componente DBPipeline:

Fonte de dados dsCustomer
Nome plCustomer

Crie um relatório e conecte-o aos dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

DataPipeline plCustomer
Nome rbCustomerList

Chame o Report Designer e acesse o Report Wizard

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Selecione Arquivo | Novo no menu do Report Designer.

3 Clique duas vezes no ícone do Assistente de relatório (ou clique no botão OK enquanto o ícone é selecionado).

Use o assistente de relatório para criar um relatório de estilo tabular

1 Clique na lista suspensa Nome do pipeline de dados e selecione a tabela Cliente.

2 Clique no campo Empresa na caixa de listagem Campos disponíveis e clique no botão de seta '>' para mover o campo para a caixa de listagem Campos selecionados.

Nota: Você também pode mover um campo para a caixa de listagem Selecionar campos clicando duas vezes nele.

3 Repita este procedimento para selecionar os seguintes campos:

CustNo
Contato
Telefone
FAX
Cidade
Estado
País

4 Clique no botão Avançar para acessar a próxima página do Assistente.

5 Esta página do Report Wizard é usada para definir grupos. Nenhum grupo é necessário para este exemplo; portanto, clique no botão Avançar novamente.

6 Clique no botão de opção Paisagem localizado na caixa de grupo Orientação.

7 Clique no botão Avançar.

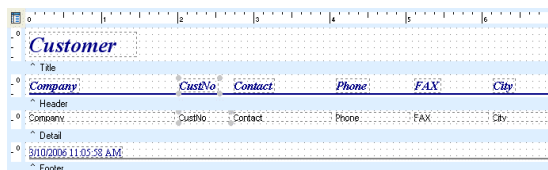
8 Clique em cada uma das opções de estilo na caixa de listagem para visualizar cada um dos estilos disponíveis.

9 Selecione o estilo Corporativo e clique no botão Avançar.

10 Clique no botão Concluir. O layout do relatório será gerado e a guia Visualização do Report Designer será selecionada para que você possa visualizar o relatório.

11 Clique na guia Design.

12 Selecione o rótulo CustNo e os componentes DBText e clique no ícone Justificar à esquerda.



Visualize o relatório

1 Clique na guia Visualizar.

2 Clique nos botões Largura da página e 100% para visualizar o relatório em uma escala maior.

3 Digite 80 na caixa de edição% e pressione Enter.

4 Digite 120 na caixa de edição e pressione Enter novamente. O relatório pode ser escalado para visualização em qualquer tamanho até 250%.

5 Clique no botão Largura da página para continuar visualizando o relatório.

6 Clique nos botões da página Próximo e Anterior para navegar pelas páginas do relatório.

7 Clique no botão Imprimir para acessar a caixa de diálogo de impressão. A partir daqui, você pode imprimir Tudo, Atual ou um intervalo específico de páginas. Clique no botão Cancelar para fechar a caixa de diálogo.

8 Clique na guia Design.

Defina o relatório para o modo de duas passagens

1 Selecione o componente Variável do sistema na extremidade direita da faixa de rodapé.

2 Use o controle da lista suspensa no canto esquerdo do Report Designer para alterar o tipo de variável de PageNoDesc para PageSetDesc.

3 Clique no ícone Selecionar Relatório, que aparece no canto superior esquerdo do Report Designer, onde as régua horizontal e vertical se encontram. Isso selecionará o objeto de relatório no Inspetor de objetos.

4 Localize a propriedade PassSetting do Relatório no Inspetor de objetos.

5 Certifique-se de que esteja definido como psTwoPass. Esta propriedade é definida automaticamente quando você seleciona um tipo SystemVariable que requer uma contagem de páginas.

Nota: Definir o PassSetting do relatório em psTwoPass fará com que o mecanismo de relatório gere todas as páginas automaticamente. Os relatórios de uma passagem geram apenas a primeira página e, em seguida, quaisquer páginas solicitadas depois disso. Isso não permite que o relatório calcule o número total de páginas. Relatórios de duas passagens são necessários sempre que você estiver usando o PageCount em uma variável do sistema.

Salvar o layout do relatório em um arquivo de modelo

1 Selecione Arquivo | Salvar como no menu do Report Designer.

2 Digite CustTab na caixa de edição Nome do arquivo e clique no botão Salvar. Certifique-se de salvar o modelo no mesmo diretório do seu projeto.

Nota: Por padrão, o layout do relatório será salvo com o formulário no qual o componente TppReport reside. Salvar o layout de um relatório em um arquivo de modelo é opcional. No entanto, para este exemplo, queremos criar dois layouts de relatório para os mesmos dados. O usuário pode escolher qual layout imprimir no tempo de execução. Portanto, vamos salvar os layouts de relatório para arquivos rtm e carregue o layout apropriado em tempo de execução.

Use o assistente de relatório para criar um relatório de estilo vertical

1 Selecione Arquivo | Novo no menu do Report Designer.

2 Clique duas vezes no ícone ReportWizard (ou clique no botão OK enquanto o ícone é selecionado).

3 Clique na caixa de combinação Nome do pipeline de dados e selecione a tabela Cliente.

4 Clique na linha Empresa da caixa de listagem Campos disponíveis e clique no botão de seta '>' para mover o campo para a caixa de listagem Campos selecionados.

5 Repita este procedimento para selecionar os seguintes campos:

CustNo

Contato

Telefone

FAX

Cidade

Estado

País

6 Clique no botão Avançar para acessar a próxima página do Assistente.

7 Esta página do Report Wizard é usada para definir grupos. Para este exemplo, nenhum grupo é necessário; portanto, clique no botão Avançar novamente.

8 Clique no botão de opção Vertical localizado na caixa de grupo Layout.

9 Clique no botão Avançar.

10 Selecione o estilo Soft Grey e clique no botão Next.

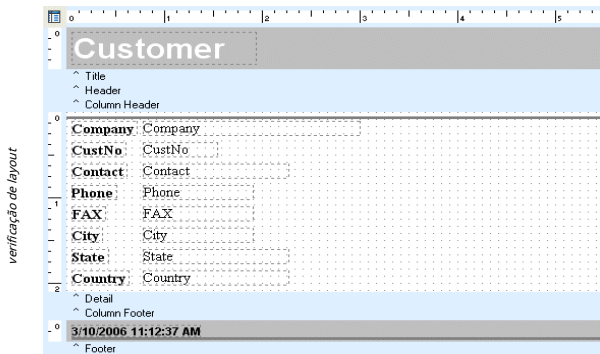
11 Clique no botão Concluir. O layout do relatório será gerado e a guia Visualização do Report Designer será selecionada para que você possa visualizar o layout do relatório.

12 Clique na guia Design.

13 Selecione o rótulo CustNo e os componentes DBText e clique no ícone Justificar à esquerda.

14 Selecione o componente Variável do sistema na extremidade direita da faixa de rodapé.

15 Use o controle da lista suspensa no canto esquerdo do Report Designer para alterar o tipo de variável de PageNoDesc para PageSetDesc.



Modifique o layout do relatório para conter colunas

1 Selecione Arquivo | Configuração da página no menu principal do Report Designer.

2 Selecione a guia Layout.

3 Clique no botão Seta para cima localizado à direita da caixa de edição Colunas. Isso deve modificar o número de colunas de 1 a 2.

4 Clique no botão OK.

5 Clique com o botão direito sobre o componente de linha na parte superior da banda de detalhes e defina a propriedade ParentWidth como True.

Observação: ao usar os menus pop-up do componente, uma marca de seleção ao lado de uma opção do menu indica que a propriedade está definida como True. Selecionar a opção de menu alterna a propriedade de Marcado para Desmarcado (ou seja, Verdadeiro para Falso) e vice-versa.

6 Visualize o relatório novamente. Você deve ver duas colunas.

Nota: Frequentemente, você desejará usar o Assistente de Relatório para fazer o layout de um relatório e, em seguida, personalizar o layout do relatório usando o Report Designer.

Salvar o layout do relatório em um arquivo de modelo

1 Volte para a área de trabalho de design.

2 Selecione Arquivo | Salvar como no menu principal do Report Designer.

3 Digite CustVert na caixa de edição Nome do arquivo e clique no botão Salvar. Certifique-se de salvar o template no diretório My RB Tutorials.

Nota: Você pode usar o arquivo | Abra o menu para acessar a caixa de diálogo de abertura de arquivo e carregar os arquivos CustTab.rtm e CustVert.rtm em tempo de design. Para salvar quaisquer alterações, use o arquivo | Salvar opção de menu.

Adicione os componentes de interface de tempo de execução

1 Feche o Report Designer selecionando Arquivo | Feche no menu principal do Report Designer.

Nota: Você também pode fechar o Report Designer clicando no botão X localizado no canto superior direito da barra de título da janela ou clicando duas vezes no ícone R localizado no canto superior esquerdo da barra de título da janela.

2 Selecione a guia Padrão da paleta de componentes Delphi.

3 Adicione um componente Group Box ao formulário.

4 Configure o Grupo B componente boi:

Nome	gbReportStyle
Rubrica	Estilo de Relatório
Deixou	15
Principal	11
Largura	131
Altura	82

5 Adicione dois componentes de botão de rádio ao grupo Caixa:

Nome	rbTabular
Rubrica	Tabular
Verificado	Verdadeiro
Deixou	18
Principal	23
Largura	65
Nome	rbVertical
Rubrica	Vertical
Deixou	18
Principal	50
Largura	65

6 Adicione dois componentes de botão ao formulário.

7 Configure os componentes do botão:

Nome	btnPreview
Rubrica	Antevisão
Deixou	163
Principal	28
Nome	btnPrint
Rubrica	Impressão
Deixou	163
Principal	56

Codifique os manipuladores de eventos

1 Na seção privada da unidade de formulário, adicione a seguinte declaração:

privado

FPathName: String;

2 Adicione um manipulador de eventos para o evento Form

OnCreate:

Código	Manipulador de eventos para o evento Form OnCreate
	<pre>procedimento TfrmViaReportWizard.FormCreate (Sender: TObject); começar {obtenha o nome do caminho para este aplicativo} FPathName := ExtractFilePath (ParamStr (0)); rbCustomerList.Template.FileName := {definir o nome do arquivo do modelo de relatório inicial} FPathName + 'CustTab.rtm'; fim;</pre>

3 Adicione manipuladores de eventos para cada evento OnClick do

botão:

Código	Manipulador de eventos para o evento OnClick do botão
	<pre>procedimento TfrmViaReportWizard.btnPreviewClick (Sender: TObject); começar {carregue o arquivo de modelo e imprima na tela} rbCustomerList.Template.LoadFromFile; rbCustomerList.DeviceType := dtScreen; rbCustomerList.Print; fim; procedimento TfrmViaReportWizard.btnPrintClick (Sender: TObject); começar {carregue o arquivo de modelo e imprima na impressora} rbCustomerList.Template.LoadFromFile; rbCustomerList.DeviceType := dtPrinter; rbCustomerList.Print; fim;</pre>

4 Adicione manipuladores de eventos para cada evento OnClick do botão de rádio, conforme mostrado abaixo.

5 Adicione ppTypes à cláusula uses da unidade do seu formulário.

Observação: a unidade ppTypes contém a declaração de todos os tipos enumerados e constantes usados no ReportBuilder. Nesse caso, é necessário oferecer suporte às referências a dtPrinter e dtScreen nos manipuladores de eventos de clique do botão anteriores.

Compilar e executar o aplicativo

1 Selecione o projeto | Compile | rbRWProj. Corrija quaisquer problemas de compilação.

2 Selecione Arquivo | Salve no menu principal do Delphi.

3 Feche o formulário e a unidade.

4 Execute o projeto.

5 Clique no botão Visualizar para visualizar o relatório de estilo tabular.

6 Clique no botão de opção Vertical.

7 Clique no botão Visualizar para ver o relatório de estilo vertical.

Nota: Você também pode enviar o relatório para a impressora clicando no botão Imprimir.

Código	Manipulador de eventos para botões de rádio 'OnClick'
<pre>procedimento TfrmViaReportWizard.rbTabularClick (Sender: TObject); começar {definir o nome do arquivo de modelo para o relatório tabular} rbCustomerList.Template.FileName:= FPathName + 'CustTab.rtm'; fim; procedimento TfrmViaReportWizard.rbVerticalClick (Sender: TObject); começar {definir o nome do arquivo de modelo para o relatório vertical} rbCustomerList.Template.FileName:= FPathName + 'CustVert.rtm'; fim;</pre>	

Relatório Tabular

Relatório Vertical

ReportBuilder - rhCustomerList

File Edit View Report Help

Data

Calc

Design

Preview

67%

Cancel

Man

Page 1

Page 2

Page 3

Page 4

Page 5

Page 6

Page 7

Customer

Company

Kamal Drive Shoppe

Country

1221

Contact

Erica Norman

Phone

808-555-0269

FAX

808-555-0278

City

Kono Kono

State

HI

Country

US

Company

Ukiko

Country

1211

Contact

Geonze Weathers

Phone

808-555-3915

FAX

808-555-4958

City

Pleasant

State

HI

Country

Bahamas

Company

Sight Drive

Country

1331

Contact

Phyllis Spooner

Phone

357-6-876700

FAX

357-6-870943

City

Kono Pehos

State

Cyprus

Company

Coman Drive World Unlimited

Country

1334

Contact

Jay Bailey

Phone

011-5-697044

FAX

011-5-697064

City

Orand Coman

State

British West Indies

Company

Tom Servey Drive Centre

Country

1336

Contact

Chris Thomas

Phone

304-798-3022

FAX

304-798-7772

City

Christianssted

State

St. Croix

Country

US Virgin Islands

Company

Elio Jack Aqua Center

Country

1380

Contact

Eliott Buerst

Phone

401-609-7623

FAX

401-609-9403

City

Wapahin

State

HI

Country

US

Company

VID Drive Club

Country

1384

Contact

Russell Christopher

Phone

809-453-5976

FAX

809-453-5952

City

Christianssted

State

St. Croix

Country

US Virgin Islands

Company

Ocean Paradise

Country

1370

Contact

Paul Oberst

Phone

808-555-8231

FAX

808-555-8450

City

Kaika Kona

State

HI

Country

US

2010/06/11 17:27 AM

Page 1 of 7

Um relatório simples da maneira mais difícil

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configurar objetos de acesso a dados Delphi para uso por relatórios
- Trabalhar com o Report Designer
- Crie e configure os componentes de relatório mais comuns

Customer List		Marine Adventures and Sunken Treasures			
Company	Contact	Phone	Address	City	State
Kauai Dive Shoppe	Erica Norman	808-555-0269	4-976 Sigulouf Hwy	Kapa Koni	HI
Unairo	George Weather	809-555-3915	PO Box Z-547	Freeport	
Sight Diver	Phyllis Spooner	357-6-876708	1 Neptune Lane	Kato Paglos	
Cayman Divers World Unlimited	Joe Bailey	011-5-697044	PO Box 541	Grand Cayman	
Tom Sawyer Diving Centre	Chris Thomas	304-798-3022	632-1 Third Frydenlej	Christianssted	St. Croix
Blue Jack Aqua Center	Ernest Burnett	401-609-7623	23-738 Piddington Lane	Waipahu	HI
VIP Divers Club	Russell Christopher	809-455-5976	32 Main St	Christianssted	St. Croix
Ocean Paradise	Paul Overizer	808-555-8251	PO Box 8745	Kailua-Kona	HI
Fantastique Aquatics	Susan Wong	0571-779424	232 999 #2A-77 A.A.	Bogota	
Manatee Divers Club	Joyce Mault	416-690-0399	872 Queens St	Kitchener	Ontario
The Depth Charge	Sam Wilkenson	800-555-5798	15240 Underwater Ferry	Mandarin	FL
Blue Sports	Thomas Kmet	610-772-6704	202 12th Ave. Box 746	Garibaldi	OR
Malibu SCUBA Club	Donna Slaus	317-640-9098	PO Box 8334	Kailua-Kona	HI
Artisan Club	Michael Spruiling	013-670-0239	PO Box 5451-F	Sarasota	FL
Jamaica SCUBA Centre	Babara Harvey	011-3-697043	PO Box 68	Negril	Jamaica
Island Paddlers	Demond Ortega	713-423-5675	6133 1/3 Shore Avenue	St Simons Isle	GA
Adventure Undersea	Olivia Cronshaw	011-34-0654	PO Box 744	Belize City	
Blue Sports Club	Harry Bathone	612-897-0342	63365 Mee Pene Street	Largo	FL
Funk's Divers Supply	Lloyd Feltrow	503-555-2778	1455 North 44th St	Engle	OR

Criar uma nova aplicação

1 Selecione Arquivo | Novo aplicativo do menu

Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmCustomerList'.

3 Selecione Arquivo | Salvar como no menu Delphi e

salvar o formulário com o nome rbCust no diretório My RB Tutorials (estabelecido na página 181).

Nota: É importante nomear o formulário e salvar a unidade do formulário usando os nomes fornecidos acima porque o relatório será usado em tutoriais posteriores.

4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

5 Clique com o botão direito sobre o nome do projeto no

Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

6 Salve o projeto com o nome rbCSPProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie uma tabela, uma fonte de dados e um componente DataPipeline

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblCustomer
Nome da tabela	customer.db

4 Adicione um componente DataSource ao formulário.

5 Configure o componente DataSource:

DataSet	tblCustomer
Nome	dsCustomer

6 Selecione a aba RBuilder do Delphi compo-paleta atual.

7 Adicionar um componente DBPipeline  para o formulário.

8 Configure o componente DBPipeline:

DataSource	dsCustomer
Nome	plCustomer

Crie um relatório e conecte-o ao Dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

DataPipeline	plCustomer
Nome	rbCustomerList

Chame o Report Designer e defina a orientação do papel

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

3 Selecione o arquivo | Opção Configuração de página no menu principal do Report Designer.

4 Clique na guia Tamanho do papel e altere a orientação do papel de Retrato para Paisagem e clique no botão OK.

Esquematize o rótulo do título da lista de clientes na faixa do cabeçalho

1 Clique com o botão direito sobre o espaço em branco da banda do cabeçalho e acesse a caixa de diálogo Posição

2 Defina a altura para 0,9167 e clique no botão OK. A banda do cabeçalho deve expandir em altura.

Nota: Você também pode alterar a altura da banda do cabeçalho arrastando o divisor que aparece abaixo do espaço em branco da banda do cabeçalho. Este método é rápido e fácil, mas não tão preciso.

3 Clique no ícone do componente Label  no Paleta do componente Report Designer.

4 Clique no lado esquerdo da faixa do cabeçalho. Um rótulo será criado na banda de cabeçalho. Ela se tornará a seleção atual no Inspetor de objetos.

5 Localize a caixa Editar no canto superior esquerdo do Report Designer.

6 Selecione o texto dentro dele e substitua-o por Lista de clientes.

7 Localize os controles de fonte na barra de formatação.

Observação: os controles de fonte aparecem à direita da caixa de edição e fornecem as seguintes funções:

- Lista suspensa de nomes de fontes
- Lista suspensa Tamanho da fonte
- Negrito
- itálico
- Estilo de fonte sublinhado
- Alinhamento de Texto
- Cor da fonte
- Cor de destaque

8 Defina a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito itálico

9 Localize a régua horizontal. Está acima da banda do cabeçalho.

10 Posicione o mouse sobre a régua e clique com o botão direito.

11 Selecione a opção Milímetros no menu de velocidade.

12 Clique com o botão direito sobre o rótulo e selecione a opção de menu Posição

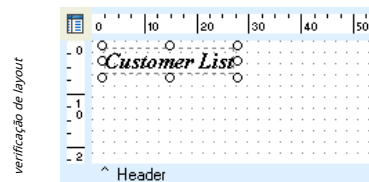
13 Insira os seguintes valores na caixa de diálogo Posição

registro:

Deixou	2.381
Principal	1.323
Largura	24.871
Altura	5.027

14 Clique no botão OK. O rótulo deve mover-se para o canto superior esquerdo da banda de cabeçalho.

Nota: Você também pode alterar a posição de um componente arrastando-o. A posição exata do componente será exibida na barra de status na parte inferior do Report Designer depois que ele for solto. Você pode usar qualquer unidade de medida (polegadas, pixels da tela, milímetros, etc.) de sua preferência.



Esquematize a etiqueta de título da empresa na faixa do cabeçalho

1 Coloque outro rótulo próximo ao centro do cabeçalho banda.

2 Configure o rótulo:

Rubrica	Aventuras marinhas e Sunken Treasures Co.
Nome da fonte	Times New Roman
Tamanho da fonte	16
Estilo de fonte	Negrito itálico

3 Clique no ícone suspenso da cor da fonte e  e selecione azul escuro.

4 Selecione Exibir | Barras de ferramentas do menu principal do Report Designer. Uma lista de nomes de barra de ferramentas deve ser exibida. Selecione a barra de ferramentas Alinhar ou Espaço.

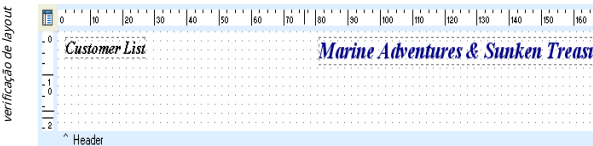
5 Clique no componente do rótulo que você acabou de criar na faixa de cabeçalho.

6 Clique no ícone Centralizar horizontalmente  no Barra de ferramentas Alinhar ou Espaço. Isso deve fazer com que o rótulo fique centralizado na banda do cabeçalho.

7 Clique no rótulo Lista de clientes.

8 Mantenha pressionada a tecla Shift e clique no relatório Rótulo do título.

9 Clique no ícone Alinhar Superior  do Alinhar ou Barra de ferramentas do espaço. Os rótulos do título agora devem estar alinhados.



Crie etiquetas para a faixa de cabeçalho

1 Coloque seis etiquetas na faixa do cabeçalho e posicione-as da esquerda para a direita. Defina as legendas do rótulo da seguinte maneira:

- Empresa
- Contato
- Telefone
- Endereço
- Cidade
- Estado

2 Clique no rótulo da empresa.

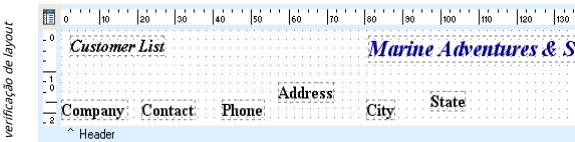
3 Mantenha pressionada a tecla Shift e clique no rótulo do contato.

4 Repita este procedimento até que todos os seis rótulos sejam selecionados.

5 Defina a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito
Cor da fonte	Preto

6 Deixe essas etiquetas em suas posições atuais; vamos alinhá-los mais tarde.



Complete o Layout da Faixa do Cabeçalho

1 Clique no ícone do componente Linha na  no paleta do componente Report Designer.

2 Clique na faixa do cabeçalho. Um componente de linha será criado.

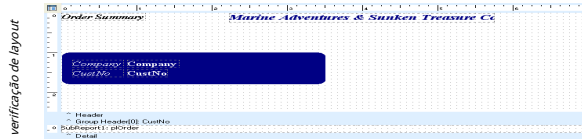
3 Clique com o botão direito sobre a linha e selecione a opção de menu ParentWidth. A linha será redimensionada para a largura da banda do cabeçalho.

4 Selecione a vista | Opção de menu Barras de ferramentas do menu principal do Report Designer e tornar a barra de ferramentas Desenhar visível.

5 Arraste a barra de ferramentas Desenhar para o lado esquerdo do Report Designer e encaixe-a abaixo da barra de ferramentas Alinhar ou Espaço.

6 Clique no ícone Espessura da Linha  no sorteio barra de ferramentas e selecione a espessura de 1 pt. A espessura da linha deve se ajustar de acordo.

7 Mantenha pressionada a tecla Ctrl e pressione a tecla de seta para baixo até que a linha esteja posicionada na parte inferior da banda de cabeçalho.



Esquematize a faixa de detalhes

1 Clique no ícone do componente DBText  no Paleta do componente Report Designer.

2 Clique na faixa de detalhes. Um componente DBText será criado.

3 Localize as duas listas suspensas no canto superior esquerdo do Report Designer. A lista à esquerda contém os pipelines de dados. Ele deve usar como padrão o pipeline de dados do cliente, pois esse é o pipeline de dados para o relatório. A lista à direita contém uma lista de campos para o pipeline de dados atualmente selecionado.

4 Selecione o campo Empresa na lista de campos. O nome da empresa deve ser exibido no componente DBText.

5 Defina a fonte do componente DBText:

Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Regular
Cor da fonte	Preto

6 Coloque mais cinco componentes DBText na faixa de detalhes. Atribua-os aos seguintes campos:

Contato

Telefone

Addr1

Cidade

Estado

7 Clique com o botão direito do mouse na régua horizontal e altere a unidade de medida para Pixels da tela.

8 Clique com o botão direito do mouse no componente DBText atribuído ao campo Empresa e acesse a caixa de diálogo Posição Defina os valores da posição:

Esquerda 2

4 principais

9 Acesse a caixa de diálogo Posição ... de cada componente DBText restante e defina a posição esquerda para o seguintes valores:

Contato	219
Telefone	378
Endereço	519
Cidade	739
Estado	896

10 Selecione o componente Company DBText e, em seguida, clique com a tecla Shift pressionada para selecionar os componentes DBText restantes.

11 Clique no ícone Alinhar Superior da barra de ferramentas Alinhar ou Espaço. Os componentes devem estar alinhados com o componente DBText da empresa.

12 Clique com o botão direito sobre cada um dos componentes e defina Autosize como True.

13 Clique com o botão direito sobre o espaço em branco da banda de detalhes e acesse a caixa de diálogo Posição

14 Defina a altura como 25. A banda de detalhes deve diminuir de altura.

Alinhe as etiquetas da faixa do cabeçalho verticalmente

1 Selecione o rótulo da empresa e clique com o botão direito sobre ele para acessar a caixa de diálogo Posição

2 Defina a posição superior para 65.

3 Mantenha pressionada a tecla Shift e selecione os componentes restantes do rótulo na banda do cabeçalho:

Contato

Telefone

Endereço

Cidade

Estado

4 Clique no ícone Alinhar Superior na barra de ferramentas Alinhar ou Espaço. As etiquetas devem estar alinhadas com a parte superior da etiqueta da empresa.

Alinhe as etiquetas da faixa do cabeçalho com os componentes DBText da faixa de detalhe

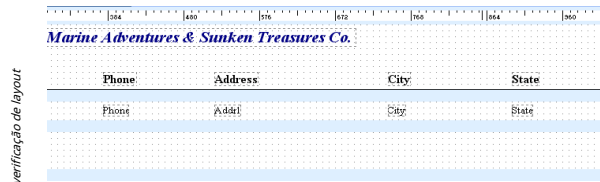
1 Selecione o componente Company DBText na banda de detalhes.

2 Mantenha pressionada a tecla Shift e clique no componente Rótulo da empresa na faixa do cabeçalho.

3 Clique no ícone Alinhar à Esquerda em Alinhar ou Barra de ferramentas do espaço. O rótulo e o componente DBText agora devem estar alinhados.

4 Repita este procedimento para os componentes DBText restantes e componentes Label correspondentes.

Certifique-se de selecionar o componente DBText primeiro. O primeiro componente selecionado determina a posição usada para o alinhamento.



Definir PrintDateTime

1 Clique com o botão direito sobre o espaço em branco da faixa de rodapé e acesse a caixa de diálogo Posição

2 Defina a altura para 40. A faixa de rodapé deve encolher de altura.

3 Clique no ícone do componente SystemVariable na paleta de componentes do Report Designer.

4 Clique no lado esquerdo da faixa de rodapé. Um componente de variável do sistema será criado.

5 Localize a lista suspensa no lado esquerdo do Report Designer. Ele deve conter uma lista de tipos de variáveis para o componente.

6 Selecione o item PrintDateTime nesta lista. A data e hora atuais serão exibidas no componente.

7 Defina a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Regular
Cor da fonte	Preto

8 Clique com o botão direito do mouse no componente SystemVariable e acesse a caixa de diálogo Posição

9 Defina os valores da posição:

Deixou	9
Principal	22

Definir números de página

1 Localize a barra de rolagem horizontal na parte inferior do Report Designer.

2 Role para a direita até que a borda direita da faixa de rodapé esteja visível.

3 Clique no ícone do componente SystemVariable na paleta de componentes do Report Designer.

4 Clique no lado direito da faixa de rodapé. Um componente SystemVariable deve ser criado.

5 Use a lista suspensa no lado esquerdo do Report Designer para selecionar o tipo de variável PageSetDesc. A página 1 de 1 deve ser exibida no componente.

6 Clique com o botão direito sobre o componente e acesse a caixa de diálogo Posição

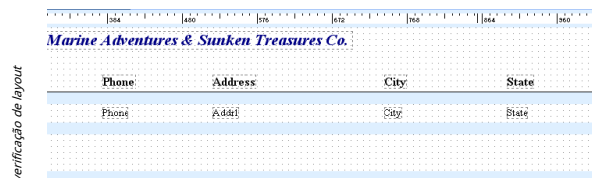
7 Defina os valores da posição:

Deixou	943
Principal	22

8 Justifique à direita o componente clicando no ícone Justificar à direita na barra de ferramentas Formatar.

9 Selecione Arquivo | Salve no menu principal do Delphi.

10 Clique na guia Visualizar no Report Designer. Um relatório de três páginas deve aparecer.



Visualize o relatório em tempo de execução

1 Feche o Report Designer.

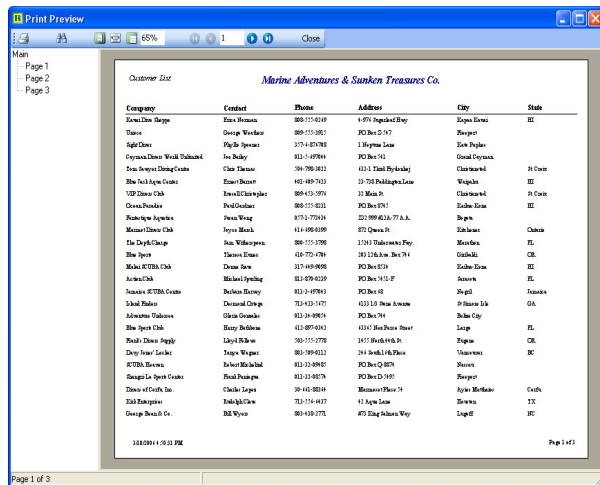
2 Selecione a guia Padrão da paleta de componentes Delphi.

3 Adicione um componente Botão ao formulário.

4 Configure o componente Botão:

Nome btnPreview
Antevisão da legenda

8 Clique no botão Visualizar. O relatório será exibido no formulário de visualização de impressão:



Grupos, cálculos e a faixa de resumo

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Divida um relatório em seções usando grupos
- Execute cálculos dentro de um relatório
- Utilize uma faixa de resumo

Gekko Investments

Stock Portfolio Analysis

rlRecommendation

PTT

GENERAL INFO	General				Prices				
	EXCHANGE	INDUSTRY	PROF ORTH	OUTLOOK	CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG	
NYSE	3726	45	3		\$160.88	\$116.38	\$49.00	\$39.00	
Val/Size				Recommendation					
P/E RATIO		RANK RATING		BETA	RECOMMENDATION	REK	SAFETY		
22.8		1 B		1.15	BUY	MED	3		

IPT

GENERAL INFO	General				Prices				
	EXCHANGE	INDUSTRY	PROF ORTH	OUTLOOK	CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG	
NYSE	3879	119	3		\$48.00	\$64.75	\$37.00	\$19.00	
Val/Size				Recommendation					
P/E RATIO		RANK RATING		BETA	RECOMMENDATION	REK	SAFETY		
16.8		3 A		1.3	BUY	MED	3		

MCOM

GENERAL INFO	General				Prices				
	EXCHANGE	INDUSTRY	PROF ORTH	OUTLOOK	CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG	
NYSE	3726	39	3		\$9.25	\$10.36	\$4.75	(\$2.00)	
Val/Size				Recommendation					
P/E RATIO		RANK RATING		BETA	RECOMMENDATION	REK	SAFETY		
26.4		3 C		1.15	BUY	HIGH	4		

3/10/00 5:43:40 PM

Page 1 of 39

5/10/00 5:43:40 PM

Page: 1 of 38

Criar uma nova aplicação

1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmStockSummary'.

3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbStock no diretório My RB Tutorials (estabelecido na página 229).

Nota: É importante nomear o formulário e salvar a unidade do formulário usando os nomes fornecidos acima porque o relatório será usado em tutoriais posteriores.

4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

6 Salve o projeto com o nome rbSTProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie um componente Query, DataSource e DataPipeline

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Consulta ao formulário.

3 Configure o componente Query:

Nome do banco de dados	DBDEMOS
SQL	SELECCION * FROM master ORDER BY rcmdation
Nome	qryStock

4 Clique duas vezes na propriedade Active no Object Inspector para defini-la como true.

5 Adicione um componente DataSource ao formulário.

6 Configure o componente DataSource:

DataSet	qryStock
Nome	dsStock

7 Selecione a aba RBuilder do Delphi compo-paleta atual.

8 Adicionar um componente DBPipeline  para o formulário.

9 Configure o componente DBPipeline:

DataSource	dsStock
Nome	plStock

Crie um relatório e conecte-o aos dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

DataPipeline	plStock
Nome	rbStock

Chame o Report Designer

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

3 Defina a orientação do papel como Paisagem (Arquivo | Configurar página | Tamanho do papel).

4 Arraste a banda do cabeçalho até a marca de uma polegada na régua vertical. Clique com o botão direito sobre o espaço em branco da banda do cabeçalho e acesse a caixa de diálogo Posição. O valor da altura deve ser 1.

Crie etiquetas na faixa de cabeçalho

1 Coloque uma etiqueta no lado esquerdo da faixa do cabeçalho.

2 Defina a legenda como Gekko Investments.

3 Defina a fonte e a posição:

Nome da fonte	Times New Roman
Tamanho da fonte	20
Estilo de fonte	Negrito itálico
Deixou	0,0417
Principal	0,0417

4 Selecione o rótulo.

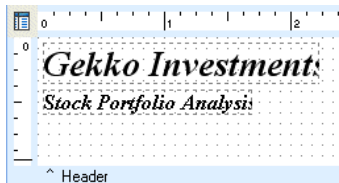
5 Copie o rótulo.

6 Cole o rótulo na banda do cabeçalho.

7 Defina a legenda deste rótulo para Análise de portfólio de ações.

8 Defina a fonte e a posição:

Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito itálico
Deixou	0,0417
Principal	0,4063

**Crie um grupo no campo 'rcmndation'**

1 Selecione Relatório | Grupos do menu principal. A caixa de diálogo Grupos será exibida.

2 A lista suspensa na parte superior da caixa de diálogo contém uma lista de campos da Consulta. Selecione o campo plStock.RCMNDATION e clique no botão Adicionar. Uma nova entrada intitulada Grupo1: plStock.RCMNDATION deve ser adicionado à lista.

3 Localize a caixa de seleção Iniciar nova página perto do meio da caixa de diálogo.

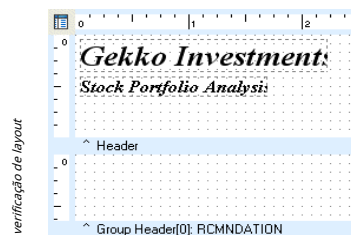
4 Marque esta caixa. Isso fará com que o relatório inicie uma nova página cada vez que o valor do campo RCMNDATION for alterado.

Nota: Como os grupos são baseados em uma mudança no valor do campo, o campo no qual um grupo se baseia deve sempre ser usado para ordenar os registros. Você deve se lembrar que a consulta para este relatório é ordenada pelo campo 'rcmndation'. Você deve sempre ordenar os registros pelos campos que pretende usar para grupos porque o ReportBuilder não classificará os registros para você.

5 Clique no botão OK. Você deve ver duas novas bandas: uma acima e outra abaixo da banda de detalhes. Essas faixas não têm altura; portanto, nenhum espaço em branco aparece acima das divisórias.

6 Clique com o botão direito sobre o divisor de banda do cabeçalho do grupo e selecione a opção de menu Posição

7 Defina a altura para 0,6458.

**Esquematize a faixa do cabeçalho do grupo**

1 Coloque um rótulo no lado esquerdo da banda do cabeçalho do grupo.

2 No Inspetor de objetos, defina o Nome do componente como rlRecommendation.

3 Defina a fonte e a posição:

Nome da fonte	Times New Roman
Tamanho da fonte	14
Estilo de fonte	Negrito
Deixou	0,0417
Principal	0,0937

4 Coloque um componente de linha próximo à parte inferior da banda do cabeçalho do grupo.

5 Clique com o botão direito sobre o componente de linha e selecione a opção ParentWidth.

6 Localize a lista suspensa Tipo de linha no canto superior esquerdo do Report Designer.

7 Selecione 'Bottom' nesta lista. Isso faz com que a linha seja desenhada na parte inferior do retângulo forma delimitadora.

8 Clique no ícone de espessura da linha na barra de ferramentas Desenhar e selecione 1 1/2 ponto.

9 Selecione a linha e use as teclas de seta para posicioná-la de forma que fique alinhada com a parte inferior da faixa do cabeçalho do grupo.



Codifique o Evento BeforeGenerate da Banda do Cabeçalho do Grupo

1 Clique no espaço em branco da faixa do cabeçalho do grupo. A banda deve ser selecionada no Object Inspector.

2 Clique na guia Events do Object Inspector e dê um clique duplo no evento BeforeGenerate. Adicione o código mostrado abaixo ao manipulador de eventos.

Nota: Esta rotina recupera o valor do campo de recomendação em uma variável de string local. Em seguida, atribui uma cor com base no valor do campo. A cor é armazenada em uma variável privada do formulário para que todos os manipuladores de eventos de relatório possam usá-la. O rótulo de recomendação na banda de cabeçalho do grupo é definido com a cor. Finalmente, a legenda do rótulo de recomendação é definida usando a propriedade FirstPage do grupo. Se FirstPage for False, o grupo transbordou para uma página adicional.

Código

Manipulador de eventos para o evento BeforeGenerate

```
procedimento TfrmStockSummary.ppGroupHeaderBand1BeforeGenerate (Sender: TObject);
var
    lsRecommendation: Fragmento;
começar
    lsRecommendation := pIStock ['Rcmndation'];

    {f (lsRecommendation = 'BUY') t en
    eu determinar a cor da fonte com base sobre o valor da recomendação}
    FRRecommendationColor := clGreen

    mais se (lsRecommendation = 'HOLD') então
        FRRecommendationColor := clOlive

    mais se (lsRecommendation = 'SELL') então
        FRRecommendationColor := clMaroon;

    rlRecommendation.Font.Color := FRRecommendationColor;

    {f rbStock.Groups [0] .FirstPage a
    eu verifique se a primeira página do grupo ou uma continuação}
    rlRecommendation.Caption := 'Recomendado' + lsRecommendation + 'Lista'

    senão
        rlRecommendation.Caption := 'Recomendado' + lsRecommendation + " 'continua ...'";

fim;
```

3 Role para cima no Editor de código e adicione o seguinte código à seção 'privada' da declaração de classe do formulário:

```
FRecommendationColor: TColor;
```

4 Selecione Exibir | Alterne Form / Unit no menu principal do Delphi.

5 Selecione a guia Standard da paleta Delphi Component.

6 Crie um botão no formulário.

7 Configure o botão:

Nome	btnPreview
Rubrica	Antevisão

8 Adicione o seguinte código ao evento OnClick do botão:

```
rbStock.Print;
```

9 Selecione Arquivo | Salve no menu principal do Delphi.

10 Execute o projeto. O relatório deve ter a seguinte aparência:



Adicionar dados gerais à faixa de detalhes

1 Acesse o Report Designer.

2 Clique com o botão direito sobre o espaço em branco da banda de detalhes e selecione a opção de menu Posição

3 Defina a altura como 1,8542.

4 Selecione Exibir | Barras de ferramentas | Árvore de dados. A janela da árvore de dados deve aparecer. Posicione-o sobre o Inspetor de objetos.

5 Clique na guia Layout na parte inferior da Árvore de dados e defina as configurações de arrastar e soltar:

Crio	Tudo
Estilo	Tabular
Grade de etiqueta	Falso
Nome da fonte do rótulo	Novos tempos
Estilo de fonte da etiqueta	Regular
Tamanho da fonte da etiqueta	10 pontos
Cor da fonte do rótulo	Preto
Grade de campo	Falso
Nome da fonte do campo	Novos tempos
Estilo da fonte do campo	Negrito
Tamanho da fonte do campo	10 pontos
Cor da fonte do campo	Preto

6 Clique na guia Dados e use o método Ctrl-clique para selecionar esses campos em ordem:

INTERCÂMBIO
INDÚSTRIA
PROJ_GRTH
PANORAMA

7 Arraste os campos selecionados para a banda de detalhes. Quatro componentes DBText e quatro rótulos associados serão criados.

8 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

Deixou	1.0729
Principal	0,3542

Nota: Conforme você move a seleção, a posição aparece na barra de status. Use essas medidas para determinar quando você alcançou a posição.

9 Coloque uma linha acima da seleção.

10 Configure o componente de linha:

Tipo de linha	Principal
Espessura	1 1/2 pt
Posição Esquerda	1.0729
Principal	0,2083
Largura	4,2604

11 Coloque um componente de rótulo sobre a linha.

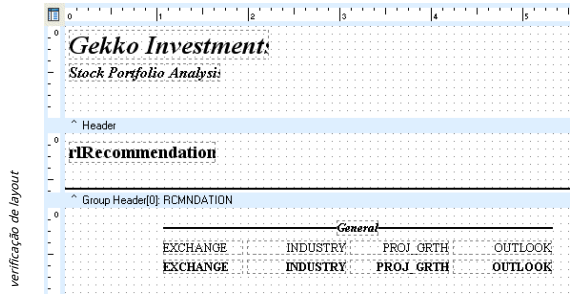
12 Configure o componente de rótulo:

Rubrica	Em geral
Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Negrito itálico
Alinhamento de Texto	Centrado
Posição superior	0,1354
Transparente	Falso

13 Selecione a linha e clique no rótulo com a tecla Shift pressionada. Clique

o ícone Alinhar ao Meio em Alinhar ou Espaço

barra de ferramentas.



Adicione os dados de estatísticas vitais à faixa de detalhes

1 Use o método Ctrl-clique para selecionar esses campos em ordem na Árvore de Dados:

P_E_RATIO
CLASSIFICAÇÃO
AVALIAÇÃO
BETA

2 Arraste os campos selecionados diretamente abaixo dos componentes Gerais. Quatro componentes DBText e quatro rótulos associados serão criados.

3 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

Deixou	1.0729
Principal	1,1875

4 Selecione a linha e o rótulo Geral.

5 Copie e cole a seleção. Uma nova linha e etiqueta devem aparecer logo abaixo dos originais.

6 Posicione esses dois componentes acima dos componentes recém-criados.

7 Defina o topo da linha para 1.0417.

8 Defina o topo do rótulo para 0,9687.

9 Clique com o botão direito sobre o rótulo e selecione o item de menu Trazer para a frente.

10 Defina a legenda do rótulo como Vital Stats.

11 Selecione o rótulo PE RATIO e, em seguida, clique com a tecla Shift pressionada linha.

12 Clique no ícone Alinhar à esquerda Barra de  do Alinhar ou ferramentas Espaço.

13 Selecione a linha e, em seguida, clique com a tecla Shift em Estatísticas vitais rótulo.

14 Clique no ícone Alinhar ao Meio ou na barra  do Alinhamento de ferramentas Espaço.

verificação de layout

Gekko Investments			
<i>Stock Portfolio Analysis</i>			
^ Header			
Recommendation			
^ Group Header[0]: RCMNDATION			
General			
EXCHANGE	INDUSTRY	PROJ GRTH	OUTLOOK
EXCHANGE	INDUSTRY	PROJ GRTH	OUTLOOK
Vital Stats			
P E RATIO	RANK RATING	BETA	
P E RATIO	RANK RATING	BETA	

Adicione os dados de preços à faixa de detalhes

1 Use o método Ctrl-clique para selecionar esses campos em ordem na Árvore de Dados:

CUR_PRICE
YRL_HIGH
YRL_LOW
PRICE_CHG

2 Arraste os campos selecionados para a direita imediata dos componentes Gerais. Quatro componentes DBText e quatro rótulos associados serão criados.

3 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

Deixou	5,7188
Principal	0,3542

4 Clique com o botão direito sobre cada um dos quatro componentes DBText e defina DisplayFormat como \$ #, 0,00; (\$ #, 0,00) (a primeira seleção com um cifrão).

5 Selecione a linha e o rótulo Geral.

6 Copie e cole a seleção.

7 Posicione a seleção acima dos componentes recém-criados.

8 Defina a legenda do rótulo como Preço.

k

Price			
CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG
CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG

Alinhar os componentes de dados de preços

1 Selecione o rótulo CUR_PRICE e, em seguida, o componente da linha.

2 Clique no ícone Alinhar à Esquerda Barra de  do Alinhar ou ferramentas Espaço.

3 Defina a largura da linha para 4,4479.

4 Selecione o rótulo Geral e, a seguir, o rótulo Preço.

5 Clique na barra de ferramentas Espaço do  do Alinhar ou ícone Alinhar superior.

- 6 Selecione a linha atrás da etiqueta Geral e, em seguida, a linha atrás da etiqueta Preços.
- 7 Clique no ícone Alinhar Superior.
- 8 Selecione a linha de preços e, em seguida, o preço rótulo.
- 9 Clique no ícone Alinhar ao Meio  do Alinhamento ou barra de ferramentas Espaço. A etiqueta deve ser centralizada horizontalmente em relação à linha.

e/e

Pricing			
CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG
CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG

Adicione os dados de recomendação à faixa de detalhes

- 1 Use o método Ctrl-clique para selecionar esses campos em ordem na Árvore de Dados:

RCMNDATION
RISCO
SEGURANÇA

- 2 Arraste os campos selecionados imediatamente abaixo dos componentes de preço.
- 3 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

Deixou	5,7188
Principal	1,1875

- 4 Feche a janela da árvore de dados.

- 5 Clique no espaço em branco da banda para desmarcar os componentes e, em seguida, selecione o componente DBText para o campo RCMNDATION.
- 6 Defina o nome deste componente como dbtRecommendation no Object Inspector. Usaremos esse nome em um manipulador de eventos.
- 7 Selecione a linha de preços e o rótulo.
- 8 Copie e cole a seleção.
- 9 Posicione a seleção acima dos componentes recém-criados.
- 10 Defina a legenda do rótulo como Recomendação.

verificação de layout

Pricing			
CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG
CUR PRICE	YRL HIGH	YRL LOW	PRICE CHG
Recommendation			
RCMNDATION RISK	SAFETY		
RCMNDATION RISK	SAFETY		

Alinhe os componentes de dados de recomendação

- 1 Selecione o rótulo RCMNDATION e, em seguida, o componente da linha.
- 2 Clique no ícone Alinhar à Esquerda Barra de  do Alinhar ou ferramentas Espaço.
- 3 Selecione o rótulo Vital Stats e, em seguida, o rótulo Recomendação.
- 4 Clique na barra de ferramentas Espaço do  do Alinhar ou ícone Alinhar superior.
- 5 Selecione a linha atrás do rótulo Vital Stats e, em seguida, a linha atrás do rótulo Recomendação.

6 Clique na barra de ferramentas Espaço do  do Alinhar ou ícone Alinhar superior.

7 Selecione a linha de recomendação e o rótulo.

8 Clique no ícone Alinhar ao Meio  do Alinhamento ou barra de ferramentas Espaço. A etiqueta deve ser centralizada horizontalmente em relação à linha.

Adicione o símbolo de ações e os dados da empresa à faixa de detalhes

1 Coloque um componente DBText no canto superior esquerdo da banda de detalhes

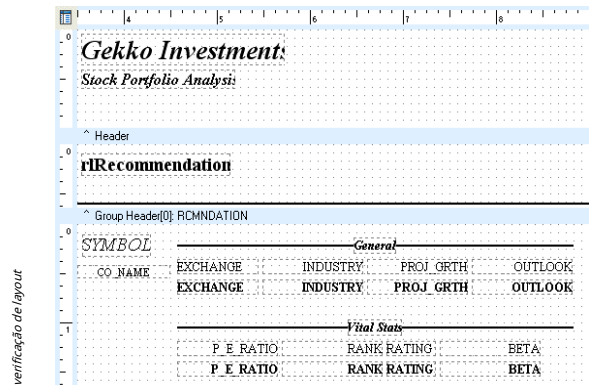
2 Configure o componente DBText:

Campo de dados	SÍMBOLO
Tamanho automático	Verdadeiro
Nome	dbtSymbol
Nome da fonte	Times New Roman
Tamanho da fonte	14
Estilo de fonte	itálico
Alinhamento de Texto	Justificado à esquerda
Deixou	0,0937
Principal	0,0104

3 Coloque outro componente DBText no canto superior esquerdo da banda de detalhes.

4 Configure o componente DBText:

Campo de dados	CO_NAME
Nome	dbtCompany
Nome da fonte	Times New Roman
Tamanho da fonte	8
Estilo de fonte	Regular
Deixou	0
Principal	0,4167
Largura	1



Atribuir o manipulador de eventos

BeforeGenerate da faixa de detalhes

1 Clique no espaço em branco da banda de detalhes.

2 Clique na guia Eventos do Inspetor de objetos.

3 Clique duas vezes no evento BeforeGenerate. Um shell de manipulador de eventos será gerado em seu formulário Delphi.

4 Adicione o código conforme mostrado abaixo.

Código	Defina a cor da fonte
<pre>{definir a cor da fonte do símbolo e recomendação} dbtRecommendation.Font.Color:= FRecommendationColor; dbtSymbol.Font.Color:= FRecommendationColor;</pre>	

Nota: Esta rotina define a cor do símbolo de ação e o campo de recomendação para a cor definida no evento BeforeGenerate da banda do cabeçalho do grupo.

5 Selecione Arquivo | Salve no menu principal do Delphi.

6 Selecione o projeto | Compile o rbSTProj do menu principal do Delphi. Corrija quaisquer problemas.

7 Execute o projeto.

8 Visualize o relatório. Clique no ícone Última página da janela Visualizar impressão. Um relatório de trinta e sete páginas deve aparecer. O rótulo no cabeçalho do grupo deve mudar de cor com base no grupo. O símbolo de ação e os campos de recomendação também devem ser codificados por cores.

Esquematize a faixa de rodapé

1 Acesse o Report Designer.

2 Coloque um componente de variável de sistema no lado esquerdo da faixa de rodapé.

3 Configure o componente SystemVariable:

VarType	PrintDateTime
Alinhamento de Texto	Justificado à esquerda
Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Regular
Cor da fonte	Preto
Deixou	0
Principal	0,2292

4 Role para a direita até ver a borda da faixa de rodapé.

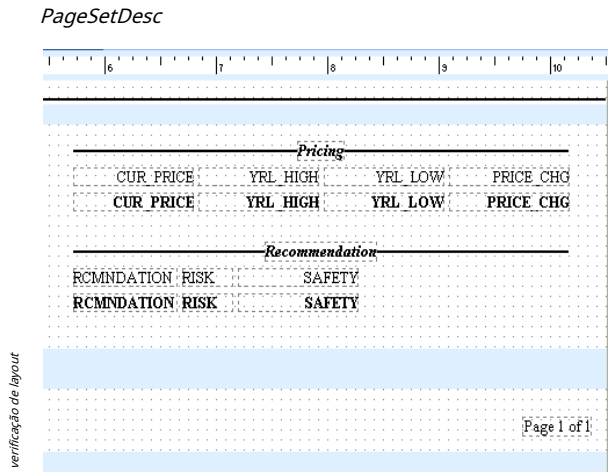
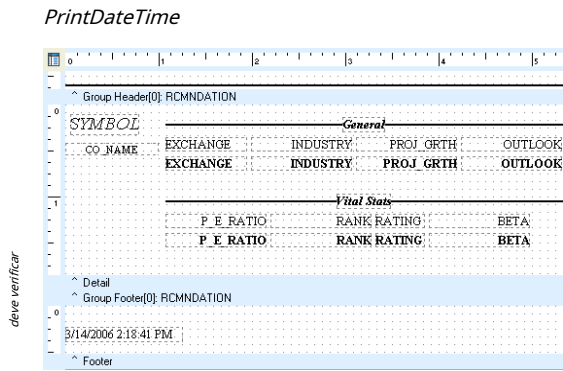
5 Crie um componente SystemVariable no lado direito da faixa de rodapé.

6 Configure o componente SystemVariable:

VarType	PageSetDesc
Alinhamento de Texto	Justificado corretamente
Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Regular
Cor da fonte	Preto
Deixou	9,75
Principal	0,2292

7 Selecione Arquivo | Salve no menu principal do Delphi.

8 Visualize o relatório executando o aplicativo.



Esquematize a Faixa de Resumo

1 Retorne ao Report Designer.

2 Selecione Relatório | Resumo do menu principal do Report Designer. Uma faixa de resumo deve aparecer como a faixa mais inferior no Report Designer.

3 Clique com o botão direito sobre o espaço em branco da faixa de resumo e selecione a opção de menu Posição

4 Defina a altura para 3,4063.

5 Role para baixo para ver toda a faixa de resumo.

6 Coloque um componente de etiqueta no lado esquerdo da faixa de resumo.

7 Configure o componente de rótulo:

Rubrica	Página de Resumo
Nome da fonte	Times New Roman
Tamanho da fonte	16
Estilo de fonte	Negrito
Cor da fonte	Preto
Alinhamento de Texto	Justificado à esquerda
Deixou	0
Principal	0,0313

8 Clique com o botão direito do mouse no espaço em branco da faixa de resumo e selecione a opção de menu Nova página. Isso fará com que a faixa de resumo seja impressa em uma nova página no final do relatório.

9 Coloque um componente de rótulo próximo ao centro da faixa de resumo.

10 Configure o componente de rótulo:

Rubrica	Recomendações
Nome da fonte	Times New Roman
Tamanho da fonte	16
Estilo de fonte	Regular
Cor da fonte	Preto
Alinhamento de Texto	Justificado à esquerda
Posição Esquerda	1.625
Principal	0,6667

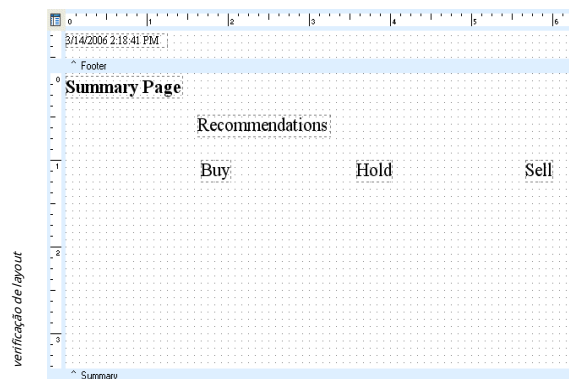
11 Coloque mais três componentes de etiqueta na faixa de resumo.

12 Defina as legendas:

Comprar

Segure

Vender



Ajuste os rótulos da banda de resumo

1 Coloque a etiqueta Comprar na seguinte posição:

Esquerda 2.8750
1.0208 principais

2 Coloque a etiqueta de venda na seguinte posição:

2.0104 principais

3 Selecione os rótulos Comprar, Reter e Vender naquele pedido.

4 Clique no ícone Alinhar à direita Barra de  no Alinhar ou ferramentas Espaço.

5 Clique no ícone Espaço verticalmente ou na barra  no alinhamento de ferramentas Espaço.

6 Clique no ícone Justificar à direita na barra de ferramentas Formato.

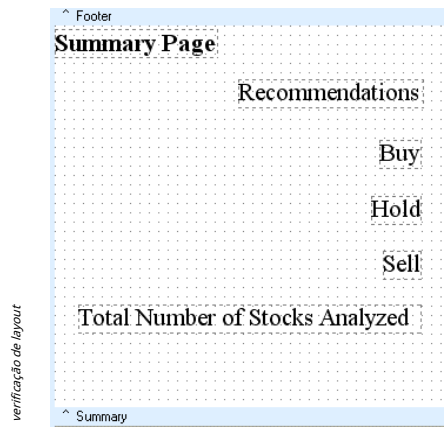
7 Coloque um componente de rótulo próximo à parte inferior da faixa de resumo.

8 Configure o componente de rótulo:

Rubrica	Número total de
	Estoque analisado
Nome da fonte	Times New Roman
Tamanho da fonte	16
Estilo de fonte	Regular
Posição superior	2,4896

9 Clique no rótulo Vender e, em seguida, clique no novo rótulo com a tecla Shift pressionada.

10 Clique no ícone Alinhar à direita Barra de  no Alinhar ou ferramentas Espaço.



Criar e ajustar componentes variáveis

1 Coloque três componentes variáveis à direita dos rótulos Buy, Hold e Sell.

2 Defina os nomes como vrBuyTotal, vrHoldTotal e vrSellTotal no Inspetor de objetos.

3 Use a lista suspensa na barra de ferramentas Editar para definir o DataType de cada uma das variáveis como Inteiro.

4 Selecione a variável vrBuyTotal e defina a posição:

Deixou 3,5521

5 Clique com a tecla Shift pressionada nos componentes das variáveis vrHoldTotal e vrSellTotal.

6 Clique na barra de ferramentas  ícone no Alinhar Alinhar à direita ou Espaço.

7 Clique no ícone Justificar à direita na barra de ferramentas Formato.

8 Selecione o rótulo Comprar e, em seguida, clique com a tecla Shift pressionada na variável vrBuyTotal.

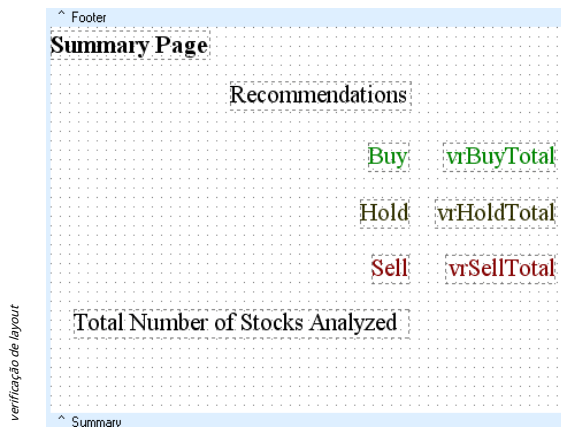
9 Clique no ícone Alinhar Superior  do Alinhar ou Barra de ferramentas do espaço. Alinhe as variáveis Hold e Sell com os rótulos correspondentes.

10 Selecione o rótulo e a variável Comprar e defina a cor da fonte como verde.

11 Selecione o rótulo e a variável Hold e defina a cor da fonte como oliva.

12 Selecione o rótulo e a variável Vender e defina a cor da fonte para marrom.

13 Selecione Arquivo | Salve no menu principal do Delphi.



Atribuir manipuladores de eventos aos eventos OnCalc dos componentes variáveis

1 Selecione a variável vrBuyTotal.

2 Clique na guia Eventos do Inspetor de objetos.

3 Clique duas vezes no evento OnCalc. Um shell de manipulador de eventos será gerado em seu formulário Delphi.

4 Atribua o seguinte código a este manipulador de eventos:

```
E se(qryStock.FieldByName ('Rcmndation'). AsString = 'COMPRAR') então  
    Valor: = Valor + 1;
```

5 Atribua o seguinte código ao evento OnCalc da variável vrHoldTotal:

```
E se(qryStock.FieldByName ('Rcmndation'). AsString = 'HOLD') então  
    Valor: = Valor + 1;
```

6 Atribua o seguinte código ao evento OnCalc da variável vrSellTotal:

```
E se(qryStock.FieldByName ('Rcmndation'). AsString = 'VENDER') então  
    Valor: = Valor + 1;
```

Observação: todos os três manipuladores de eventos verificam o valor do campo para certificar-se de que corresponde ao total e, em seguida, incrementam o total. O evento OnCalc é um procedimento onde o parâmetro Value contém o valor atual da variável. Essas variáveis específicas são inteiros porque definimos DataType como Integer quando as criamos.

7 Selecione Arquivo | Salve no menu principal do Delphi.

8 Selecione o projeto | Compile o rbSTProj do menu principal do Delphi. Corrija quaisquer problemas.

9 Execute o projeto.

10 Visualize o relatório. Clique no ícone Última página da janela Visualizar impressão. Um relatório de trinta e oito páginas deve aparecer. A faixa de resumo deve conter totais para cada tipo de recomendação.

Nota: Você deve compilar e executar este relatório para ver exatamente como ele se parecerá. Isso ocorre porque os manipuladores de eventos foram atribuídos ao relatório. Quando manipuladores de eventos são atribuídos, o relatório deve ser compilado e executado; caso contrário, o código do manipulador de eventos não será usado ao gerar o relatório e quaisquer alterações feitas por meio dos manipuladores de eventos não serão refletidas no relatório.

Crie o Grande Total

1 Retorne ao Report Designer.

2 Coloque um componente DBCalc  no fundo do na faixa de resumo.

3 Clique com o botão direito sobre o componente e acesse a opção de menu Cálculos

4 Selecione a função Contagem na lista suspensa Tipo de Calc e clique no botão OK.

5 Configure o componente:

Nome da fonte	Times New Roman
Tamanho da fonte	16
Estilo de fonte	Negrito
Cor da fonte	Preto
Alinhamento de Texto	Justificado corretamente
Largura	1.0729

6 Clique na variável vrSellTotal e, em seguida, clique com Shift no componente DBCalc.

7 Clique no ícone Alinhar à direita ou na barra  do Alinhamento de ferramentas Espaço.

8 Selecione o rótulo Total ... e, em seguida, o DBCalc componente.

9 Clique na barra de ferramentas Espaço do  do Alinhar ou ícone Alinhar superior.

10 Coloque um componente de linha na parte inferior da faixa de resumo.

11 Clique com o botão direito sobre o componente da linha e selecione a opção de menu Duplo. A linha agora deve aparecer como duas linhas paralelas.

12 Defina o topo da linha para 2,3437.

13 Selecione o componente DBCalc e clique na linha com a tecla Shift pressionada.

14 Selecione a vista | Barras de ferramentas | Opção de menu Tamanho no menu principal do Report Designer.

15 Encaixe a barra de ferramentas Tamanho no lado esquerdo inferior do Report Designer.

16 Clique no ícone Aumentar a largura para o maior  de a barra de ferramentas Tamanho. A linha agora deve ter a mesma largura do DBCalc.

17 Clique no ícone Alinhar à direita Barra de  no Alinhar ou ferramentas Espaço.

Usando regiões para agrupar componentes dinâmicos de maneira lógica

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Use regiões
- Use um memorando de alongamento
- Posicione os componentes em relação a um memorando de alongamento

Clown Triggerfish	Red Emperor
	
Also known as the big spotted triggerfish, inhabits outer reef areas and feeds upon invertebrates and mollusks by crushing them with powerful beak. They are voracious eaters, and divers report seeing the clown triggerfish devour heads of peer species. Do not eat this fish. According to an 1875 account, "the poisonous fluid acts primarily upon the nervous tissue of the skin, occasioning violent spasms of that organ, and shortly afterwards all the muscles of the body. The flesh becomes roiled with spasm, the tongue blackened, the eye fixed, the breathing laborious, and the patient expires in a paroxysm of terrible suffering." Not edible. Range is Indo-Pacific and East Africa to South.	Called wasper in Australia, inhabits the area around lagoon coral reefs and sandy bottom. The red emperor is a valuable food fish and considered a great sporting fish that fights with fury when hooked. The flesh of an old fish is just as tender as that of the very young. Range is from the Indo-Pacific to Australia.
Category: Triggerfish Common Name: Clown Triggerfish Species Name: <i>Pomacentrus coelestis</i> Length: 30	Category: Triggerfish Common Name: Red Emperor Species Name: <i>Uranoscopus scaber</i> Length: 24

5/16/00 11:09:43 AM

Criar uma nova aplicação

- 1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.
 - 2 Defina o nome do formulário como 'frmRegions'.
 - 3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbRegion no diretório My RB Tutorials (estabelecido na página 181).
 - 4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.
 - 5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.
 - 6 Salve o projeto com o nome rbRGProj no diretório My RB Tutorials.
- Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie uma tabela, uma fonte de dados e um componente DataPipeline

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblBiolife
Nome da tabela	biolife.db

4 Selecione o componente Tabela e, a seguir, clique duas vezes na propriedade Active no Inspetor de objetos para defini-la como verdadeira.

5 Adicione um componente DataSource ao formulário.

6 Configure o componente DataSource:

DataSet	tblBiolife
Nome	dsBiolife

7 Selecione a aba RBuilder do Delphi compo-paleta atual.

8 Adicionar um componente DBPipeline  para o formulário.

9 Configure o componente DBPipeline:

Fonte de dados	dsBiolife
Nome	plBiolife

Crie um relatório e conecte-o aos dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

DataPipeline	plBiolife
Nome	rbRegions

Chame o Report Designer

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

3 Defina a orientação do papel como Paisagem (Arquivo | Configurar página | Tamanho do papel).

4 Clique na guia Layout da caixa de diálogo Configurar página e defina as colunas como 2.

5 Clique no botão OK para fechar a caixa de diálogo.

Configure as bandas de cabeçalho e rodapé

1 Defina a altura das bandas de cabeçalho e rodapé para 0,3.

2 Role para a direita no Report Designer até que você possa ver a borda direita das bandas.

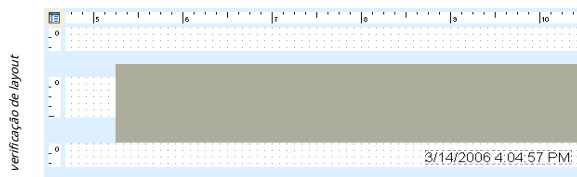
3 Coloque um componente SystemVariable no lado direito da faixa de rodapé.

4 Localize a lista suspensa no canto superior esquerdo do Report Designer.

5 Selecione a opção PrintDateTime na lista suspensa.

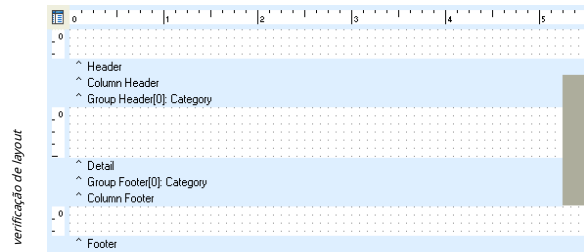
6 Configure o componente:

Nome da fonte	Arial
Tamanho da fonte	12
Estilo de fonte	Regular
Cor da fonte	Preto
Deixou	8,7188
Principal	0,0833
Alinhamento de Texto	Justificado corretamente

**Crie um grupo no campo 'Categoria'**

- 1 Selecione Relatório | Grupos do menu principal.
- 2 Selecione o campo 'plBiolife.Category' na lista suspensa no topo da caixa de diálogo.
- 3 Clique no botão Adicionar. Um novo grupo será adicionado à lista.
- 4 Clique na caixa de seleção Iniciar nova coluna na seção 'Mudança de grupo' da caixa de diálogo.
- 5 Esta opção força uma nova coluna cada vez que o grupo muda, o que permite que cada um dos registros da tabela Biolife apareça em uma coluna separada.
- 6 Clique em OK para fechar a caixa de diálogo Grupos.
- 7 Role para a esquerda.

Nota: Você pode notar que o cabeçalho do grupo e as bandas de rodapé do grupo criadas para o grupo Categoria têm a mesma largura do cabeçalho da coluna, rodapé da coluna e bandas de detalhes. Isso se deve ao fato de as bandas de grupo serem impressas antes e depois da banda de detalhe. Para imprimir antes e depois da banda de detalhe, as bandas de grupo devem ter a mesma largura que a banda de detalhe.

**Criar uma região de imagem**

- 1 Defina a altura da banda de detalhe para 3,5104.

2 Coloque um componente de região (do

Barra de ferramentas avançada) na parte superior da faixa de detalhes.

Observação: uma região é um tipo especial de componente que pode conter outros componentes. A região tem o mesmo comportamento de impressão de uma forma retangular, pois pode imprimir um retângulo colorido com uma borda de um pixel. A área sombreada cinza que você vê ao redor da borda da região não é impressa, mas é fornecida para que você possa distinguir uma região de um componente de forma.

3 Configure a região:

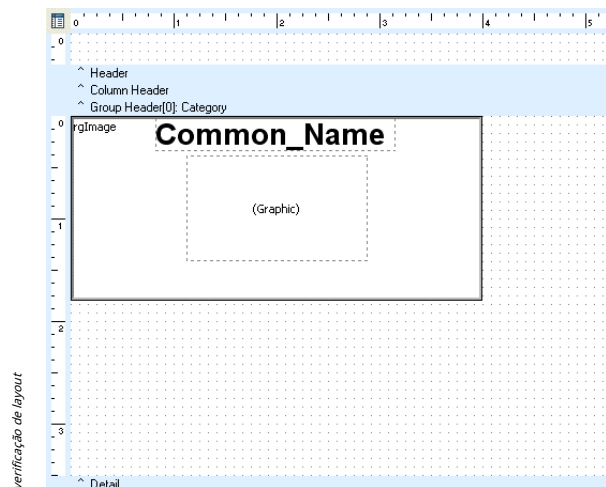
Nome	rgImage
Deixou	0
Principal	0
Largura	4
Altura	1,8021

4 Selecione a vista | Barras de ferramentas | Árvore de dados.**5** Posicione a janela Data Tree à esquerda do Report Designer (na frente do Object Inspector).**6** Clique na guia Layout da Árvore de Dados e clique no botão de opção Campos.**7** Clique na guia Dados.**8** Selecione os campos Common_Name e Graphic.**9** Arraste a seleção para o espaço em branco da região. Se o componente Imagem for criado fora da região, arraste-o para a região abaixo do componente DBText.**10** Clique no espaço em branco da região para desmarcar os componentes.**Ajustar componentes da região da imagem****1** Configure o componente DBText:

Tamanho automático	Verdadeiro
Nome da fonte	Arial
Tamanho da fonte	20
Estilo de fonte	Negrito
Cor da fonte	Preto
Deixou	0,8229
Principal	0,0208

2 Configure o componente DBImage:

Deixou	1,125
Principal	0,3854
Largura	1.7604
Altura	1.0313
MaintainAspectRatio	Verdadeiro

3 Selecione Arquivo | Salve no menu principal do Delphi.

Criar uma região de memorando

1 Coloque uma região imediatamente abaixo da imagem região.

2 Configure a região:

Nome	rgMemo
Deixou	0
Principal	1,8021
Largura	4
Altura	0,4687

Nota: Observe que a região tem uma borda preta de um pixel de largura. Você pode desativar essa borda definindo Line Color como 'No Line' no Report Designer ou definindo Pen.Style como psClear no Object Inspector.

3 Selecione o campo Notas na lista de campos da Árvore de Dados.

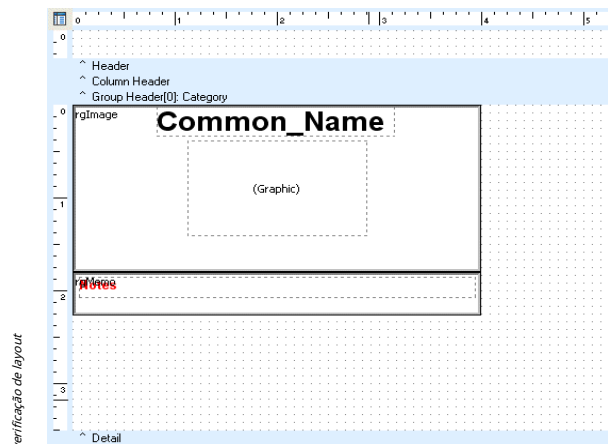
4 Arraste o campo Notas da Árvore de dados para o espaço em branco da região do memorando.

5 Configure o componente DBMemo:

Nome	dbmNotes
Alongamento	Verdadeiro
Nome da fonte	Arial
Tamanho da fonte	10
Estilo de fonte	Negrito
Cor da fonte	vermelho
Deixou	0,0625
Principal	1,8541
Largura	3,8854
Altura	0,2292
Deslocamento inferior	0,0208

6 Selecione Arquivo | Salve no menu principal do Delphi.

Nota: Ao definir a propriedade Stretch como True, estamos informando o Memo para expandir ou contrair com base na quantidade de texto no campo Notas. O texto será quebrado dentro da largura do componente. A propriedade Bottom Offset determina a quantidade de espaço em branco que aparecerá abaixo do memorando após a conclusão da impressão.

**Criar uma região de campos**

1 Coloque uma região no espaço imediatamente abaixo a região de memorando.

2 Configure a região:

Nome	rgFields
Deixou	0
Principal	2.2708
Largura	4
Altura	1.2396

3 Clique na guia Layout da Árvore de Dados.

4 Defina as configurações de layout:

Crio	Tudo
Estilo	Vertical
Nome da fonte do rótulo	Arial
Estilo de fonte da etiqueta	Negrito
Tamanho da fonte da etiqueta	10
Cor da fonte do rótulo	Preto
Nome da fonte do campo	Arial
Estilo da fonte do campo	Regular
Tamanho da fonte do campo	10
Cor da fonte do campo	Preto

5 Clique na guia Dados para retornar à lista de campos.

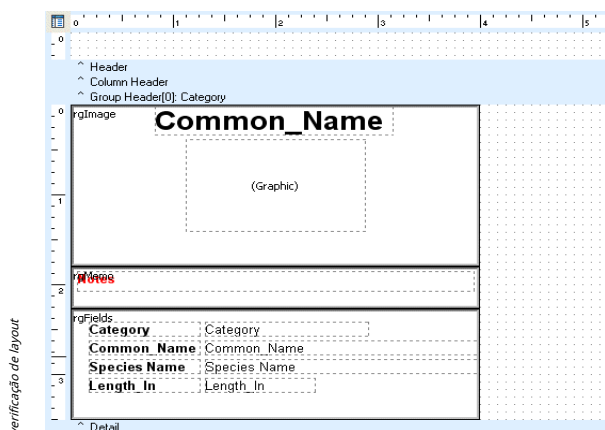
6 Selecione os seguintes campos e arraste a seleção para a região dos campos:

Categoria
 Nome comum
 Nome da Espécie
 Comprimento_In

7 Mantenha pressionada a tecla Ctrl e use a tecla de seta para posicione a seleção:

Deixou 0,1771
 Principal 2,4166

8 Feche a árvore de dados.



Ajuste os componentes DBText

1 Clique com o botão direito sobre o componente DBText

'Length_In' e defina DisplayFormat como

#, 0; - #, 0 (o primeiro DisplayFormat na lista).

2 Selecione todos os componentes nesta região.

3 Defina a propriedade AutoSize como True no Inspetor de objetos.

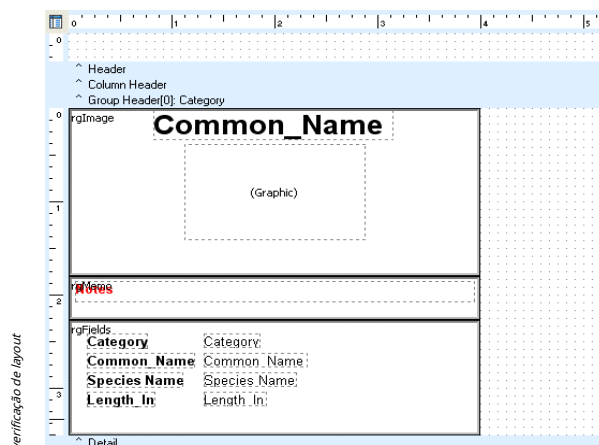
4 Clique com o botão direito sobre o espaço em branco da região e selecione a opção de menu ShiftRelativeTo

5 Selecione a região rgMemo na lista suspensa e clique no botão OK.

Observação: definimos a propriedade Stretch do componente DBMemo como True. Isso significa que cada vez que o memorando é impresso, a altura se expande ou se contrai com base na quantidade de texto no campo Notas. Como o memorando está contido em uma região, essa região também se estenderá para acomodar o memorando. Quando isso acontecer, queremos posicionar os campos contendo os componentes DBText imediatamente abaixo da região do Memorando. Fazemos isso definindo a propriedade ShiftRelativeTo da região Fields para a região Memo.

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Feche o Report Designer.



Adicionar um manipulador de eventos de codificação de cores

1 Adicione a seguinte variável à seção privada do formulário:

```
FColors: TList;
```

2 Adicione o seguinte código ao evento OnCreate do formulário:

3 Adicione o seguinte código ao evento OnDestroy do formulário:

```
procedimento TfrmRegions.FormStroy
(Sender: TObject);
começar
    FColors.Free;
fim;
```

Código	Manipulador de eventos para o evento OnCreate
	<pre>procedimento TfrmRegions.FormCreate (Sender: TObject); começar FColors := TList.Create; FColors.Add (TObject (clYellow)); FColors.Add (TObject (clRed)); FColors.Add (TObject (clLime)); FColors.Add (TObject (clAqua)); FColors.Add (TObject (clRed)); FColors.Add (TObject (clSilver)); FColors.Add (TObject (clYellow)); FColors.Add (TObject (clRed)); FColors.Add (TObject (clGray)); FColors.Add (TObject (clGreen)); FColors.Add (TObject (clOlive)); FColors.Add (TObject (clGray)); FColors.Add (TObject (clSilver)); FColors.Add (TObject (clMaroon)); FColors.Add (TObject (clSilver)); FColors.Add (TObject (clAqua)); FColors.Add (TObject (clRed)); FColors.Add (TObject (clSilver)); FColors.Add (TObject (clAqua)); FColors.Add (TObject (clMaroon)); FColors.Add (TObject (clGray)); FColors.Add (TObject (clBlue)); FColors.Add (TObject (clYellow)); FColors.Add (TObject (clRed)); FColors.Add (TObject (clSilver)); FColors.Add (TObject (clMaroon)); FColors.Add (TObject (clYellow)); fim;</pre>

- 4 Clique duas vezes no componente Relatório para acessar o Designer de Relatórios.
- 5 Clique no espaço em branco da banda de detalhes.
- 6 Selecione a guia Eventos do Inspetor de objetos e clique duas vezes no evento BeforeGenerate. Adicione o seguinte código:

3 Configure o componente Botão:

Nome	btnPreview
Rubrica	Antevisão
4 Adicione o seguinte código ao manipulador de eventos OnClick do botão:	
rbRegions.Print;	

Código	Manipulador de eventos para o evento BeforeGenerate
<pre>procedimento TfrmRegions.ppDetailBand1BeforeGenerate (Sender: TObject); var IColor: TColor; começar se não(rbRegions.DetailBand.Overflow) então começar IColor := TColor (FColors [plBiolife.TraversalCount]); rgFields.Brush.Color := IColor; dbmNotes.Font.Color := IColor; fim; fim;</pre>	

Nota: Este manipulador de eventos define a cor da região Fields e a fonte Memo. As cores foram escolhidas a dedo para corresponder às imagens de bitmap. Como há apenas um registro por coluna, a propriedade DataPipeline.TraversalCount (que rastreia o número de registros percorridos) pode ser usada para mapear a cor do objeto TList para a coluna.

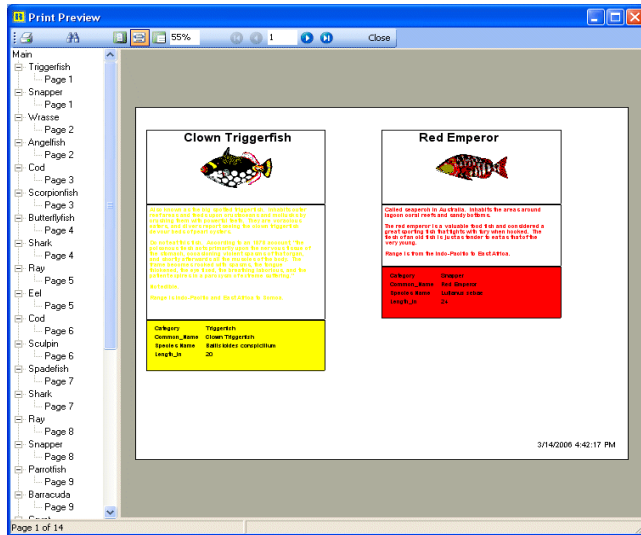
Visualize o relatório em tempo de execução

- 1 Selecione a guia Padrão da paleta de componentes Delphi.
- 2 Adicione um componente Botão ao formulário.

- 5 Selecione o projeto | Compile rbRGProj. Corrija quaisquer problemas de compilação.
- 6 Selecione Arquivo | Salve do menu principal do Delphi
- 7 Execute o projeto ..

8 Clique no botão Visualizar. Clique no último

Ícone da página. O relatório deve ser quatorze páginas, com dois peixes por página. A fonte Memo e a região inferior devem ser codificadas por cores. O relatório deve ter a seguinte aparência:



a. Control number	22222	VOID ☐	For Official Use Only OMB No. 1545-0007			
b. Employer's identification number 78-5676-809898			1. Wages, tips, other compensation		2. Federal income tax withheld	
c. Employer's name, address, and ZIP code Digital Metaphors Corporation 16775 Addison Rd., Suite 613 Addison, TX 75001			3. Social security wages		4. Social security tax withheld	
			5. Medicare wages and tips		6. Medicare tax withheld	
			7. Social security tips		8. Alcohol-related tips	
d. Employee's social security number 000-00-1221			9. Advance EIC payment		10. Dependent care benefits	
e. Employee's name (last, middle initial, first) Erica Norman			11. Nonqualified plans		12. Benefits included in box 1	
Digital Metaphors Corporation 16775 Addison Rd., Suite 613 Addison, TX 75001			13. See instructions for box 13		14. Other	
f. Employee's address and ZIP code 13 State Employee's state 14 D. No.			15 Sick leave, hr., etc.		16 Sick leave in	
			17 Total income in		18 Unemployment	
			19 Unemployment		20 Unemployment, hr., etc.	
			21 Unemployment in			

Wage and Tax Statement

Copy A for Social Security administration

Cat. No. 10134D

Department of the Treasury—Internal Revenue Service

For Paperwork Reduction Act Notice, see [esp and instructions](#).

DO NOT Cut or Separate Forms on This Page

a. Control number	22222	VOID ☐	For Official Use Only OMB No. 1545-0007			
b. Employer's identification number 78-5676-809898			1. Wages, tips, other compensation		2. Federal income tax withheld	
c. Employer's name, address, and ZIP code			3. Social security wages		4. Social security tax withheld	

Crie uma tabela, uma fonte de dados e um componente DataPipeline

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblCustomer
Nome da tabela	customer.db

4 Adicione um componente DataSource ao formulário.

5 Configure o componente DataSource:

DataSet	tblCustomer
Nome	dsCustomer

6 Selecione a aba RBuilder do Delphi compo-
paleta atual.

7 Adicionar um componente DBPipeline  para o formulário.

8 Configure o componente DBPipeline:

DataSource	dsCustomer
Nome	plCustomer

Crie um relatório e conecte-o ao
Dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

DataPipeline	plCustomer
Nome	rbFormEmu

Criar campos calculados

1 Clique duas vezes no componente da tabela. O Editor de campo será exibido.

2 Clique com o botão direito sobre o espaço em branco do Editor de campo e selecione a opção de menu Adicionar campos Adicione todos os campos listados.

3 Clique com o botão direito sobre o espaço em branco do Editor de campo e selecione a opção de menu Novo campo

4 Digite Salários na caixa de edição de nome. Defina o DataType como moeda. O Tipo de campo deve ser calculado como padrão.

5 Clique OK.

6 Repita esse processo para criar os seguintes campos:

SSWages
MedicareWages
FederalTaxWithheld
SSTaxWithheld
MedicareTaxWithheld

7 Feche o Editor de campo. Selecione o componente da tabela.

8 Clique na guia Eventos do Inspetor de objetos.

9 Clique duas vezes no evento OnCalcFields. Um manipulador de eventos será gerado. Adicione o código conforme mostrado abaixo.

Código	Manipulador de eventos para o evento OnCalcFields
<pre>procedimento TfrmFormsEmulation.tblCustomerCalcFields (DataSet: TDataSet); começar tblCustomerWages.AsCurrency: = 40000; tblCustomerSSWages.AsCurrency: = 40000; tblCustomerMedicareWages.AsCurrency: = 40000; tblCustomerFederalTaxWithheld.AsCurrency: = 40000 * 0,2; tblCustomerSSTaxWithheld.AsCurrency: = 40000 * 0,05; tblCustomerMedicareTaxWithheld.AsCurrency: = 40000 * 0,01; fim;</pre>	

Nota: Este manipulador de eventos é executado sempre que a posição do registro da tabela do cliente muda. Os cálculos colocados neste evento devem ser sempre intra-registro, o que significa que os valores devem ser calculados a partir de outros valores no registro atual ou devem ser codificados permanentemente (como neste caso).

10 Clique duas vezes no componente DBPipeline para exibir o Editor de campo.

11 Verifique se os campos calculados estão listados. Se não estiverem, feche o Editor de campo; defina a propriedade AutoCreateFields como False no Inspetor de objetos e, em seguida, defina-a novamente como True. Inicie o Editor de campo novamente e certifique-se de que os novos campos estejam listados.

Observação: alternar a propriedade AutoCreateFields para DBPipeline atualizará a lista de campos sempre que você alterar os campos no conjunto de dados.

12 Selecione o projeto | Compile o rbFEProj do menu principal do Delphi. Corrija quaisquer erros de compilação.

13 Selecione Arquivo | Salve no menu principal do Delphi.

Configure o tamanho da página e as bandas

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

3 Selecione Relatório | Cabeçalho do menu principal do Report Designer. A banda do cabeçalho será removida.

4 Selecione Relatório | Rodapé do menu principal do Report Designer. A faixa de rodapé será removida.

5 Clique no ícone de seleção de relatório  localizado na intersecção das régua horizontal e vertical. O relatório será selecionado no Object Inspector.

6 Expanda a propriedade PrinterSetup e defina todas as margens (MarginTop, MarginBottom, etc.) como zero.

7 Defina a altura da banda de detalhes para 11.

Criar um componente de imagem

1 Coloque um componente de imagem  no lado esquerdo da banda de detalhes.

2 Configure o componente de imagem:

Centro	Falso
DirectDraw	Verdadeiro
MaintainAspectRatio	Verdadeiro
Alongamento	Verdadeiro

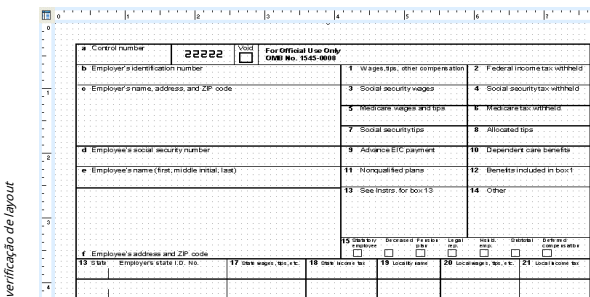
Nota: Queremos que a imagem seja redimensionada porque é um pouco grande para caber na página, apesar das margens zero. Não queremos distorcer o formulário de forma alguma, então MaintainAspectRatio foi definido como True. O centro não é necessário porque queremos imprimir o formulário na posição 0,0 (esta imagem do formulário já contém um espaço em branco ao redor das bordas). Finalmente, a propriedade DirectDraw forçará a imagem a ser impressa diretamente na impressora sem utilizar uma tela intermediária, o que geralmente resulta em uma saída de qualidade superior.

3 Clique com o botão direito do mouse no componente Imagem e selecione a opção Imagem ... do menu.

4 Abra o arquivo W2.WMF. Este arquivo deve estar localizado no diretório RBuilder \ Tutoriais.

5 Defina a posição do componente da imagem:

Deixou	0
Principal	0
Largura	7,8125
Altura	10.9792



Crie os componentes Wages DBText

1 Selecione Exibir | Barras de ferramentas | Árvore de dados no menu principal do Report Designer.

2 Posicione a Árvore de dados à esquerda do Report Designer (na frente do Object Inspector).

3 Clique na guia Layout da Árvore de Dados e configure-o:

Crio	Campos
Estilo	Vertical
Nome da fonte do campo	Courier New
Estilo da fonte do campo	Regular
Tamanho da fonte do campo	10
Cor da fonte do campo	Preto

4 Clique na guia Dados. Selecione os seguintes campos em ordem:

Remunerações
SSWages
MedicareWages

5 Arraste os campos para a faixa de detalhe, posicionando o mouse sobre a área do formulário rotulada '1 Salários, gorjetas e outras remunerações'. Solte o botão do mouse.

6 Feche a árvore de dados.

7 Use o Inspetor de objetos para definir a largura para 1.5208. A largura dos componentes selecionados aumentará.

8 Clique com o botão direito do mouse no componente Wages DBText e selecione a opção de menu DisplayFormat.

9 Defina o formato de exibição como \$ #, 0,00; (\$ #, 0,00), (a primeira seleção com um cifrão) selecione o texto na caixa Editar e copie-o para a área de transferência. Clique em Cancelar.

10 Clique na propriedade DisplayFormat no Object Inspector. O DisplayFormat selecionado deve aparecer. Cole o DisplayFormat neste campo e clique fora da propriedade.

11 Clique na imagem e, em seguida, clique em cada um dos componentes DBText em sucessão, verificando se o DisplayFormat foi definido.

12 Defina a posição do Wages DBText compo-

nent:

Deixou	4.375
Principal	0,8021

13 Defina a posição do SSWages DBText componente:

Deixou	4,375
Principal	1,1146

14 Defina a posição do MedicareWages DBText componente:

Deixou	4,375
Principal	1,4271

verificação de layout

Criar os componentes DBText retidos

1 Selecione os componentes Wages, SSWages e MedicareWages DBText.

2 Copie e cole a seleção.

3 Arraste a seleção para a direita, posicionando-a sobre a área '2 Imposto de renda federal retido' da imagem.

4 Defina a largura dos componentes como 1,6042 no Inspetor de objetos.

5 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar os componentes:

Deixou	6,0521
Principal	0,8021

6 Defina AutoSize como True no Object Inspector. Todos os três componentes DBText devem ser dimensionados automaticamente.

7 Clique na imagem para desmarcar os componentes.

8 Selecione o componente Wages DBText recém-criado e defina o campo como FederalTaxWithheld.

9 Selecione o componente SSWages DBText e defina o campo como SSTaxWithheld.

10 Selecione o componente MedicareWages DBText e defina o campo como MedicareTaxWithheld.

verificação de layout

Crie as informações de endereço

1 Role para a parte superior e esquerda para que todo o canto esquerdo da imagem fique visível.

2 Coloque uma etiqueta na caixa 'b Número de identificação do funcionário'.

3 Configure o rótulo:

Rubrica	78-5676-809898
Deixou	0,5729
Principal	0,8021

4 Coloque uma caixa de endereço de componente de no empregador memorando do formulário (caixa c).

5 Defina a posição do memorando:

Deixou	0,5313
Principal	1,125
Largura	3,5521
Altura	0,7708

6 Clique com o botão direito sobre o memorando e acesse a opção de menu Linhas Insira o seguinte texto no Editor de Memorando:

Digital Metaphors Corporation
16775 Addison Rd., Suite 613
Addison, TX 75001

7 Selecione o componente memo.

8 Copie e cole a seleção.

9 Arraste o novo componente de memorando para a caixa de endereço do funcionário (caixa f).

10 Defina a posição do memorando:

Deixou	0,5
Principal	2,7813

11 Defina o nome do memorando como mmEmployeeAddress1.

12 Coloque um componente DBText na caixa do número da previdência social do funcionário (caixa d).

13 Configure o componente DBText:

Tamanho automático	Verdadeiro
Campo de dados	CustNo
Formato de apresentação	000-0000
Deixou	0,5
Principal	2.0416

14 Coloque um componente DBText na caixa Nome do funcionário (caixa e).

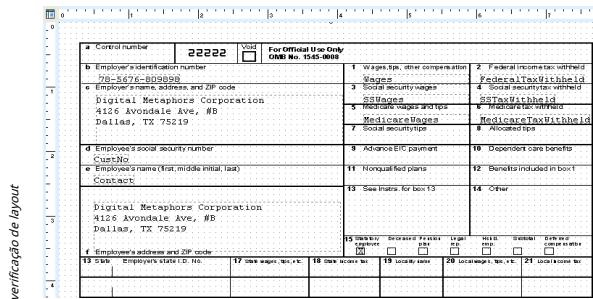
15 Configure o componente DBText:

Tamanho automático	Verdadeiro
Campo de dados	Contato
Deixou	0,5
Principal	2,3646

16 Coloque uma etiqueta na caixa '15' do formulário. Defina a legenda como X.

17 Posicione o rótulo:

Deixou	4,2917
Principal	3.4583



Criar informação de componente duplicada

1 Selecione todos os componentes na imagem do formulário.

2 Copie e cole a seleção.

3 Role para baixo até o formulário duplicado.

4 Posicione a seleção dentro das caixas do formulário duplicado.

5 Mantenha pressionadas as setas do teclado e posicione a seleção:

Deixou	0,5
Principal	5,9896

6 Selecione o memorando na caixa Endereço do funcionário do formulário duplicado (caixa f).

7 Defina o nome do memorando como mmEmployeeAddress2.

8 Selecione Arquivo | Salve no menu principal do Delphi.

Escreva a rotina de 'aperto de endereço'

1 Feche o Report Designer.

2 Localize a lista suspensa de componentes na parte superior do Object Inspector.

3 Selecione a banda de detalhes nesta lista (deve ser nomeada ppDetailBand1).

4 Selecione a guia Eventos e clique duas vezes no evento BeforeGenerate.

5 Coloque o seguinte código no manipulador de eventos:

```
BuildEmployeeAddress (mmEmployeeAddress1.Lines);
BuildEmployeeAddress (mmEmployeeAddress2.Lines);
```

6 Role para cima até a declaração da classe do formulário.

Substitua o comentário das declarações privadas pela seguinte declaração de procedimento:

procedimento BuildEmployeeAddress (aStrings: TStrings);

7 Role para baixo abaixo do manipulador de eventos DetailBand BeforeGenerate e insira o código mostrado abaixo como o procedimento BuildEmployeeAddress.

Observação: neste manipulador de eventos, precisamos construir o mesmo endereço para dois componentes de memorando diferentes. Para fazer isso, podemos criar um

Código	Manipulador de eventos para o procedimento BuildEmployeeAddress
	<pre> procedimento TfrmFormsEmulation.BuildEmployeeAddress (aStrings: TStrings); var lsLine: String; lsState: String; lsZIP: String; começar {limpar memo} aStrings.Clear; {adicionar contato} lsLine := tblCustomer.FieldByName ('Contact'). AsString; aStrings.Add (lsLine); {adicionar linha de endereço 1} lsLine := tblCustomer.FieldByName ('Addr1'). AsString; E se (lsLine <> '') então aStrings.Add (lsLine); {adicionar linha de endereço 2} lsLine := tblCustomer.FieldByName ('Addr2'). AsString; E se (lsLine <> '') então aStrings.Add (lsLine); {adicionar cidade, CEP do estado} lsLine := tblCustomer.FieldByName ('City'). AsString; lsState := tblCustomer.FieldByName ('State'). AsString; E se (lsState <> '') então lsLine := lsLine + ',' + lsState; lsZIP := tblCustomer.FieldByName ('ZIP'). AsString; E se lsZIP <> '' então lsLine := lsLine + " " + lsZIP; aStrings.Add (lsLine); fim; </pre>

rotina geral (BuildEmployeeAddress) que podemos chamar para cada componente, ou podemos construir o endereço em uma variável de string local e então atribuí-lo a ambos os componentes. Escolhemos a última abordagem para facilitar a leitura.

Nota: Esta rotina simplesmente recupera cada elemento do endereço do Employee, concatenando e armazenando o resultado no objeto TStringList passado no parâmetro. As instruções 'if' verificam se há strings vazias, garantindo que nenhuma linha em branco apareça no endereço.

Visualize o relatório em tempo de execução

1 Selecione a guia Padrão da paleta de componentes Delphi.

2 Adicione um componente Botão ao formulário.

3 Configure o componente Botão:

Nome btnPreview

Antevisão da legenda

4 Adicione o seguinte código ao manipulador de eventos OnClick do botão:

```
rbFormEmu.Print;
```

5 Selecione o projeto | Compile rbFEProj. Corrija quaisquer problemas de compilação.

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Execute o projeto.

8 Clique no botão Visualizar. O relatório deve ser exibido no formulário Visualizar impressão. O relatório deve ter a seguinte aparência:

The screenshot shows a 'Print Preview' window with a 'Wage and Tax Statement' form. The form is titled 'Wage and Tax Statement' and includes the following information:

- Form Information:** Cat. No. 101340, Department of the Treasury-Internal Revenue Service, For Paperwork Reduction Act Notice, see separate instructions.
- Form Fields:**
 - 1. Form number:** 22222
 - 2. Employee identification number:** 78-5676-809898
 - 3. Employee name, address, and ZIP code:** Digital Metaphors Corporation, 4126 Avondale Ave, #B, Dallas, TX 75219
 - 4. Federal income tax withheld:** \$8,000.00
 - 5. Social security wages:** \$40,000.00
 - 6. Social security tax withheld:** \$2,000.00
 - 7. Medicare wages and tips:** \$40,000.00
 - 8. Medicare tax withheld:** \$400.00
 - 9. Employee's social security number:** 000-00-1234
 - 10. Advance EIC payment:** 0
 - 11. Nonqualified plans:** 0
 - 12. Benefits included in box 1:** 0
 - 13. See instructions for box 13:** 0
 - 14. Other:** 0
 - 15. Date:** 12/31/2010
 - 16. Use wages, box 1:** 0
 - 17. Use income tax:** 0
 - 18. Use income tax:** 0
 - 19. Use income tax:** 0
 - 20. Use income tax:** 0
 - 21. Use income tax:** 0
- Employee Information:**
 - Employee name (last, middle initial, first):** Erica Norman
 - Employee address and ZIP code:** Erica Norman, 4-976 Sugarloaf Hwy, Suite 103, Kapaa Kauai, HI 96766-1234

Mestre → Relatório de Detalhe

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configurar objetos de acesso a dados Delphi para uso em um relatório mestre / detalhado
- Configure objetos de acesso a dados Delphi para uso como uma tabela de pesquisa
- Use um sub-relatório para imprimir dados detalhados
- Definir uma pausa em grupo

Order Summary *Marine Adventures & Sunken Treasure Co.*

Company: **Kawaii Dive Shoppe**
Customer: **1221**

Number	Date	Order	Sales Rep.	Via	Ship	Date	Method	Amount
1023	07/01/1988	Kim		UPS	07/01/1988	Check		\$4,674.00
1076	12/16/1994	Phil		UPS	04/26/1995	Yess		\$17,721.00
1123	05/24/1993	Rabrous		UPS	05/24/1993	Check		\$13,945.00
1169	07/06/1994	Tau		UPS	07/06/1994	Credit		\$9,471.95
1176	07/26/1994	Cool		UPS	07/26/1994	Yess		\$4,172.25
1289	12/16/1994	Ass		UPS	12/16/1994	Credit		\$1,400.00
Kawaii Dive Shoppe Total								\$51,480.00

2:40:03.333.02 PM Page: 1 of 1

Criar uma nova aplicação

- 1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.
- 2 Defina o nome do formulário como 'frmMasterDetail'.
- 3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbMD no diretório My RB Tutoriais (estabelecido na página 181).

Nota: É importante nomear o formulário e salvar a unidade do formulário usando os nomes fornecidos acima porque o relatório será usado em tutoriais posteriores.

- 4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

- 5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

- 6 Salve o projeto com o nome rbMDProj no diretório My RB Tutoriais.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie a tabela, a fonte de dados e a linha de dados para a tabela mestre

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblCustomer
Nome da tabela	customer.db

4 Adicione um componente DataSource ao formulário.

5 Configure o componente DataSource:

DataSet	tblCustomer
Nome	dsCustomer

6 Selecione a aba RBuilder do Delphi compo-
paleta atual.

7 Adicionar um DBPipeline  componente do formulário.

8 Configure o componente DBPipeline:

DataSource	dsCustomer
Nome	plCustomer

Crie a tabela, a fonte de dados e a linha de dados para a tabela de detalhes

1 Adicione um segundo conjunto de componentes Table, DataSource e DataPipeline ao seu formulário.

2 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblOrder
Nome da tabela	pedidos.db

3 Configure o componente DataSource:

DataSet	tblOrder
Nome	dsOrder

4 Configure o componente DBPipeline:

Fonte de dados	dsOrder
Nome	plOrder

Crie a tabela, a fonte de dados e a linha de dados para a tabela de pesquisa

1 Adicione um terceiro conjunto de componentes Table, DataSource e DataPipeline ao seu formulário.

2 Configure o componente Tabela:

DatabaseName	DBDEMOS
Nome	tblEmployee
TableName	employee.db

3 Configure o componente DataSource:

DataSet	tblEmployee
Nome	dsEmployee

4 Configure o componente DBPipeline:

DataSource	dsEmployee
Nome	plEmployee

Defina a relação entre o cliente e a tabela de pedidos

1 Selecione o componente Tabela de pedidos.

2 Defina a propriedade MasterSource como dsCustomer.

3 Selecione a propriedade MasterFields e clique no botão "...". Isso exibe o Field Link Designer.

4 Clique na lista suspensa Available Indexes e selecione CustNo.

5 Selecione o CustNo na lista Campos de detalhes.

6 Selecione o item CustNo na lista Master Fields.

7 Clique no botão Adicionar. Um novo item (CustNo → CustNo) será adicionado à lista Campos Associados.

8 Clique no botão OK para fechar a caixa de diálogo.

Observação: a propriedade IndexName agora está definida como CustNo. Essa propriedade foi definida automaticamente quando selecionamos o índice CustNo na etapa 4 acima.

Nota: Para estabelecer um relacionamento de detalhes mestres, a tabela de detalhes deve ser indexada no valor ou valores do campo de ligação. Você pode pensar em um Índice como definindo a ordem de classificação da tabela. Portanto, se precisarmos vincular a tabela Order à tabela Customer com base em CustNo, a tabela Order deve ter um índice (ou seja, ordem de classificação) com base em CustNo.

Defina a relação entre o pedido e a tabela de funcionários

1 Selecione o componente da tabela Employee.

2 Defina a propriedade MasterSource como dsOrder.

3 Inicie o Field Link Designer a partir da propriedade MasterField.

4 Selecione o campo do item EmpNo na lista Campos de detalhes.

5 Selecione o campo EmpNo na lista Campos mestre.

6 Clique no botão Adicionar. Um novo item (EmpNo → EmpNo) será adicionado à lista de Campos Associados.

7 Clique no botão OK para fechar a caixa de diálogo.

8 Selecione Arquivo | Salve no menu principal do Delphi.

Observação: neste exemplo, você notará que os campos vinculados têm o mesmo nome. Este é um bom padrão a ser seguido ao projetar um banco de dados. No entanto, não é um requisito para vincular campos.

Crie um relatório e conecte-o aos dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

DataPipeline plCustomer
Nome rbOrderSummary

3 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

4 Posicione o Report Designer de modo que o objeto O inspetor está visível.

5 Coloque um sub-relatório  componente (do Barra de ferramentas avançada) na faixa de detalhes.

Nota: Ao adicionar o sub-relatório, duas guias serão exibidas na parte inferior da área de trabalho: Principal e SubRelatório1. Há um espaço de trabalho separado para o relatório principal e para cada sub-relatório. Você pode acessar a área de trabalho de qualquer relatório clicando na guia.

6 Use o Object Inspector para definir a propriedade `DataPipeline` do sub-relatório como `plOrder`.

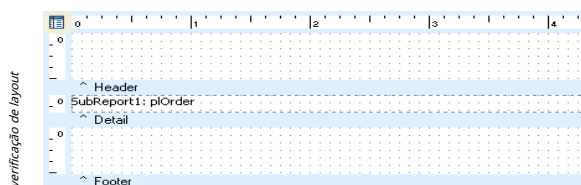
Nota: Quando inicialmente definimos a propriedade `DataPipeline` do relatório principal para a tabela do cliente, estávamos instruindo o relatório a imprimir uma banda de detalhes para cada cliente. Agora que definimos a propriedade `DataPipeline` do sub-relatório, estamos dizendo ao sub-relatório para percorrer todos os pedidos de cada cliente, imprimindo uma banda de detalhes para cada pedido.

7 Selecione Relatório | Dados do menu principal. `plCustomer` deve ser selecionado como o pipeline de dados.

Nota: A caixa de diálogo Dados pode ser usada como uma maneira conveniente de definir a propriedade `DataPipeline` de um relatório.

8 Posicione o sub-relatório no topo da faixa de detalhes.

9 Arraste o divisor da banda de detalhes para cima até que esteja alinhado com a parte inferior do sub-relatório.



Crie as etiquetas da faixa do cabeçalho

1 Defina a altura da banda do cabeçalho em 2,5 polegadas.

2 Coloque um rótulo no canto superior esquerdo do banda de cabeçalho.

3 Configure o rótulo:

Rubrica	resumo do pedido
Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito itálico

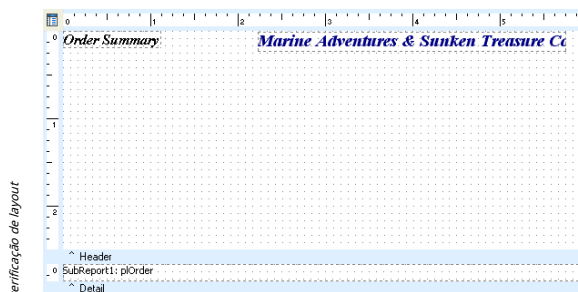
4 Coloque outra etiqueta no centro superior do banda de cabeçalho.

5 Configure o rótulo:

Rubrica	Aventuras marinhas e Sunken Treasure Co.
Nome da fonte	Times New Roman
Tamanho da fonte	16
Estilo de fonte	Negrito itálico
Cor da fonte	Marinha
Alinhamento de Texto	Centrado

6 Centralize o rótulo clicando em Center Horizontal no ícone da banda na ferramenta Alinhar ou Espaço-bar.

7 Selecione os dois rótulos e arraste-os para o topo da banda do cabeçalho.



Crie a forma da faixa do cabeçalho

1 Coloque um componente de forma no lado esquerdo da faixa do cabeçalho.

2 Selecione 'Retângulo arredondado' na lista suspensa no canto superior esquerdo do Report Designer.

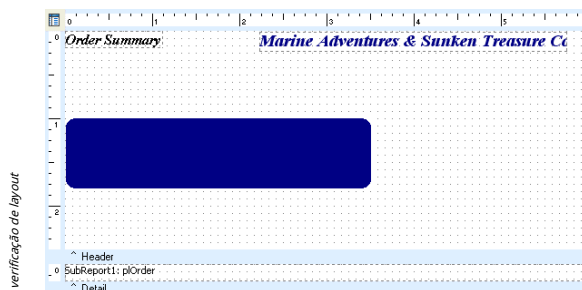
3 Use a barra de ferramentas Desenhar para definir as cores de preenchimento e linha da forma como Marinha.

4 Arraste a forma para a borda esquerda da banda e posicione a parte superior na marca de 1 polegada na régua vertical.

5 Defina o tamanho da forma:

Largura	3,5
Altura	0,8

6 Selecione Arquivo | Salve no menu principal do Delphi.



Use a árvore de dados para completar o layout da faixa do cabeçalho

1 Selecione Exibir | Barras de ferramentas | Árvore de dados no menu Report Designer. Posicione a Árvore de dados à esquerda do Report Designer, na frente do Object Inspector.

2 Clique na guia Layout na parte inferior da Árvore de dados e defina as configurações de arrastar e soltar:

Crio	Tudo
Estilo	Vertical
Nome da fonte do rótulo	Novos tempos
Estilo de fonte da etiqueta	itálico
Tamanho da fonte da etiqueta	12 pontos
Cor da fonte do rótulo	Branco
Nome da fonte do campo	Novos tempos
Estilo da fonte do campo	Negrito
Tamanho da fonte do campo	12 pontos
Cor da fonte do campo	Branco

3 Clique na guia Dados da Árvore de Dados. Certifique-se de que o pipeline de dados do cliente esteja selecionado.

4 Clique no campo Empresa e, em seguida, clique com a tecla Ctrl pressionada no campo CustNo.

5 Arraste os campos selecionados para a área de trabalho, posicionando o mouse apenas dentro do canto superior esquerdo da forma. Solte o botão do mouse. Dois componentes DBText e rótulos correspondentes serão criados.

6 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

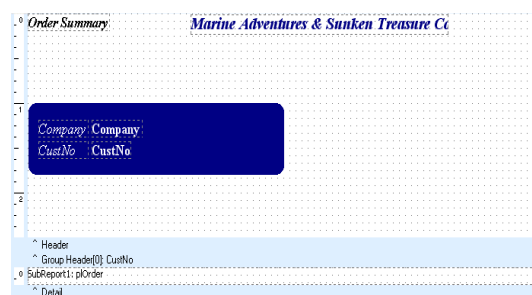
Deixou	0,1458
Principal	1,1875

7 Clique no espaço em branco da banda do cabeçalho para desmarcar os componentes.

8 Selecione o rótulo CustNo e clique com a tecla Shift pressionada no componente DBText correspondente. Clique no ícone Justificar à esquerda na barra de ferramentas Formatar.

9 Clique com o botão direito do mouse no componente Company DBText e defina a propriedade AutoSize como True.

10 Clique com o botão direito do mouse no componente CustNo DBText e defina a propriedade AutoSize como True.



Criar um grupo no campo CustNo

1 Selecione Relatório | Grupos no menu principal do Report Designer. A caixa de diálogo Grupos será exibida.

2 Selecione o campo 'plCustomer.CustNo' na lista suspensa na parte superior da caixa de diálogo.

3 Clique no botão Adicionar. Um novo grupo será criado.

4 Clique nas caixas de seleção Iniciar nova página e Redefinir número da página perto do meio da caixa de diálogo. Isso fará com que a lista de cada cliente comece em uma nova página e apareça com seus próprios números de página.

Nota: Como os grupos são baseados em uma mudança no valor do campo, o campo no qual um grupo se baseia deve sempre ser usado para ordenar os registros. ReportBuilder não classifica registros; portanto, você deve ordenar os registros usando um Índice ou especificando uma cláusula Order By em uma consulta SQL.

5 Clique no botão OK. Você deve ver duas novas bandas: uma acima e outra abaixo da banda de detalhes. Essas faixas não têm altura; portanto, nenhum espaço em branco aparece acima das divisórias.

6 Selecione Arquivo | Salve no menu principal do Delphi.

Comece o Layout de Informações do Pedido

1 Clique na guia 'SubReport1' na parte inferior do designer.

Nota: O objeto selecionado no Object Inspector mudará para ppChildReport1. O componente de sub-relatório é um contêiner para o relatório filho. O sub-relatório controla as propriedades que direcionam o comportamento de impressão do relatório. O sub-relatório também controla o relacionamento do relatório com outros componentes no layout do relatório principal. O relatório filho contém o layout real do relatório (bandas, rótulos, etc.) e pode ser acessado por meio da propriedade Relatório do componente de sub-relatório (ou seja, SubReport1.Report).

2 Coloque um componente de forma no lado esquerdo da faixa do título.

3 Use a barra de ferramentas Desenhar para definir as cores de preenchimento e linha da forma como Marinha.

4 Clique com o botão direito sobre a forma para exibir o menu pop-up. Selecione as opções ParentHeight e ParentWidth. A forma será redimensionada e cobrirá toda a banda.

5 Clique na guia Layout na parte inferior da Árvore de dados e defina as configurações de arrastar e soltar:

Crio	Tudo
Estilo	Tabular
Nome da fonte do rótulo	Novos tempos
Estilo de fonte da etiqueta	Negrito
Tamanho da fonte da etiqueta	12 pontos
Cor da fonte do rótulo	Branco
Nome da fonte do campo	Novos tempos
Estilo da fonte do campo	Regular
Tamanho da fonte do campo	10 pontos

6 Clique na guia Dados na parte inferior da Árvore de dados e selecione o pipeline de dados plOrder.

7 Use o método Ctrl-clique para selecionar estes campos:

OrderNo

Data de venda

8 Arraste os campos selecionados para a faixa do título, posicionando o mouse no canto inferior esquerdo da forma. Solte o botão do mouse para criar os componentes.

9 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

Deixou 0,1458

Principal 0,2813

10 Defina as legendas do rótulo para Número e Data.

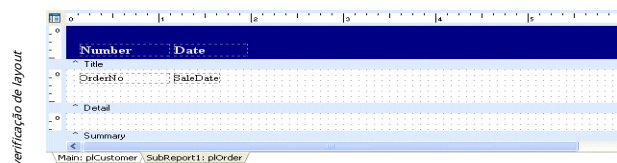
11 Justifique à esquerda todos os componentes.

12 Clique com o botão direito do mouse no componente SaleDate DBText e defina DisplayFormat como 'mm / dd / aaaa'.

Nota: A caixa de diálogo Formato de exibição verifica automaticamente o tipo de dados do componente e exibe uma lista dos formatos apropriados.

13 Defina a largura do rótulo de data e o componente DBText correspondente como 0,8.

14 Selecione os componentes DBText e mova-os para uma posição superior de 0,0521.



Preencher o Layout de Informações do Pedido

1 Selecione o pEmployee DataPipeline na árvore de dados.

2 Arraste o campo FirstName para a faixa do título, posicionando o mouse imediatamente à direita do rótulo Date.

3 Posicione a seleção:

Esquerda 2.2083

Top 0,2813

4 Alinhe a parte superior do novo DBText com a parte superior do componente Date DBText.

5 Defina o rótulo Legenda para Representante de vendas.

6 Coloque um componente de forma no lado esquerdo da faixa do título.

7 Alinhe a borda esquerda da forma com a etiqueta Número e estique a largura até que a borda direita fique alinhada com a borda direita da etiqueta Representante de vendas.

8 Configure a forma:

Principal	0,1
Altura	0,08
Cor de Preenchimento	Branco
Cor da linha	Branco

9 Coloque um rótulo diretamente no topo da forma, posicionando-o próximo ao ponto médio da forma.

10 Configure o rótulo:

Tamanho automático	Falso
Rubrica	Pedido
Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito
Cor da fonte	Branco
Cor de destaque	Marinha
Alinhamento de Texto	Centrado
Largura	0,55

11 Selecione a forma branca e, em seguida, clique com a tecla Shift

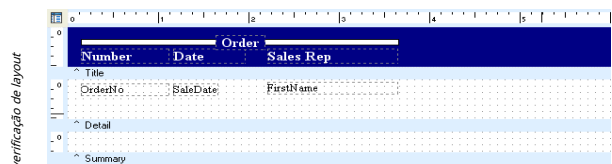
pressionada no rótulo do pedido. Clique em Alinhar ao Meio  e

Ícones de alinhamento central da ferramenta Alinhar ou Espaço - bar.

12 Selecione Arquivo | Salve no menu principal do Delphi.

13 Clique na guia Visualizar para ver o relatório. Clique nos botões da próxima página para visualizar várias páginas.

Quando terminar de visualizar, volte para a área de trabalho Design.



Esquematize os componentes de informações de envio

1 Selecione o plOrder DataPipeline na árvore de dados e selecione os seguintes campos em ordem:

ShipVia

Data de envio

2 Arraste a seleção para a faixa do título, imediatamente à direita do rótulo Sales Rep. Posição a seleção:

Deixou	3.875
Principal	0,2813

3 Alinhe a parte superior dos novos componentes DBText com a parte superior dos componentes existentes.

4 Defina as legendas do rótulo como Via e Data.

5 Defina a largura do rótulo de data e o componente DBText correspondente como 0,8.

6 Defina o formato de exibição do componente Date DBText como 'mm \ dd \ aaaa'.

7 Selecione o rótulo do pedido e clique com a tecla Shift pressionada na forma branca atrás dele.

8 Copie e cole a seleção.

9 Posicione os novos componentes acima dos rótulos Via e Data.

10 Defina a posição e o tamanho da forma:

Deixou	3,8333
Largura	1,61

11 Alinhe a parte superior da nova forma com a parte superior da forma existente.

12 Clique com o botão direito sobre o rótulo e selecione a opção de menu Trazer para a frente.

13 Defina a legenda do rótulo como Enviar.

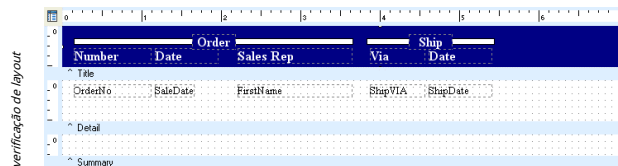
14 Selecione o componente de forma e, em seguida, clique com a tecla

Shift pressionada no rótulo Navio. Clique em Alinhar ao Meio e

Alinhar Centro  ícones da ferramenta Alinhar ou Espaço - bar.

15 Selecione Arquivo | Salve no menu principal do Delphi.

Visualize conforme desejado.



Esquematize os componentes de informações de pagamento

1 Selecione o plOrder DataPipeline na árvore de dados e selecione os seguintes campos em ordem:

Forma de pagamento

Quantia paga

2 Arraste a seleção para a faixa do título, imediatamente à direita do rótulo ShipDate. Posicione a seleção:

Deixou	5,5833
Principal	0,2813

3 Alinhe a parte superior dos novos componentes DBText com a parte superior dos componentes existentes.

4 Defina as legendas do rótulo para Método e Quantidade.

5 Defina DisplayFormat do componente Amount DBText como \$ #, 0,00; (\$ #, 0,00) (o primeiro formato com um cifrão).

6 Justifique à direita o texto no rótulo Quantidade.

7 Selecione o rótulo de navio e clique com a tecla Shift pressionada na forma branca atrás dele.

8 Copie e cole a seleção.

9 Arraste os novos componentes para a direita, alinhando a borda esquerda da forma com a borda esquerda do rótulo Método.

10 Alinhe a parte superior da nova forma com a parte superior das formas existentes.

11 Defina a largura da nova forma para 2,27.

12 Clique com o botão direito sobre o rótulo e selecione Trazer para Frente.

13 Configure o rótulo:

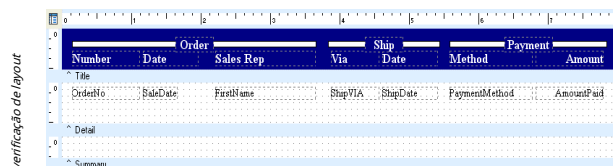
Rubrica	Pagamento
Largura	0,7

14 Selecione a forma e clique no rótulo com a tecla Shift pressionada. Clique em Align Middle and Align Center



Ícones da barra de ferramentas Alinhar ou Espaço.

15 Selecione Arquivo | Salve no menu principal do Delphi. Visualize conforme desejado.



Complete a faixa de detalhes

1 Coloque um componente Linha no canto superior esquerdo da banda de detalhes.

2 Localize a lista suspensa Posição da linha no canto superior esquerdo do Report Designer.

3 Selecione Esquerda nesta lista. Uma linha vertical aparecerá no lado esquerdo do componente de linha.

4 Configure a linha:

Cor	Marinha
Espessura	1 1/2 ponto
Deixou	0
Largura	0,10
ParentHeight	Verdadeiro

5 Copie e cole o componente da linha.

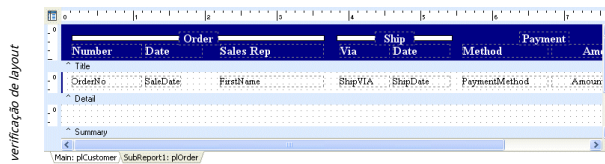
6 Selecione Direita na lista suspensa Posição da linha. A linha vertical se moverá para o lado direito do componente de linha.

7 Arraste a nova linha para o lado direito da banda de detalhes. Mantenha pressionada a tecla Ctrl e use o botão de seta para a direita para posicionar a linha de forma que fique alinhada com a borda direita da banda. Visualize para ter certeza de que a forma e a linha na faixa do título estão alinhadas.

8 Defina a altura da banda de detalhe para 0,3.

9 Defina a altura da faixa de resumo para 0,3125.

10 Selecione Arquivo | Salve no menu principal do Delphi. Visualize conforme desejado.



Esquematize a Faixa de Resumo

1 Coloque um componente DBCalc na faixa de resumo.

2 Configure o componente DBCalc:

Campo de dados	Quantia paga
Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Negrito
Cor da fonte	Preto
Principal	0,083
Tamanho automático	Verdadeiro
Formato de apresentação	\$ #, 0,00; (\$ #, 0,00)
Alinhamento de Texto	Justificado corretamente

3 Selecione o componente AmountPaid DBText na banda de detalhes e, em seguida, clique com a tecla Shift pressionada no componente DBCalc. Clique no ícone Aumentar a largura para o maior



da barra de ferramentas Tamanho. Clique em Alinhar à direita

ícone  da barra de ferramentas Alinhar ou Espaço.

4 Coloque uma linha diretamente abaixo do componente DBCalc nent.

5 Configure a linha:

Posição	Inferior
Estilo	Dobro
Principal	0,25
Largura	1,18
Altura	0,05

6 Mantenha pressionada a tecla Ctrl e use a tecla de seta para a direita para alinhar a borda direita da linha com a borda direita do DBCalc.

7 Coloque um rótulo à esquerda do componente DBCalc. Configure o rótulo:

Rubrica	Total
Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Negrito
Deixou	6,2292

8 Alinhe a parte superior da etiqueta com a parte superior do DBCalc.

9 Coloque um componente DBText à esquerda do rótulo. Configure o DBText:

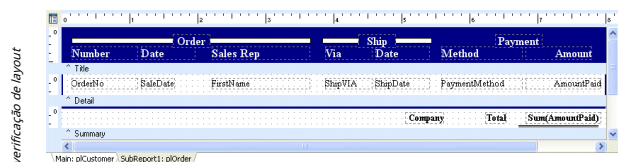
DataPipeline	plCustomer
Campo de dados	Empresa
Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Negrito
Deixou	5
Tamanho automático	Verdadeiro
Alinhamento de Texto	Justificado corretamente

10 Alinhe a parte superior do componente DBText com a parte superior da etiqueta.

11 Coloque um componente de linha no canto superior esquerdo da faixa de resumo. Configure a linha:

ParentWidth	True
Cor da Linha	Marinho
Espessura da linha	2 1/4 ponto
Top	0
Altura	0,05

12 Selecione Arquivo | Salve no menu principal do Delphi. Visualize conforme desejado.



Esquematize a faixa de rodapé

1 Clique na guia 'Principal' para retornar ao relatório principal.

2 Coloque um componente SystemVariable no canto esquerdo inferior da faixa de rodapé.

3 Selecione PrintDateTime na lista suspensa no canto superior esquerdo do Report Designer.

4 Defina a posição esquerda para 0,0625.

5 Defina a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	10
Cor da fonte	Preto
Estilo de fonte	Regular
Alinhamento de Texto	Justificado à esquerda

6 Coloque outro componente da Variável de sistema no canto inferior direito da faixa de rodapé.

7 Selecione PageSetDesc na lista suspensa no canto superior esquerdo do Report Designer.

8 Defina a posição esquerda para 7,3229.

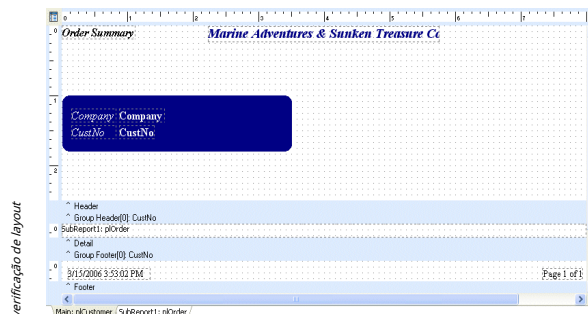
9 Defina a fonte:

Nome da fonte	Times New Roman
Tamanho da fonte	10
Cor da fonte	Preto
Estilo de fonte	Regular
Alinhamento de Texto	Justificado corretamente

10 Alinhe as partes superiores dos componentes da variável do sistema.

11 Arraste o divisor de faixa do rodapé para cima até que ele encontre a parte inferior dos componentes da variável do sistema.

12 Selecione Arquivo | Salve no menu principal do Delphi. Visualize o relatório concluído.



Visualize o relatório em tempo de execução

1 Feche o Report Designer.

2 Selecione a guia Padrão da paleta de componentes Delphi.

3 Adicione um componente Botão ao formulário.

4 Configure o componente Botão:

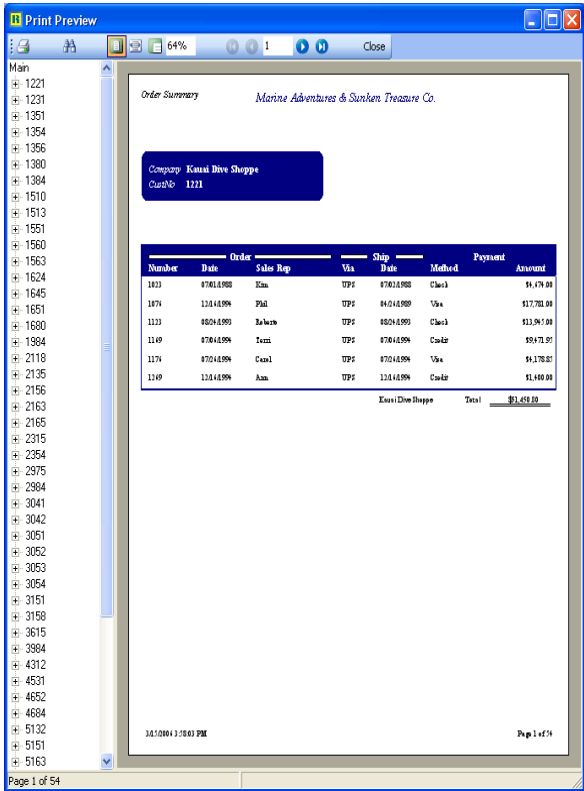
Nome	btnPreview
Rubrica	Antevisão

5 Adicione o seguinte código ao manipulador de eventos OnClick do botão:

```
rbOrderSummary.Print;
```

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Execute o projeto. Clique no botão Visualizar. O relatório deve ter a seguinte aparência:



Mestre → Detalhe → Relatório de Detalhe

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configurar objetos de acesso a dados Delphi para uso em um relatório mestre / detalhe / detalhe
- Configure objetos de acesso a dados Delphi para uso como uma tabela de pesquisa
- Use um sub-relatório para imprimir dados detalhados
- Faça cálculos
- Definir uma pausa em grupo

Criar uma nova aplicação

Observação: este tutorial se baseia no relatório Master-Detail criado na seção anterior. Você pode concluir este tutorial ou copiar o formulário rbMD do diretório RBuilder \ Tutorials.

1 Seleção Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.

2 Feche o novo formulário e unidade sem salvar.

3 Selecione Arquivo | Abra no menu principal do Delphi.
Localize a unidade rbMD.pas e abra-a. Altere o nome do formulário para 'frmMasterDetailDetail'.

4 Seleccione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbMDD no diretório My RB Tutoriais (estabelecido na página 181).

Nota: É importante nomear o formulário e salvá-lo usando os nomes fornecidos acima porque o relatório será usado em tutoriais posteriores.

5 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

6 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione Adicionar. Adicione o formulário rbMDD ao projeto.

7 Clique com o botão direito sobre o nome do projeto mais uma vez e selecione a opção de menu Salvar.

8 Salve o projeto com o nome `rbMDDProj` no diretório `My RB Tutorials`.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

<i>Order Detail</i>		<i>Marine Adventures & Sunken Treasure Co.</i>				
<i>Company:</i> Kauai Dive Shoppe <i>Customer:</i> 1221						
Number	Date	Order Sales Rep	Via	Ship Date	Payment Method	Amount
1023	07/01/98B	Em	UPF	07/01/98B	Cash	\$4,478.00
Bom No.	Description	Quantity	List Price	Bom Total		
1	Ra gulator System	3	\$150.00	\$0.00		
2	Depth/Sensor Gauge	4	\$20.00	\$0.00		
3	Airmaxx Inflator/Ra gulator	10	\$140.00	\$0.00		
Total				\$0.00		
1074	12/24/99A	Phil	UPF	04/24/98B	V/a	\$17,788.00
Bom No.	Description	Quantity	List Price	Bom Total		
1	Samuel's Opened Value \$y	4	\$2,349.00	\$0.00		
2	Towable Value Camera (E&S)	4	\$1,999.00	\$0.00		
3	Second Shm Ra gulator	1	\$309.00	\$0.00		
Total				\$0.00		
1123	08/24/99B	Kelene	UPF	08/24/99B	Cash	\$13,943.00
Bom No.	Description	Quantity	List Price	Bom Total		
1	Second Shm Ra gulator	14	\$343.00	\$0.00		
2	Second Shm Ra gulator	24	\$153.00	\$0.00		
3	Depth/Sensor Gauge Comp	2	\$188.00	\$0.00		
4	Wet Band Thermometer (P)	2	\$18.00	\$0.00		
5	Chisel Point Knife	2	\$41.00	\$0.00		
6	Flashlight	3	\$13.00	\$0.00		
Total				\$0.00		
1149	07/04/99A	Ismi	UPF	07/04/99A	Credit	\$9,471.95

12/06/99

Page 1

Atualize o nome do relatório

- 1 Selecione o componente do relatório.
- 2 Defina o Nome como `rbOrderDetail`.
- 3 Clique duas vezes no botão Visualizar em seu formulário Delphi.
- 4 No manipulador de eventos `OnClick`, altere o nome do relatório para:
`rbOrderDetail.Print;`

Adicione a Tabela, DataSource e DataPipeline para a Tabela de Detalhes de Itens

- 1 Adicione um componente Tabela ao formulário.
- 2 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblItem
Nome da tabela	items.db
- 3 Adicione um componente DataSource ao formulário.
- 4 Configure o componente DataSource:

DataSet	tblItem
Nome	dsItem
- 5 Selecione a aba RBuilder do Delphi compo-
paleta atual.
- 6 Adicionar um DBPipeline  componente do formulário.
- 7 Configure o componente DBPipeline:

DataSource	dsItem
Nome	plItem

Defina a relação entre o pedido e a tabela de itens

- 1 Selecione o componente da tabela de itens.
- 2 Defina a propriedade `MasterSource` como `dsOrder`.
- 3 Selecione a propriedade `MasterFields` e pressione o botão "...". Isso exibe o Field Link Designer.
- 4 Clique na lista suspensa `Available Indexes` e selecione `ByOrderNo`.
- 5 Selecione o item `OrderNo` na lista `Detail Fields`.
- 6 Selecione o item `OrderNo` na lista `Master Fields`.
- 7 Clique no botão Adicionar. Um novo item (`OrderNo` → `OrderNo`) será adicionado à lista Campos unidos.
- 8 Pressione o botão OK para fechar a caixa de diálogo.

Observação: a propriedade `IndexName` agora está definida como `ByOrderNo`. Esta propriedade foi definida automaticamente quando selecionamos o índice `ByOrderNo` na etapa 4 acima. Para estabelecer um relacionamento de detalhes mestres, a tabela de detalhes deve ser indexada no valor ou valores do campo de ligação. Você pode pensar em um Índice como definindo a ordem de classificação da tabela. Portanto, se precisarmos vincular a tabela Itens à tabela Pedido com base no `PedidoNo`, a tabela Itens deve ter um índice (ou seja, ordem de classificação) com base no `PedidoNo`.

Crie a tabela, a fonte de dados e a linha de dados para a tabela de pesquisa de peças

1 Adicione um componente Tabela ao seu formulário.

2 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblPart
Nome da tabela	parts.db

3 Adicione um componente DataSource ao formulário.

4 Configure o componente DataSource:

DataSet	tblPart
Nome	dsPart

5 Selecione a aba RBuilder do Delphi compo-
paleta atual.

6 Adicionar um DBPipeline  componente do formulário.

7 Configure o componente DBPipeline:

Fonte de dados	dsPart
Nome	plPart

Defina a relação entre o item e a tabela de peças

1 Selecione o componente Tabela de peças.

2 Defina a propriedade MasterSource como dsItem.

3 Inicie o Field Link Designer a partir da
propriedade MasterFields.

4 Selecione o PartNo na lista Campos de detalhes.

5 Selecione o PartNo na lista Master Fields.

6 Clique no botão Adicionar. Um novo item (PartNo →
PartNo) será adicionado à lista de Campos Associados.

7 Clique no botão OK para fechar a caixa de diálogo.

Observação: neste exemplo, você notará que os campos vinculados têm o mesmo nome. Este é um bom padrão a ser seguido ao projetar um banco de dados. No entanto, não é um requisito para vincular campos.

8 Selecione Arquivo | Salve no menu principal do Delphi.

Atualize o título do relatório

1 Acesse o Report Designer.

2 Localize a etiqueta de resumo do pedido no canto
superior esquerdo da faixa do cabeçalho.

3 Defina a legenda como Detalhe do pedido.

Crie o SubReport2 e conecte-o aos dados

1 Clique na guia 'Sub-relatório1'.

2 Clique com o botão direito sobre o componente de linha
vertical localizado na extremidade esquerda da banda de
detalhes. Selecione as opções de menu StretchWithParent e
ReprintOnOverflow.

3 Clique com o botão direito sobre o componente de linha
vertical na extremidade direita da banda de detalhes. Selecione
as opções de menu StretchWithParent e ReprintOnOverflow.

Nota: A banda de detalhes conterá um sub-relatório do tipo
filho que se estenderá para imprimir todos os itens do pedido
atual. Ao definir StretchWithParent como True, estamos
dizendo "Qualquer que seja a altura da banda de detalhe no
relatório, redimensione a linha para corresponder a ela". Às
vezes, os itens de um pedido transbordam para uma página
adicional. Ao definir ReprintOnOverflow como True, estamos
dizendo "Quando a banda de detalhes transbordar para uma
página adicional, certifique-se de reimprimir a linha".

4 Defina a altura da banda de detalhe para 0,4687.

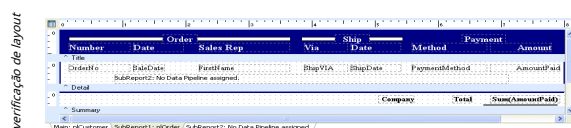
5 Coloque um componente de sub-relatório  logo abaixo do Componentes DBText na banda de detalhes.

6 Clique com o botão direito sobre o sub-relatório e defina ParentWidth como false e, em seguida, defina a posição:

Deixou	0,875
Principal	0,2708
Largura	6,25

7 Use a lista suspensa no canto superior esquerdo do Report Designer para definir o DataPipeline do sub-relatório como plItem.

8 Selecione Arquivo | Salve no menu principal do Delphi.



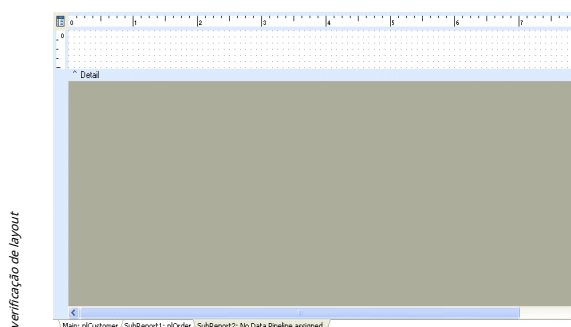
Remova o título e a faixa de resumo

1 Clique na guia 'SubReport2'.

2 Selecione Relatório | Título do menu Report Designer para remover a banda do título.

3 Selecione Relatório | Resumo no menu Report Designer para remover a banda de resumo.

Nota: No contexto de um sub-relatório filho, a faixa do título seria impressa uma vez no início do relatório e a faixa do resumo seria impressa uma vez no final. Isso seria adequado para nossos propósitos, exceto que todos os itens de um pedido específico podem não caber em uma única página e, portanto, o sub-relatório pode transbordar para páginas adicionais. Quando isso acontecer, queremos reimprimir os rótulos de cabeçalho para os campos de item. A propriedade ReprintOnSubsequent das bandas de cabeçalho de grupo funcionará bem para esse requisito, portanto, usaremos um cabeçalho de grupo e uma banda de rodapé de grupo em vez da banda de título e de resumo.



Crie um grupo no campo 'OrderNo'

1 Selecione Relatório | Grupos no menu principal do

Report Designer. A caixa de diálogo Grupos será exibida.

2 Selecione `plItem.OrderNo` na lista suspensa na parte superior da caixa de diálogo e clique no botão Adicionar. Uma nova entrada intitulada Group [0]: `plItem.OrderNo` será adicionada à lista.

3 Clique no botão OK. Você deve ver duas novas bandas: uma acima e outra abaixo da banda de detalhes. Essas faixas não têm altura; portanto, nenhum espaço em branco aparece acima das divisórias.

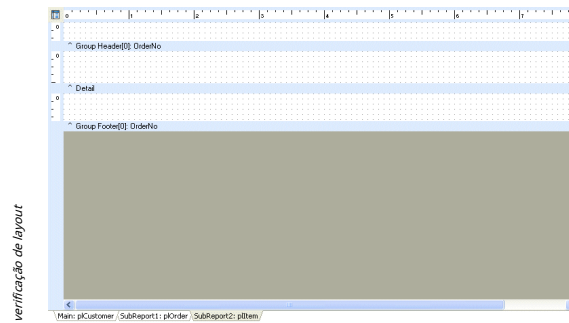
Observação: criamos um grupo no campo `OrderNo` porque esse é o campo de vinculação entre as tabelas `Order` e `Item`. O valor deste campo nunca será alterado durante o curso da geração do sub-relatório; assim, o grupo nunca vai 'quebrar'. O efeito será que o cabeçalho do grupo será impresso uma vez no início e uma vez para todas as páginas subsequentes. O rodapé do grupo será impresso uma vez no final. Você pode usar esta técnica 'sem interrupção' sempre que precisar de um cabeçalho de separação em um sub-relatório.

4 Defina a altura das novas bandas:

Banda do cabeçalho do grupo 0,30

Faixa de rodapé do grupo 0,45

5 Selecione Arquivo | Salve no menu principal do Delphi.



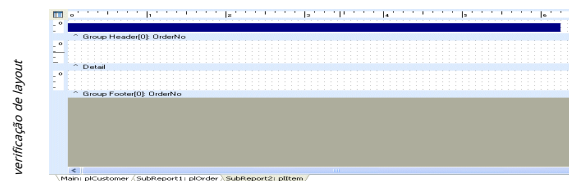
Esquematize a faixa do cabeçalho do grupo para o sub-relatório

1 Coloque um componente de forma no canto superior esquerdo da banda do cabeçalho do grupo.

2 Configure a forma com o componente:

Deixou	0
Largura	6,25
Altura	0,2
Cor de Preenchimento	Marinha
Cor da linha	Marinha

3 Mantenha pressionada a tecla Ctrl e use a tecla de seta para baixo para mover a forma até que a parte inferior esteja alinhada com a parte inferior da banda.



Comece a organizar os componentes de informações de itens para o sub-relatório

1 Selecione Exibir | Barras de ferramentas | Árvore de dados no menu Report Designer. Posicione a árvore de dados à esquerda do Report Designer.

2 Clique na guia Layout na parte inferior da Árvore de dados e defina as configurações de arrastar e soltar como segue:

Crio	Tudo
Estilo	Tabular
Nome da fonte do rótulo	Times New Roman
Estilo de fonte da etiqueta	Negrito
Tamanho da fonte da etiqueta	10 pontos
Cor da fonte do rótulo	Branco
Nome da fonte do campo	Times New Roman
Estilo da fonte do campo	Regular
Tamanho da fonte do campo	10 pontos
Cor da fonte do campo	Preto

3 Clique na guia Dados e selecione o pipeline de dados do plItem na lista no topo.

4 Selecione o campo ItemNo.

5 Arraste a seleção para a faixa de cabeçalho do grupo, posicionando o mouse apenas dentro do canto superior esquerdo da forma. Solte o botão do mouse. Um componente DBText e o rótulo correspondente serão criados.

6 Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

Deixou	0,1042
Principal	0,1354

7 Justifique os componentes à esquerda.

8 Defina a legenda do rótulo para o item nº

9 Selecione o pipeline de dados plPart na árvore de dados.

10 Selecione o campo Descrição.

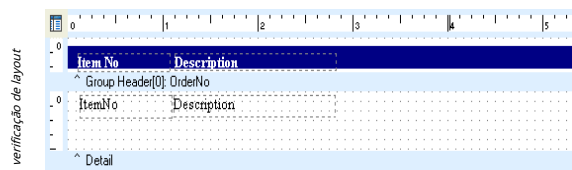
11 Arraste a seleção para a faixa do cabeçalho do grupo, posicionando o mouse imediatamente à direita do rótulo Nº do item. Solte o botão do mouse. Um componente DBText e o rótulo correspondente serão criados.

12Mantenha pressionada a tecla Ctrl e use as teclas de seta para posicionar a seleção:

Deixou	1,125
Principal	0,1354

13Mantenha pressionada a tecla Shift e use a tecla de seta para a esquerda para diminuir a largura de ambos os componentes para 1,7083.

14 Defina o topo de ambos os componentes DBText na banda de detalhes para 1,6667.



Complete o Layout de Informações de Itens

1 Selecione o campo Qty do pipeline de dados do plItem na Árvore de Dados.

2 Arraste a seleção para a direita imediata do Rótulo de descrição.

3 Posicione a seleção:

Deixou	2,8646
Principal	0,1354

4 Justifique os componentes à direita.

5 Defina a legenda do rótulo para Quantidade.

6 Alinhe a parte superior do novo DBText com a parte superior dos componentes DBText existentes.

7 Selecione o campo Lista de preços do pipeline de dados plPart.

8 Arraste a seleção para a direita imediata do Etiqueta de quantidade.

9 Posicione a seleção:

Deixou	3,8646
Principal	0,1354

10 Justifique os componentes à direita.

11 Defina a legenda do rótulo para o preço de tabela.

12 Defina o formato de exibição do componente DBText para \$ #, 0,00; (\$ #, 0,00).

13 Alinhe a parte superior do novo componente DBText com a parte superior dos componentes DBText existentes.

14 Coloque um componente de rótulo à direita do rótulo de preço de lista.

15 Configure o rótulo:

Tamanho automático	Falso
Rubrica	Total de itens
Nome da fonte	Times New Roman
Tamanho da fonte	10 pontos
Estilo de fonte	Negrito
Cor da fonte	Branco
Deixou	4,8646
Principal	0,1354
Largura	0,9792
Alinhamento de Texto	Justificado corretamente

16 Coloque um componente variável na banda de detalhes logo abaixo do rótulo Total do item.

17 Configure o componente variável:

Nome	vrItemTotal
Tipo de dados	dtCurrency
Nome da fonte	Times New Roman
Tamanho da fonte	10 pontos
Estilo de fonte	Regular
Cor da fonte	Preto
Alinhamento de Texto	Direito
Formato de apresentação	\$ #, 0,00; (\$ #, 0,00)
Largura	0,9792

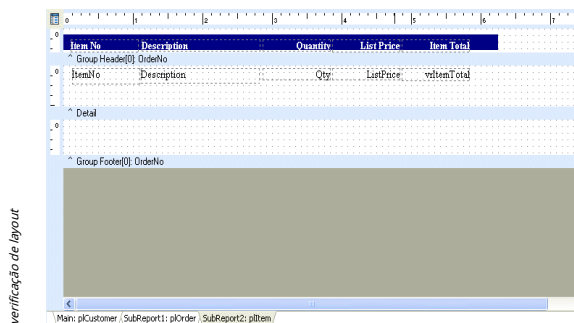
18 Selecione o rótulo Item Total e, em seguida, clique com a tecla Shift pressionada na variável vrItemTotal. Clique no ícone Alinhar à direita



na barra de ferramentas Alinhar ou Espaço.

19 Alinhe a parte superior do componente variável com a parte superior dos componentes DBText.

20 Selecione Arquivo | Salve no menu principal do Delphi.



Complete o Layout da Faixa de Detalhe para o Sub-relatório

1 Coloque um componente de forma no canto superior esquerdo da banda de detalhes.

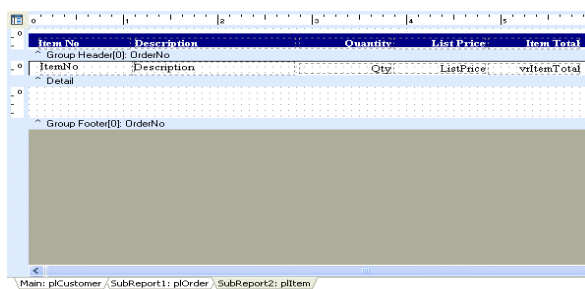
2 Configure o componente de forma:

Deixou	0,0
Principal	0,0
Largura	6,25
Altura	0,2187
Cor da linha	Preto
Cor de Preenchimento	Nenhum

3 Clique com o botão direito sobre a forma e escolha a opção de menu Enviar para trás.

4 Defina a altura da banda de detalhe para 0,2083.

Nota: Observe como a parte inferior da forma se estende um pixel da tela além do final da banda de detalhes. Essa técnica permite que as formas se sobreponham quando renderizadas na página, criando o efeito de uma única linha entre as bandas de detalhes.



Esquematize a faixa de rodapé do grupo para o sub-relatório

1 Selecione o componente de forma na banda de cabeçalho do grupo.

2 Copie e cole a seleção. Arraste a nova forma para a faixa de rodapé do grupo e posicione-a em o canto superior esquerdo:

Deixou	0,0
Principal	0,0

3 Selecione o rótulo Item Total na banda do cabeçalho do grupo.

4 Copie e cole a seleção. Arraste o novo rótulo para a faixa de rodapé do grupo.

5 Defina a legenda do rótulo como Total.

6 Selecione o rótulo Preço de lista na banda do cabeçalho do grupo e, em seguida, clique com a tecla Shift pressionada no rótulo Total. Clique no

Ícone Alinhar à direita na barra de ferramentas Alinhar ou Espaço.

7 Selecione a forma na faixa de rodapé do grupo e, em seguida, clique com a tecla Shift pressionada para selecionar o rótulo Total. Clique no ícone Align Center na barra de ferramentas Align ou Space.

8 Selecione a variável vrItemTotal na banda de detalhes.

9 Copie e cole a seleção. Arraste a nova variável para a faixa de rodapé do grupo.

Nota: Esta técnica de copiar / colar pode ser muito mais eficiente do que criar um novo componente porque os valores apropriados de DisplayFormat e Font são retornados.

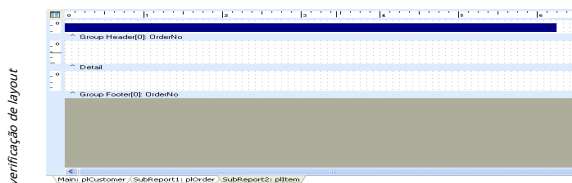
10 Configure a variável:

Nome	vrOrderTotal
Cor da fonte	Branco

11 Use a barra de ferramentas Alinhar ou Espaço para alinhar à direita a variável vrOrderTotal com a variável vrItemTotal.

12 Alinhe a parte superior da variável vrOrderTotal com a parte superior do rótulo Total.

13 Selecione Arquivo | Salve no menu principal do Delphi.

**Codifique os cálculos para os totais**

1 Selecione a variável vrItemTotal na banda de detalhes.

2 Selecione a guia Eventos do Inspetor de objetos.

3 Clique duas vezes no evento OnCalc. Um shell de manipulador de eventos será gerado em seu formulário Delphi.

4 Adicione o seguinte código a este manipulador de eventos:

```
Valor := plItem ['Qty'] * plPart ['ListPrice'];
```

Observação: este manipulador de eventos recupera a quantidade e o preço de lista do item e retorna o valor total por meio do parâmetro Value. O resultado será o valor do componente variável. O evento OnCalc será disparado uma vez para cada registro de item.

5 Retorne ao Report Designer e selecione a variável vrOrderTotal na faixa de rodapé do grupo.

6 Codifique o evento OnCalc:

```
Valor := Valor + vrItemTotal.Value;
```

Nota: Este manipulador de eventos recupera o total atual do item e o adiciona ao total atual do pedido. Este evento OnCalc também será disparado uma vez para cada registro de item.

7 Retorne ao Report Designer, clique com o botão direito do mouse na variável vrOrderTotal e selecione a opção de menu Timing

8 Selecione 'GroupEnd' na lista suspensa Reset On. Selecione 'Group0: OrderNo' na lista suspensa Grupo.

Nota: A caixa de diálogo Timing permite controlar quando o evento OnCalc será disparado e quando o valor da variável será redefinido. Para esse total, queremos calcular o valor cada vez que um registro é percorrido e queremos redefinir o valor depois que o rodapé do grupo tiver concluído a impressão.

9 Selecione o projeto | Compile rbMDDProj. Corrija quaisquer problemas de compilação.

10 Selecione Arquivo | Salve no menu principal do Delphi.

Preencha o Layout para SubReport1

1 Clique na guia 'SubReport1'.

2 Clique com o botão direito sobre o componente DBText Ordeno localizado no lado esquerdo da banda de detalhes. Defina a opção de menu ReprintOnOverflow como True.

Nota: Overflow ocorre quando a banda de detalhes fica sem espaço na página atual e deve concluir a impressão na página seguinte. Para este relatório, queremos imprimir o ItemNo seguido por 'Continuação ...' na parte superior da página quando os pedidos de um determinado cliente transbordam para uma página adicional.

3 Coloque um rótulo na banda de detalhes e configure-o:

Rubrica	Contínuo...
Nome	lblContinued
Visível	Falso
Nome da fonte	Times New Roman
Tamanho da fonte	10
Estilo de fonte	Regular
Cor da fonte	Preto
Alinhamento de Texto	Justificado à esquerda

4 Selecione o componente SaleDate DBText e, em seguida,

Clique com a tecla Shift pressionada no rótulo Continuação.

5 Clique em Alinhar Superior e Alinhar à Esquerda

ícones na barra de ferramentas Alinhar ou Espaço.

6 Adicione o seguinte código ao evento OnPrint do rótulo Continued:

```
lblContinued.Visible:=  
ppChildReport1.Detail.Overflow;
```

7 Selecione o projeto | Compile rbMDDProj. Corrija quaisquer problemas de compilação.

8 Selecione Arquivo | Salve no menu principal do Delphi.

Observação: a propriedade Overflow da banda de detalhes só será True quando os pedidos do cliente atual não couberem na página e, portanto, devem transbordar para uma página adicional. Quando isso acontecer, o rótulo ficará visível.

Converter a faixa de título em uma faixa de cabeçalho de grupo

1 Selecione Relatório | Grupos e criar um grupo em plOrder.CustNo. Desmarque a opção 'Manter o grupo junto'.

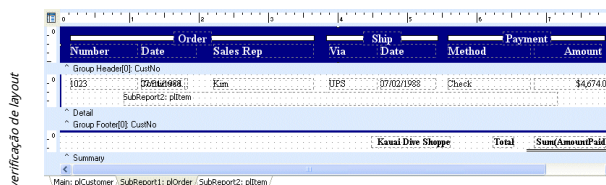
2 Defina a altura da banda do cabeçalho do grupo para 0,5208.

3 Clique com o botão direito sobre a forma na faixa do título e defina ParentHeight e ParentWidth como False.

4 Selecione todos os componentes na faixa de título, incluindo a forma.

5 Arraste a seleção para a banda do cabeçalho do grupo.

6 Selecione Relatório | Título do menu principal do Report Designer para remover a banda do título.



7 Visualize o relatório concluído.

Nota: você deve ser capaz de ter uma ideia geral de como as coisas se parecem. No entanto, você precisará visualizar o relatório em tempo de execução para ver os resultados dos cálculos e outros manipuladores de eventos.

Visualize o relatório em tempo de execução

1 Feche o Report Designer.

2 Selecione o projeto | Compile rbMDDProj. Corrija quaisquer problemas de compilação.

3 Selecione Arquivo | Salve no menu principal do Delphi.

4 Execute o projeto.

5 Visualize o relatório concluído. O relatório deve ter a seguinte aparência:

Order Detail
Marine Adventures & Sunken Treasure Co.

Company: Kawaii Dive Shoppe
Client No: 1221

Number	Date	Order	Sales Rep	Via	Ship	Date	Method	Payment	Amount
1013	07/01/09	Em		UPS	07/01/09	Check			\$5,479.00
Item No	Description	Quantity	Unit Price	Item Total					
1	Regularly Service	7	\$270.00						
2	DepthPressure Gauge	4	\$204.00						
3	Aluminum Inflator Regulator	10	\$140.00						
1074	11/04/09	Final		UPS	04/04/09	Visa			\$17,782.00
Item No	Description	Quantity	Unit Price	Item Total					
1	Emergency Oxygen Valve 8y	4	\$133.00						
2	Terrace Valve Camera (B01)	4	\$1,000.00						
3	Second Stop Regulator	1	\$100.00						
1133	08/24/09	Balance		UPS	08/24/09	Check			\$13,943.00
Item No	Description	Quantity	Unit Price	Item Total					
1	Second Stop Regulator	14	\$141.00						
2	Second Stop Regulator	24	\$141.00						
3	DepthPressure Gauge Camera	2	\$100.00						
4	Wet/Dry Thermometer (F)	2	\$10.00						
7	Clear/Pink Knife	2	\$41.00						
8	Penlight	3	\$47.00						
1149	07/04/09	Total		UPS	07/04/09	Check			\$50,471.00

3/29/2009 10:19:34 AM Page 1 of 93

Visualização interativa com sub-relatórios de detalhamento

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Use o recurso de detalhamento de sub-relatórios do tipo filho

Order Detail *Marine Adventures & Sunken Treasure Co.*

Company **Kauai Dive Shoppe**
Custo 1221

Order Number	Date	Sales Rep.	Via	Ship Date	Method	Payment	Amount
1023	07/01/1988	Kim	UPS	07/02/1988	Check		\$4,174.00
1074	12/14/1994	Phil	UPS	04/24/1995	Visa		\$17,781.00
1123	08/24/1995	Roberto	UPS	08/24/1995	Check		\$13,943.00
1149	07/04/1994	Tami	UPS	07/04/1994	Credit		\$2471.93
1174	07/04/1994	Cass1	UPS	07/04/1994	Visa		\$4,178.83
1249	12/14/1994	Ann	UPS	12/14/1994	Credit		\$1,466.00
Kauai Dive Shoppe Total							\$61,459.86

301.00 12:33 PM Page 1

Criar uma nova aplicação

Observação: este tutorial se baseia no relatório master-detaildetail criado na seção anterior.

Você pode concluir o tutorial anterior ou copiar o formulário rbMDD do diretório RBuilder \ Tutorials.

1 Selecione Arquivo | Novo aplicativo do menu

Delphi. Isso criará um novo projeto e um formulário em branco.

2 Feche o novo formulário e unidade sem salvar.

3 Selecione Arquivo | Abra no menu Delphi.

Localize a unidade rbMDD.pas e abra-a.

4 Altere o nome do formulário para

'frmDrillDownSubreport'.

5 Selecione Arquivo | Salvar como no menu Delphi e

salvar o formulário com o nome rbDrillD no diretório My RB Tutorials (estabelecido na página 181).

6 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

7 Clique com o botão direito sobre o nome do projeto no

Project Manager (geralmente Project1.exe) e selecione a opção de menu Adicionar. Adicione o formulário rbDrillD ao projeto.

8 Clique com o botão direito sobre o nome do projeto no

Project Manager e selecione a opção de menu Salvar.

9 Salve o projeto com o nome rbDDProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Chame o Report Designer e configure o Drill-Down

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

3 Selecione a variável de sistema PageCount na faixa de rodapé. Defina o tipo de variável para PageNoDesc.

4 Selecione Relatório | Configuração de passe | Uma passagem no menu principal do Report Designer.

Observação: os relatórios de detalhamento devem ser sempre definidos para uma passagem. Caso contrário, todo o relatório será gerado sempre que o usuário expandir o sub-relatório.

5 Clique na guia 'SubReport1'.

6 Coloque uma forma no lado esquerdo da faixa de detalhes.

7 Configure a forma:

Nome	shpClickMe
ParentWidth	Verdadeiro
Principal	0
Altura	0,25
Cor de Preenchimento	Amarelo
Cor da linha	Amarelo

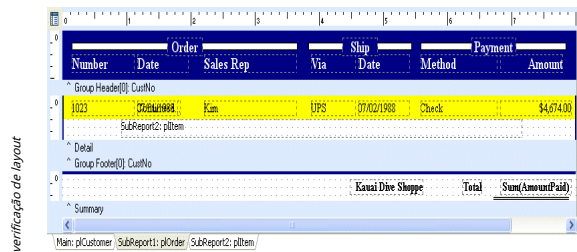
8 Clique com o botão direito sobre a forma e selecione a opção de menu Enviar para trás. A forma deve aparecer atrás dos outros componentes.

9 Clique com o botão direito sobre o sub-relatório do item e selecione DrillDown.

10 Selecione o shpClickMe na lista suspensa e clique em OK.

Observação: ao especificar a opção DrillDown do sub-relatório, associamos a forma amarela ao relatório. Depois de fazer isso, o sub-relatório não será impresso até que a forma seja clicada na janela de visualização de impressão. A caixa de diálogo DrillDown ... define a propriedade DrillDownComponent do SubReport.

11 Selecione Arquivo | Salve no menu principal do Delphi.



12 Visualize o relatório. O sub-relatório de itens não deve aparecer. Quando você move o mouse sobre a área amarela, o cursor deve mudar para uma mão apontando. Ao clicar na área amarela, o sub-relatório deve aparecer. Clicar na mesma área novamente deve fazer com que o sub-relatório desapareça.

O relatório deve ter a seguinte aparência:

Vista Inicial

ReportBuilder - rhOrderDetail

File Edit View Report Help

Data Calc Design Preview

62%

Cancel

Order Detail

Marine Adventures & Sunken Treasure Co.

Company: Kassal Dive Shoppe
Card# 1231

Number	Date	Order	Sales Rep	Via	Ship Date	Method	Payment	Amount
1013	07/01/09	Elem		TPP	07/01/09	Check		\$6,479.88
1075	11/14/09	Plat		TPP	09/14/09	Via		\$17,760.88
1113	08/04/09	Refurn		TPP	08/04/09	Check		\$11,961.88
1119	07/04/09	Scal		TPP	07/04/09	Check		\$6,475.87
1175	07/04/09	Scal		TPP	07/04/09	Via		\$6,170.87
1119	11/14/09	Scal		TPP	11/14/09	Check		\$1,489.88
Kassal Dive Shoppe								Total
								\$93,479.88

10/09/09 11:30:03 AM

Page 1

Os primeiros três sub-relatórios expandidos

ReportBuilder - rhOrderDetail

File Edit View Report Help

Data Calc Design Preview

62%

Cancel

Order Detail

Marine Adventures & Sunken Treasure Co.

Company: Kassal Dive Shoppe

Card# 1111

Number	Date	Order	Sales Rep	Via	Ship Date	Method	Payment	Amount																																			
1013	07/01/09	Elem		TPP	07/01/09	Check		\$6,479.88																																			
<table> <tr> <th>Item No</th> <th>Description</th> <th>Quantity</th> <th>Unit Price</th> <th>Item Total</th> </tr> <tr> <td>1</td> <td>Regulator System</td> <td>2</td> <td>\$17.89</td> <td></td> </tr> <tr> <td>2</td> <td>ScubaDiving Group</td> <td>4</td> <td>\$10.88</td> <td></td> </tr> <tr> <td>3</td> <td>Advanced Scuba Regulator</td> <td>10</td> <td>\$17.89</td> <td></td> </tr> </table>									Item No	Description	Quantity	Unit Price	Item Total	1	Regulator System	2	\$17.89		2	ScubaDiving Group	4	\$10.88		3	Advanced Scuba Regulator	10	\$17.89																
Item No	Description	Quantity	Unit Price	Item Total																																							
1	Regulator System	2	\$17.89																																								
2	ScubaDiving Group	4	\$10.88																																								
3	Advanced Scuba Regulator	10	\$17.89																																								
1075	11/14/09	Plat		TPP	09/14/09	Via		\$17,760.88																																			
<table> <tr> <th>Item No</th> <th>Description</th> <th>Quantity</th> <th>Unit Price</th> <th>Item Total</th> </tr> <tr> <td>1</td> <td>Advanced Openwater Value Set</td> <td>4</td> <td>\$17.89</td> <td></td> </tr> <tr> <td>2</td> <td>Scuba Value Course (SCV)</td> <td>4</td> <td>\$1,000.88</td> <td></td> </tr> <tr> <td>3</td> <td>Scuba Dive Regulator</td> <td>1</td> <td>\$10.88</td> <td></td> </tr> </table>									Item No	Description	Quantity	Unit Price	Item Total	1	Advanced Openwater Value Set	4	\$17.89		2	Scuba Value Course (SCV)	4	\$1,000.88		3	Scuba Dive Regulator	1	\$10.88																
Item No	Description	Quantity	Unit Price	Item Total																																							
1	Advanced Openwater Value Set	4	\$17.89																																								
2	Scuba Value Course (SCV)	4	\$1,000.88																																								
3	Scuba Dive Regulator	1	\$10.88																																								
1113	08/04/09	Refurn		TPP	08/04/09	Check		\$11,961.88																																			
<table> <tr> <th>Item No</th> <th>Description</th> <th>Quantity</th> <th>Unit Price</th> <th>Item Total</th> </tr> <tr> <td>1</td> <td>Scuba Dive Regulator</td> <td>11</td> <td>\$10.88</td> <td></td> </tr> <tr> <td>2</td> <td>Scuba Dive Regulator</td> <td>25</td> <td>\$10.88</td> <td></td> </tr> <tr> <td>3</td> <td>ScubaDiving Group Course</td> <td>2</td> <td>\$10.88</td> <td></td> </tr> <tr> <td>4</td> <td>Water-Proof Electronics (E)</td> <td>2</td> <td>\$10.88</td> <td></td> </tr> <tr> <td>5</td> <td>Classroom Package</td> <td>2</td> <td>\$10.88</td> <td></td> </tr> <tr> <td>6</td> <td>Regulator</td> <td>2</td> <td>\$10.88</td> <td></td> </tr> </table>									Item No	Description	Quantity	Unit Price	Item Total	1	Scuba Dive Regulator	11	\$10.88		2	Scuba Dive Regulator	25	\$10.88		3	ScubaDiving Group Course	2	\$10.88		4	Water-Proof Electronics (E)	2	\$10.88		5	Classroom Package	2	\$10.88		6	Regulator	2	\$10.88	
Item No	Description	Quantity	Unit Price	Item Total																																							
1	Scuba Dive Regulator	11	\$10.88																																								
2	Scuba Dive Regulator	25	\$10.88																																								
3	ScubaDiving Group Course	2	\$10.88																																								
4	Water-Proof Electronics (E)	2	\$10.88																																								
5	Classroom Package	2	\$10.88																																								
6	Regulator	2	\$10.88																																								
1119	07/04/09	Scal		TPP	07/04/09	Check		\$6,475.87																																			
1175	07/04/09	Scal		TPP	07/04/09	Via		\$6,170.87																																			
1119	11/14/09	Scal		TPP	11/14/09	Check		\$1,489.88																																			
Kassal Dive Shoppe								Total																																			
								\$93,479.88																																			

10/09/09 11:34:18 AM

Page 1

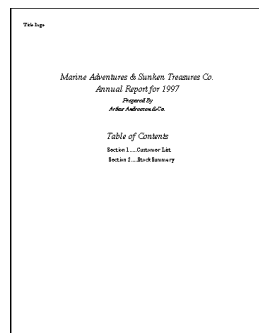
Conectando relatórios com sub-relatórios de estilo de seção

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Crie um único relatório a partir de dois relatórios diferentes
- Salvar e carregar layouts de relatório de um arquivo
- Reatribuir manipuladores de eventos quando um relatório é usado em um novo formulário

Folha de rosto



Primeira página do relatório da lista de clientes

Customer List					
Marine Adventures & Sunken Treasures					
Company	Contact	Phone	Address	City	State
First Data Shipyards	John Thomas	800-555-0001	4770 Sycamore Drive	Flagstaff	AZ
Umare	Sherry Winters	800-555-2019	PO Box 2447	Pomona	
John Deere	Phyllis Rogers	217-629-7038	1 Rogers Lane	East Naples	
Casino Drive World National	Jack Bailey	617-640-9046	PO Box 341	Grand Canyon	
Tree Service Drive Court	Chris Thomas	800-765-1031	611-1 Third Producers	Chico	IL Conn
Blue Sea - Apex Cruise	Berni Davis	401-485-7611	11-122 Washington Lane	Weymouth	MA
VP Drive Club	Barry Cummings	204-642-6976	21 West St.	Channahon	IL Conn
Onion Products	Paul Davies	205-557-0211	PO Box 2745	Kalamazoo	MI
Pennington-Agasson	Loise Wang	617-274-7648	221-100 N.17-A.A	Durham	
Hansen Drive Club	Jason Mack	410-642-6199	871 Olson St.	Kendall	OH
Two-Digit Change	Jan Williams	202-515-2792	15142 Sycamore Pkwy	Madison	FL
Blue Spine	Thomas Gates	610-975-6294	382-138 Ave. Box 748	Quakertown	PA
Hale's COBRA Club	Dennis Sims	217-649-4002	PO Box 2314	Kalamazoo	MI
James Club	Robert Spang	212-278-4109	PO Box 1611-P	Savannah	FL
Hansen COBRA Cruise	Dakota Harris	617-240-7642	PO Box 62	Prophet	Massachusetts
West Pacific	Donald Chang	714-412-0877	411-112 Lane Avenue	S. Dennis	MA
Address Database	Oliver Davidson	617-240-9024	PO Box 744	North City	
Blue Spine Club	Oliver Davidson	617-240-7642	611-83 Pop Pine Lane	Largo	FL
Paul's Drive Loggia	David Palmer	202-515-2792	1415 Peak Hill St.	Durham	OR
Open Joint Card	Terrell Rogers	202-515-4111	146-136-136 Pine	Yonkers	NY
COBRA Home	Robert Johnson	617-240-7642	PO Box 2314	Prophet	
Pennington-Agasson	Loise Wang	617-274-7648	PO Box 1611-P	Savannah	FL
Drive of Code, Inc	Charles Logan	204-642-6976	Hansen Place 14	Jason Williams	Calif.
P.A. Sengupta	Rudolph Chen	714-358-4037	41-136 Lane	Madison	FL
Sherry Davis & Co	Bill Weiss	800-442-0771	471-136 Lane Way	Logoff	NY

Primeira página do relatório da lista de clientes

Customer List					
Marine Adventures & Sunken Treasures					
Company	Contact	Phone	Address	City	State
First Data Shipyards	John Thomas	800-555-0001	4770 Sycamore Drive	Flagstaff	AZ
Umare	Sherry Winters	800-555-2019	PO Box 2447	Pomona	
John Deere	Phyllis Rogers	217-629-7038	1 Rogers Lane	East Naples	
Casino Drive World National	Jack Bailey	617-640-9046	PO Box 341	Grand Canyon	
Tree Service Drive Court	Chris Thomas	800-765-1031	611-1 Third Producers	Chico	IL Conn
Blue Sea - Apex Cruise	Berni Davis	401-485-7611	11-122 Washington Lane	Weymouth	MA
VP Drive Club	Barry Cummings	204-642-6976	21 West St.	Channahon	IL Conn
Onion Products	Paul Davies	205-557-0211	PO Box 2745	Kalamazoo	MI
Pennington-Agasson	Loise Wang	617-274-7648	221-100 N.17-A.A	Durham	
Hansen Drive Club	Jason Mack	410-642-6199	871 Olson St.	Kendall	OH
Two-Digit Change	Jan Williams	202-515-2792	15142 Sycamore Pkwy	Madison	FL
Blue Spine	Thomas Gates	610-975-6294	382-138 Ave. Box 748	Quakertown	PA
Hale's COBRA Club	Dennis Sims	217-649-4002	PO Box 2314	Kalamazoo	MI
James Club	Robert Spang	212-278-4109	PO Box 1611-P	Savannah	FL
Hansen COBRA Cruise	Dakota Harris	617-240-7642	PO Box 62	Prophet	Massachusetts
West Pacific	Donald Chang	714-412-0877	411-112 Lane Avenue	S. Dennis	MA
Address Database	Oliver Davidson	617-240-9024	PO Box 744	North City	
Blue Spine Club	Oliver Davidson	617-240-7642	611-83 Pop Pine Lane	Largo	FL
Paul's Drive Loggia	David Palmer	202-515-2792	1415 Peak Hill St.	Durham	OR
Open Joint Card	Terrell Rogers	202-515-4111	146-136-136 Pine	Yonkers	NY
COBRA Home	Robert Johnson	617-240-7642	PO Box 2314	Prophet	
Pennington-Agasson	Loise Wang	617-274-7648	PO Box 1611-P	Savannah	FL
Drive of Code, Inc	Charles Logan	204-642-6976	Hansen Place 14	Jason Williams	Calif.
P.A. Sengupta	Rudolph Chen	714-358-4037	41-136 Lane	Madison	FL
Sherry Davis & Co	Bill Weiss	800-442-0771	471-136 Lane Way	Logoff	NY

Transfira o relatório da lista de clientes para o formulário

1 Abra o formulário que contém o relatório Lista de Clientes (rbCust.pas de "Um Relatório Simples ...").

2 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

3 Selecione Arquivo | Salvar como no menu principal do Report Designer.

4 Salve o modelo de relatório no diretório My RB Tutorials com o nome CustList.

Nota: Salvar um layout de relatório em arquivo é uma maneira rápida e fácil de tornar o relatório mais portátil.

Posteriormente neste tutorial, abriremos esse layout salvo em um sub-relatório. Layouts também podem ser carregados em tempo de execução usando o método `Report.Template.LoadFromFile` ou `LoadFromDatabase`.

5 Feche o Report Designer.

6 Selecione os componentes de acesso a dados (tblCustomer, dsCustomer, plCustomer).

7 Copie os componentes em sua área de transferência.

8 Selecione Exibir | Formulários do menu principal do Delphi. Clique duas vezes em frmSectionSubreports para tornar este formulário visível.

9 Cole os componentes neste formulário.

10 Coloque um rótulo Delphi padrão no formulário, diretamente acima desses componentes.

11 Defina a legenda do rótulo como 'Dados da lista de clientes'.

12 Defina a propriedade Visible do rótulo como False.

Nota: Este rótulo invisível não será usado pelo relatório: ele simplesmente serve como um lembrete de como esses componentes de acesso a dados são usados. Sempre que houver vários conjuntos de componentes de acesso a dados em um formulário, é uma boa ideia rotulá-los. Uma maneira alternativa de fazer isso é definir o botão Ferramentas | Opções de ambiente | Preferências | Mostrar a opção de legendas de componentes ativada, mas este método requer que você espace os componentes mais amplamente do que achamos preferível.

13 Feche a unidade rbCust.

Transfira o relatório de resumo de estoque para o formulário

1 Abra o formulário que contém o relatório de Resumo de Estoque (rbStock.pas do tutorial "Grupos, Cálculos, ...").

2 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

3 Selecione Arquivo | Salvar como no menu principal do Report Designer.

4 Salve o modelo de relatório no diretório My RB Tutorials com o nome StockSum.

5 Feche o Report Designer.

6 Selecione os componentes de acesso aos dados (qryStock, dsStock, plStock).

7 Copie esses componentes em sua área de transferência.

8 Selecione Exibir | Formulários do menu principal do Delphi. Clique duas vezes em frmSectionSub para tornar este formulário visível.

9 Cole os componentes no formulário e arraste-os para uma posição abaixo dos componentes de acesso a dados da Lista de clientes.

10 Coloque um rótulo Delphi padrão no formulário, diretamente acima desses componentes.

11 Defina a legenda como 'Dados de resumo de estoque'.

12 Defina a propriedade Visible do rótulo como False.

13 Feche a unidade rbStock.

14 Selecione Arquivo | Salve no menu principal do Delphi.

Criar e configurar o relatório principal

1 Adicionar um componente de relatório  para o formulário.

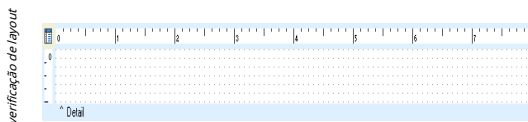
2 Nomeie o componente Relatório como rbSectSub.

3 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

4 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

5 Selecione Relatório | Cabeçalho do menu Report Designer. A banda do cabeçalho será removida.

6 Selecione Relatório | Rodapé. A faixa de rodapé será removida.



Crie o sub-relatório da lista de clientes

1 Coloque um componente SubReport na banda de detalhes.

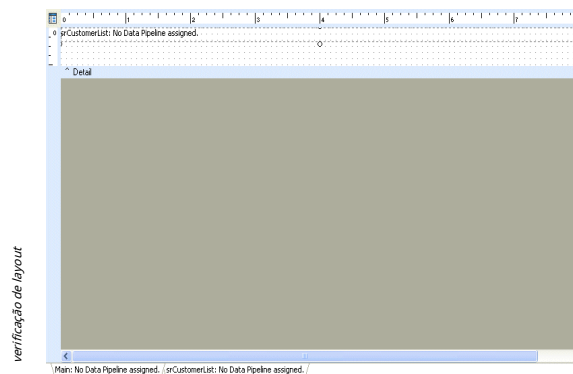
2 Posicione o sub-relatório de forma que fique alinhado com a parte superior da faixa de detalhes.

3 Clique com o botão direito do mouse no sub-relatório e selecione a opção de menu Seção.

Nota: Quando um sub-relatório é criado, o padrão é um PrintBehavior de pbChild. Os sub-relatórios filhos são impressos no contexto da banda pai, como um memorando extenso. Sub-relatórios de seção geram páginas inteiras, criando uma seção inteira dentro do relatório pai.

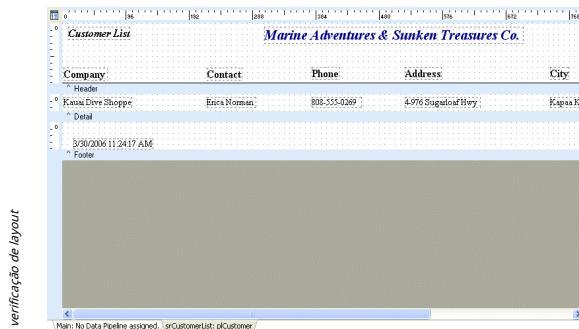
4 Nomeie o sub-relatório como srCustomerList.

Nota: Sempre que você define o nome de um sub-relatório, o texto no componente do sub-relatório e o texto na guia na parte inferior do Report Designer são atualizados. A legenda também é exibida na seção Report Outline da Report Tree. Quando você tem vários sub-relatórios, é uma boa ideia nomeá-los para que seja mais fácil controlá-los.



5 Clique na guia 'srCustomerList'.

6 Selecione Arquivo | Carregue o SubReport do menu principal do Report Designer e abra o arquivo CustList.rtm. O relatório Lista de clientes aparecerá no Report Designer.



7 Clique na guia Principal na parte inferior do Report Designer para retornar ao layout do relatório principal.

8 Selecione Arquivo | Salve no menu principal do Delphi.

Crie o sub-relatório de resumo de estoque

1 Coloque um segundo componente SubReport na banda de detalhes.

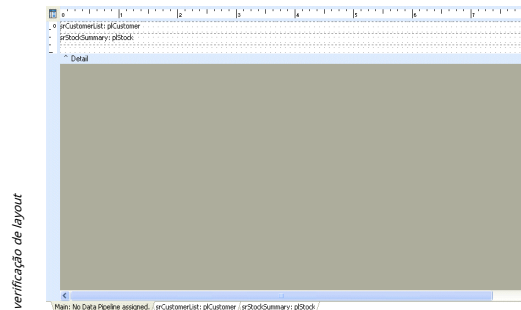
2 Posicione-o de forma que a parte superior fique alinhada com a parte inferior do sub-relatório Lista de clientes.

Observação: esta etapa não é necessária: ela apenas ajuda a criar um layout limpo. A ordem de impressão dos sub-relatórios não é determinada por sua ordem de cima para baixo dentro da banda. A ordem de impressão é, na verdade, determinada pela disposição em camadas dos componentes (ordem Enviar para Trás / Trazer para Frente). O relatório no verso é impresso primeiro; o relatório da frente é impresso por último. Você pode determinar rapidamente as camadas de sub-relatórios por meio da Árvore de Relatórios. Você também pode usar a Árvore de Relatório para alterar as camadas.

3 Clique com o botão direito do mouse no sub-relatório e selecione a opção de menu Seção.

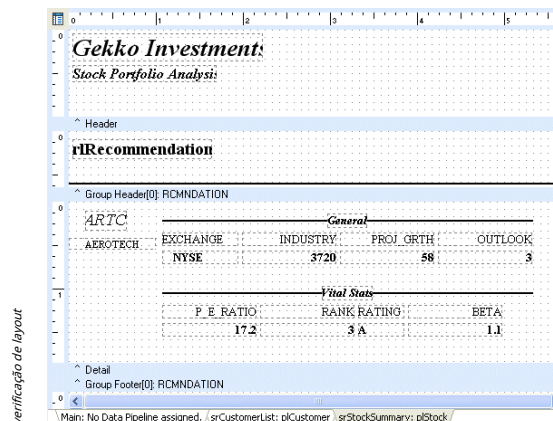
4 Nomeie o sub-relatório como srStockSummary.

Nota: Precisamos definir o nome deste sub-relatório porque iremos nos referir a este componente por nome em um manipulador de eventos posteriormente no tutorial.



5 Clique na guia 'srStockSummary'.

6 Selecione Arquivo | Carregue o SubReport do menu principal do Report Designer e abra o arquivo StockSum.rtm. O relatório de resumo de estoque aparecerá no Report Designer. Ignore qualquer mensagem aqui sobre 'Valores de propriedade inválidos', pois eles estão relacionados aos manipuladores de eventos atribuídos anteriormente ao relatório.



7 Clique na guia 'Principal' na parte inferior do Report Designer.

8 Selecione Arquivo | Salve no menu principal do Delphi.

Visualize o relatório em tempo de execução

1 Feche o Report Designer.

2 Selecione a guia Padrão da paleta de componentes Delphi.

3 Adicione um componente Botão ao formulário.

4 Configure o botão co componente:

Nome	btnPreview
Rubrica	Antevisão

5 Coloque o seguinte código no manipulador de eventos OnClick do botão:

```
rbSectSub.Print;
```

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Execute o projeto.

8 Clique no botão Visualizar. O relatório deve ser exibido no formulário Visualizar impressão. As primeiras três páginas devem conter o relatório Lista de clientes; as últimas trinta e nove páginas devem conter o relatório de resumo do estoque.

Observação: os cálculos do relatório de resumo de estoque não estão corretos e a codificação por cores deste relatório desapareceu. Isso ocorre porque não criamos os manipuladores de eventos para este relatório. Concluiremos essa tarefa na próxima seção.

Copie os manipuladores de eventos do relatório de resumo de estoque

1 Abra novamente o formulário contendo o relatório de Resumo de Estoque (rbStock.pas).

2 Localize as declarações do manipulador de eventos na declaração do formulário na parte superior da unidade.

3 Copie as seguintes declarações do manipulador de eventos para a área de transferência:

```
procedimento ppGroupHeaderBand1BeforeGenerate;  
procedimento ppDetailBand1BeforeGenerate;  
procedimento vrBuyTotalCalc;  
procedimento vrHoldTotalCalc;  
procedimento vrSellTotalCalc;
```

4 Localize a declaração da classe do formulário no topo da unidade rbSectSR.

5 Cole a declaração do manipulador de eventos imediatamente acima da seção privada da declaração da classe do formulário.

Nota: Você deve ter notado que o Delphi Form Designer adiciona automaticamente declarações à seção misteriosamente não rotulada no topo de sua declaração de classe de formulário. Esta é, na verdade, a seção publicada da classe do formulário. As declarações de todos os manipuladores de eventos e componentes de um formulário são colocadas na seção publicada para facilitar a lógica de streaming usada para carregar e salvar formulários em arquivos dfm. Ao colar essas declarações na seção publicada do formulário (e colar as implementações correspondentes na unidade), tornamos esses manipuladores de eventos atribuíveis a partir do Inspetor de Objetos.

6 Retorne à unidade rbStock e copie a variável 'FRecommendation' da seção privada da declaração de classe do formulário.

7 Retorne à unidade rbSectSR e cole esta variável na seção privada da declaração de classe do formulário.

8 Retorne para a unidade rbStock.

9 Role para baixo até a seção de implementação da unidade e copie todos os manipuladores de eventos, salve o evento TfrmStockSummary.btnPreviewClick.

10 Retorne à unidade rbSectSR e cole os manipuladores de eventos na seção de implementação da unidade.

11 Selecione Arquivo | Salve no menu principal do Delphi.

12 Retorne para a unidade rbStock. Clique com o botão direito do mouse no editor de código e selecione Fechar página para fechar esta unidade.

Reanexe os manipuladores de eventos ao sub-relatório de resumo de estoque

1 Role até o topo da unidade rbSectSR, clique duas vezes no nome da classe TfrmSectionSubreports e copie-o para a área de transferência.

2 Role para baixo até a seção de implementação e substitua o nome da classe existente (TfrmStockSummary) para cada manipulador de eventos pelo novo nome da classe (TfrmSectionSubreports).

3 Clique duas vezes no componente de relatório rbSectSub para exibir o Report Designer.

4 Clique na guia srStockSummary na parte inferior do Report Designer.

5 Clique no espaço em branco da faixa do cabeçalho do grupo.

6 Selecione a guia Eventos do Inspetor de objetos.

7 Selecione o evento BeforeGenerate, expanda a lista suspensão de manipuladores de eventos e selecione esta rotina:

ppGroupHeaderBand1BeforeGenerate

8 Selecione o evento AfterGenerate para certificar-se de que a atribuição foi bem-sucedida.

9 Clique no espaço em branco da banda de detalhes.

10 Selecione o evento BeforeGenerate, expanda a lista suspensão de manipuladores de eventos e selecione esta rotina:

ppGroupHeaderBand1BeforeGenerate

11 Selecione o evento AfterGenerate para certificar-se de que a atribuição foi bem-sucedida.

12 Clique no espaço em branco da banda do cabeçalho do grupo mais uma vez e clique duas vezes no evento BeforeGenerate para ativar o editor de código. Altere a instrução 'if' na última linha do manipulador de eventos para se referir a srStockSummay em vez de rbStockSum. O código deve ser:

E se srStockSummary.Report.Groups [0] .FirstPage
então

Observação: o manipulador de eventos original referia-se ao componente de relatório rbStockSum. Este manipulador de eventos deve ser alterado para se referir ao componente Relatório no sub-relatório srStock. Na verdade, um componente de sub-relatório consiste em dois componentes: o componente de sub-relatório e o componente de relatório contido nele. O componente de sub-relatório é usado para posicionar o sub-relatório dentro do relatório pai e para definir várias propriedades

relacionados ao comportamento de impressão. O componente do relatório pode ser acessado por meio da propriedade Relatório e contém o próprio relatório. Este componente não é diferente do componente de relatório com o qual você trabalha em um formulário Delphi.

13 Role para baixo até a faixa de resumo e reconecte os componentes variáveis aos seus correspondentes Eventos OnCalc:

Componente	Evento
vrBuyTotal	vrBuyTotalCalc
vrHoldTotal	vrHoldTotalCalc
vrSellTotal	vrSellTotalCalc

Nota: Este tutorial ilustra como reconectar os manipuladores de eventos do relatório Stock Summary para que você entenda melhor a situação. Se tivéssemos copiado esses manipuladores de eventos para a unidade SectSub antes de carregarmos o relatório Stock Summary no sub-relatório, os manipuladores de eventos seriam reconectados automaticamente pela lógica de streaming de objetos do Delphi. (Você deve se lembrar que uma mensagem de erro informando que os manipuladores de eventos não puderam ser encontrados foi exibida quando carregamos o layout do relatório.) É importante saber que os manipuladores de eventos são reconectados com base em seus nomes de procedimento. Quando você deseja que os manipuladores de eventos se reconectem automaticamente, eles devem ter os mesmos nomes daqueles salvos no layout do relatório.

14 Selecione Arquivo | Salve no menu principal do Delphi.

15 Execute o projeto. O relatório de resumo de estoque agora deve ser codificado por cores e os totais devem ser calculados corretamente.

Adicione uma página de título ao relatório

1 Clique duas vezes no componente de relatório rbSectSub para exibir o Report Designer.

2 Clique na guia 'Principal'.

3 Selecione Relatório | Título do menu principal do Report Designer. Uma banda-título será criada.

4 Clique com o botão direito sobre a faixa do título e selecione a opção de menu Posição

5 Defina a altura para 5,8125.

Comece a definir a faixa do título

1 Coloque um componente de rótulo no canto superior esquerdo da banda de título.

2 Configure o rótulo:

Rubrica	Folha de rosto
Nome da fonte	Times New Roman
Tamanho da fonte	12
Estilo de fonte	Negrito
Cor da fonte	Preto
Alinhamento de Texto	Deixou

3 Coloque um componente de rótulo próximo ao meio do banda do título.

4 Configure o rótulo:

Rubrica	Aventuras marinhas e Sunken Treasures Co.
Nome da fonte	Times New Roman
Tamanho da fonte	20
Estilo de fonte	Negrito itálico
Alinhamento de Texto	Centro
Deixou	1,3542
Principal	1,9166

5 Coloque um componente de rótulo logo abaixo do anterior rótulo.

6 Configure o rótulo:

Rubrica	Relatório Anual para
	1997
Nome da fonte	Times New Roman
Tamanho da fonte	20
Estilo de fonte	Negrito itálico
Alinhamento de Texto	Centro
Deixou	2,5729
Principal	2,3021

7 Coloque um componente de rótulo logo abaixo do anterior rótulo.

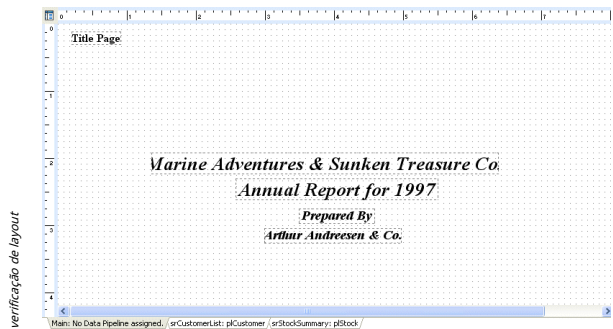
8 Configure o rótulo:

Rubrica	Preparado por
Nome da fonte	Times New Roman
Tamanho da fonte	14
Estilo de fonte	Negrito itálico
Alinhamento de Texto	Centro
Deixou	3.4896
Principal	2,75

9 Coloque um componente de rótulo logo abaixo do rótulo anterior.

10 Configure o rótulo:

Rubrica	Arthur Andreesen e
	Co.
Nome da fonte	Times New Roman
Tamanho da fonte	14
Estilo de fonte	Negrito itálico
Alinhamento de Texto	Centro
Deixou	2,9896
Principal	3.0416



Complete o Layout da Faixa de Título

1 Coloque um componente de rótulo logo abaixo do anterior rótulo.

2 Configure o rótulo:

Rubrica	Índice
Nome da fonte	Times New Roman
Tamanho da fonte	20
Estilo de fonte	Negrito itálico
Alinhamento de Texto	Centro
Deixou	2,9375
Principal	3,8646

3 Coloque um componente de rótulo logo abaixo do rótulo anterior.

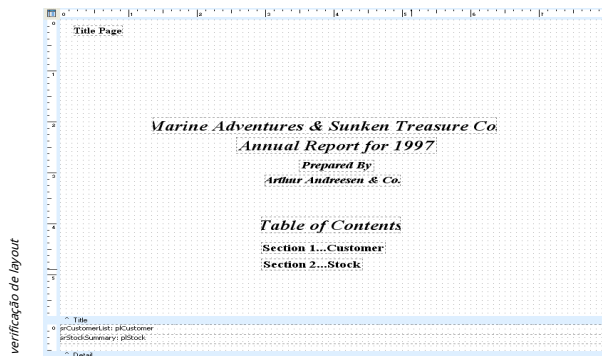
4 Configure o rótulo:

Rubrica	Seção 1 Cliente
Nome da fonte	Times New Roman
Tamanho da fonte	14
Estilo de fonte	Negrito
Alinhamento de Texto	Centro
Deixou	2,9375
Principal	4,3646

5 Coloque um componente de rótulo logo abaixo do anterior rótulo.

6 Configure o rótulo:

Rubrica	Seção 2 Estoque
Nome da fonte	Times New Roman
Tamanho da fonte	14
Estilo de fonte	Negrito
Alinhamento de Texto	Centro
Deixou	2,9375
Principal	4,6875

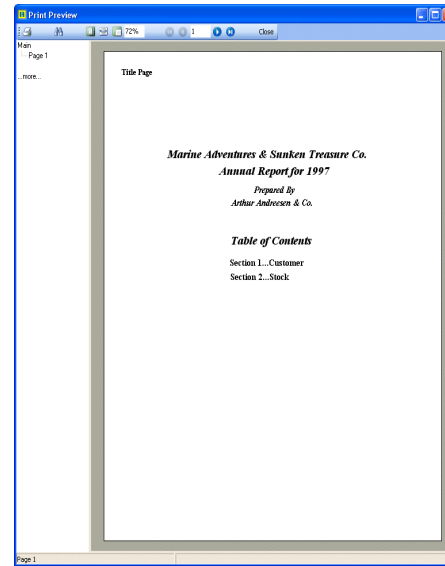


7 Selecione o projeto | Compile rbSSProj. Corrija quaisquer problemas de compilação.

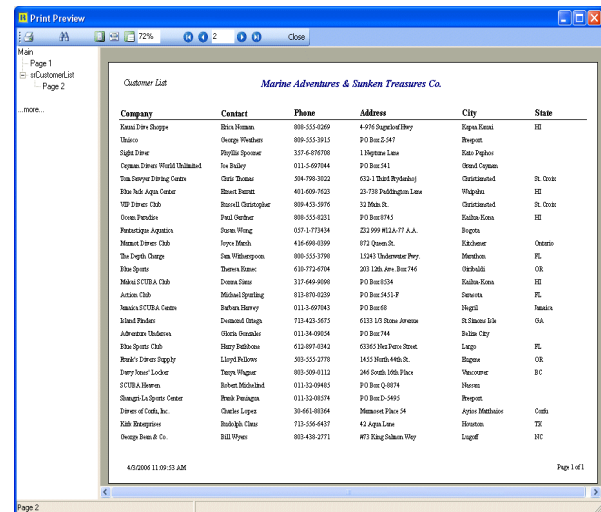
8 Selecione Arquivo | Salve no menu principal do Delphi.

9 Execute o projeto. A primeira página de cada seção deve ser semelhante a esta:

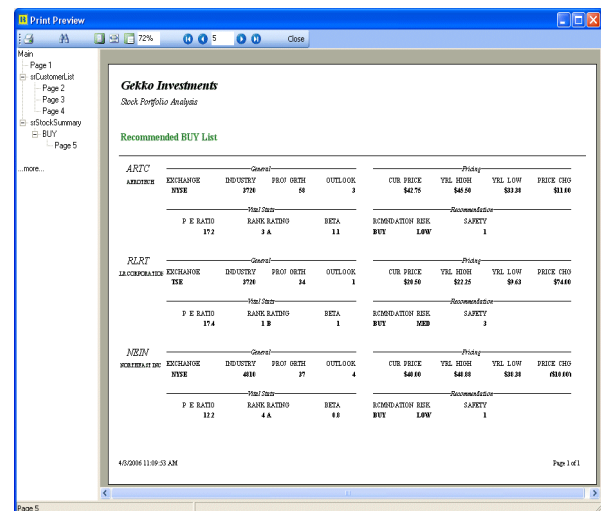
Folha de rosto



Primeira página do relatório da lista de clientes



Primeira página do relatório de resumo de estoque



Usando colunas para criar etiquetas de endereçamento

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configure um relatório para imprimir etiquetas de endereçamento

[illegible]

Criar uma nova aplicação

1 Seleione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmMailingLabels'.

3 Seleccione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome `rbMailL` no diretório `My RB Tutoriais` (estabelecido na página 181).

4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção Salvar.

6 Salve o projeto com o nome `rbMLProj` no diretório `My RB Tutorials`.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Chame o Report Designer e configure o layout da página

1 Selecione a aba RBuilder do Delphi compo-paleta atual.

2 Coloque um componente de relatório  No formulário.

3 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

4 Dimensione e mova a janela Report Designer para que o Object Inspector fique visível.

5 Selecione Arquivo | Configuração da página no menu principal do Report Designer.

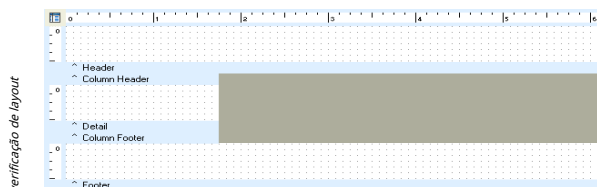
6 Selecione a guia Layout. Insira o seguinte valores:

Colunas	4
Largura da coluna	1,75
Posição da coluna 1	0,28
Posição da coluna 2	2,34
Posição da coluna 3	4,4
Posição da coluna 4	6,46

7 Selecione a guia Margens. Insira o seguinte valores:

Principal	0,4
Inferior	0,49
Deixou	0,2969
Direito	0,2969

8 Clique no botão OK para fechar a caixa de diálogo.



Configure o relatório

1 Clique no ícone Selecionar Relatório  no canto superior esquerdo canto da área de trabalho do Report Designer.

2 Defina PageLimit como 1 e a propriedade AutoStop como False no Object Inspector.

Observação: este relatório não está conectado a nenhum dado, portanto não tem base para interromper a geração de páginas. Nessa situação, o padrão da propriedade AutoStop é True para que o relatório imprima apenas uma banda de detalhes. Como queremos preencher uma única página com faixas de detalhes, desligamos o AutoStop e definimos o PageLimit como um.

3 Selecione Relatório | Cabeçalho do menu principal do Report Designer. A banda do cabeçalho será removida do relatório.

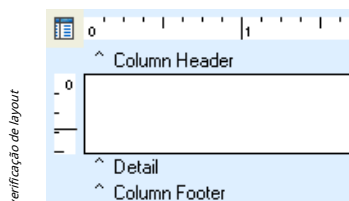
4 Selecione Relatório | Rodapé do menu principal do Report Designer. A faixa de rodapé será removida do relatório.

5 Coloque uma forma na faixa de detalhes.

6 Configure a forma:

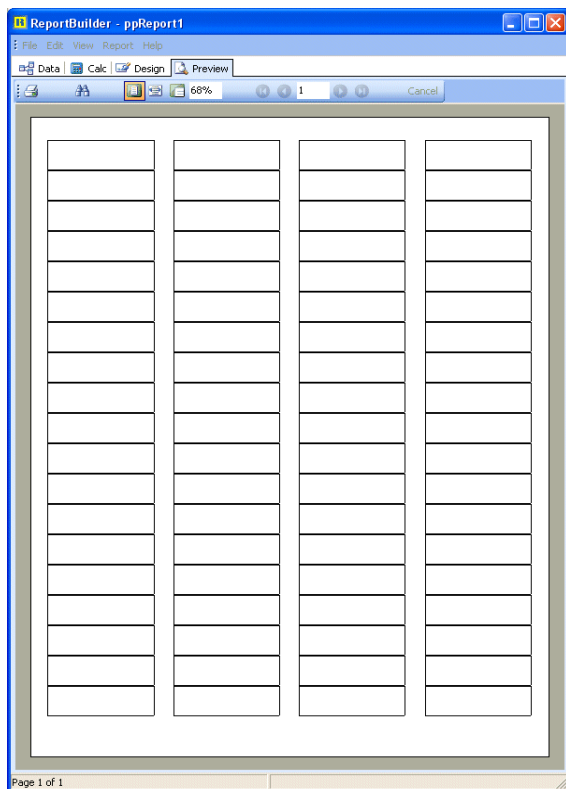
ParentHeight	Verdadeiro
ParentWidth	Verdadeiro

7 Selecione Arquivo | Salve no menu principal do Delphi.



8 Visualize o relatório. Você deve ver quatro colunas de formas. Imprima esta página na impressora.

Os rótulos devem ter a seguinte aparência quando você visualiza:



9 Pegue a folha recém-impressa e coloque-a atrás de uma folha de etiquetas Avery 5267 Laser. Segure as folhas contra uma luz forte para que você possa ver exatamente onde as linhas caem nas etiquetas. O ideal é que as bordas das linhas coincidam com as bordas das etiquetas.

Nota: Ao colocar uma forma na faixa de detalhe, podemos ver exatamente onde as informações da etiqueta serão impressas. Se a forma não corresponder aos rótulos, podemos fazer ajustes nas margens, posições das colunas e espaçamento entre linhas do detalhe

banda. É bastante comum que o formato do alinhamento fique um pouco fora do normal, embora as medidas exatas sejam enviadas para a impressora. Isso geralmente é devido à variação nas folhas de etiquetas e à calibração da impressora que está sendo usada. Geralmente, você deve ajustar o layout do relatório até que a forma de alinhamento apareça no local correto e, em seguida, definir a propriedade Visível da forma como False e usar o modelo como qualquer outro relatório.

Crie etiquetas de endereçamento por meio do assistente de modelo de etiqueta

1 Coloque outro componente de relatório no formulário e clique duas vezes nele para exibir o designer.

2 Selecione Arquivo | Novo no menu principal do designer.

3 Clique duas vezes no ícone Modelos de etiqueta.

4 Faça as seguintes seleções:

Produtos Avery Standard

5267 -ReturnAddress

5 Coloque uma forma na faixa de detalhes.

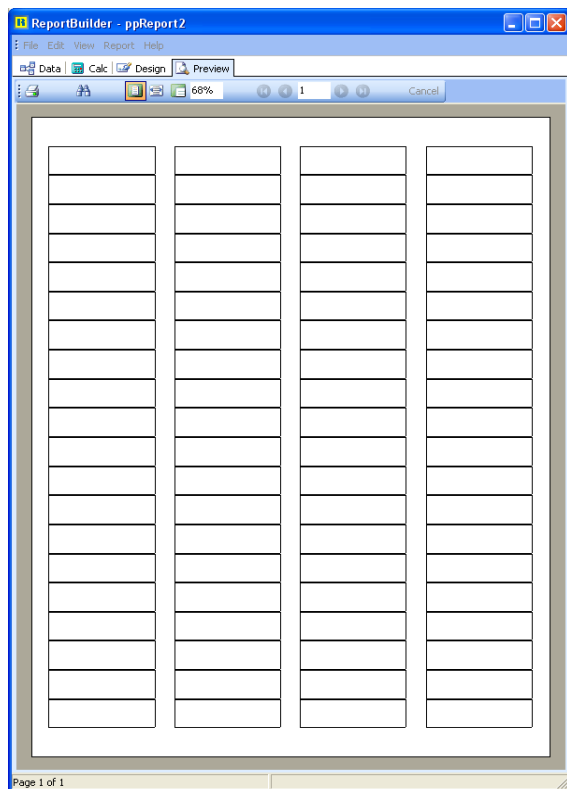
6 Defina ParentHeight e ParentWidth da forma como True.

7 Clique no ícone Selecionar Relatório e use o Inspetor de Objetos para definir AutoStop como False e PageLimit como 1.

8 Selecione Arquivo | Salve no menu principal do Delphi.

9 Pré-visualizar.

Os rótulos devem ser assim:



Impressão para um arquivo de texto

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Imprimir um relatório em um arquivo de dados ASCII

Criar uma nova aplicação

Observação: este tutorial se baseia no tutorial 'Criando um relatório por meio do assistente de relatório'. Você pode concluir este tutorial ou usar o formulário rbViaWiz localizado no diretório RBuilder \ Tutorials.

1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.

2 Feche o novo formulário e unidade sem salvar.

3 Selecione Arquivo | Abra no menu principal do Delphi. Localize a unidade rbViaWiz no diretório Tutoriais e abra-a.

4 Defina o nome do formulário como 'frmPrintToTextFile'.

5 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbToTxt no diretório My RB Tutorials (estabelecido na página 181).

Nota: É importante nomear o formulário e salvar a unidade do formulário usando os nomes fornecidos acima porque o formulário será usado em tutoriais posteriores.

6 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

7 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Adicionar.

8 Adicione rbToTxt ao projeto.

9 Clique com o botão direito sobre o nome do projeto mais uma vez e selecione a opção de menu Salvar.

10 Salve o projeto com o nome rbTTProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Conclua o tutorial do assistente de relatório

1 Renomeie o componente Relatório rbPrintData.

2 Exclua o botão Visualizar do formulário.

Carregar o modelo de relatório de estilo tabular

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Selecione Arquivo | Abra no menu principal do Report Designer.

3 Abra o arquivo CustTab.rtm.

Atribuir nomes de usuário aos componentes DBText

1 Defina o UserName para cada um dos componentes DBText com base no campo ao qual eles são atribuídos. Use o Inspetor de objetos para atribuir os seguintes nomes de usuário:

dbtCompany
dbtCustNo
dbtContact
dbtPhone
dbtFax
dbtCity
dbtState
dbtCountry

Especifique o nome e o formato do arquivo

1 Selecione Arquivo | Imprimir em configuração de arquivo no menu principal do Report Designer. A caixa de diálogo Configurar impressão em arquivo será exibida.

2 Clique no botão Arquivo.

3 Acesse o diretório My RB Tutorials e insira customer como o nome do arquivo. Clique no botão Salvar para fechar a caixa de diálogo do arquivo.

4 Selecione Delimitado por guia na lista suspensa Tipo de arquivo.

5 Selecione a banda de detalhe na lista Bandas. A lista de controles disponíveis exibirá os componentes que residem na banda de detalhes.

6 Use as setas à direita da caixa de listagem Controles disponíveis para mover cada um dos componentes DBtext para a caixa de listagem Controles selecionados na seguinte ordem:

Empresa
CustNo
Contato
Telefone
Fax
Cidade
Estado
País

7 Clique no botão OK para fechar a caixa de diálogo.

Configure o relatório para imprimir em arquivo

1 Clique no ícone Selecionar Relatório localizado no canto superior esquerdo da área de trabalho do Designer de Relatórios, onde as régua horizontal e vertical se encontram.

2 Defina a propriedade AllowPrintToFile do Relatório como True no Inspetor de Objetos.

3 Defina DeviceType do relatório como TextFile.

4 Selecione Arquivo | Salvar como no menu Report Designer e salvar as modificações em CustTab.rtm.

Codifique o nome do modelo

1 Feche o Report Designer.

2 Acesse o Editor de Código Delphi.

3 Procure na unidade todas as ocorrências de 'rbCustomerList' e altere-as para 'rbPrintData.'

Codifique o nome do arquivo de texto

1 Acesse o evento OnClick do botão Imprimir e altere o manipulador de eventos para que se pareça com o código abaixo.

2 Selecione o projeto | Compile rbTTProj. Corrija quaisquer problemas de compilação.

3 Selecione Arquivo | Salve no menu principal do Delphi.

Imprimir o relatório para arquivo

1 Execute o projeto.

2 Clique no botão Imprimir. Observe que a opção Imprimir em arquivo está selecionada.

3 Clique em OK para imprimir o arquivo.

4 Feche o formulário e retorne ao ambiente de design Delphi.

5 Selecione Arquivo | Abra no menu principal do Delphi e defina os arquivos do tipo como txt. Abra o arquivo Customer.txt. O arquivo deve conter 55 registros eliminados por tabulação.

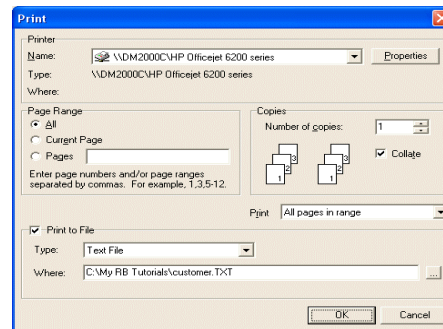
6 Execute o projeto novamente.

7 Selecione o relatório de estilo Vertical e clique no botão Imprimir. Observe que não há nenhuma opção Imprimir para arquivo na caixa de diálogo.

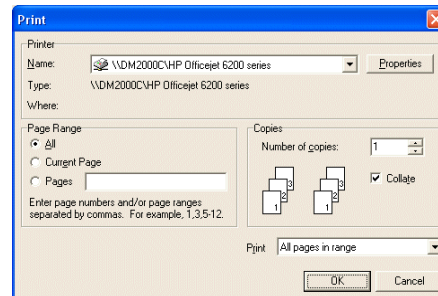
8 Feche a caixa de diálogo e o formulário. A caixa de

diálogo de impressão deve ser semelhante a esta:

Diálogo de impressão de relatório tabular com opção Imprimir para arquivo



Caixa de diálogo de impressão de relatório vertical sem a opção Imprimir para arquivo



Código Manipulador de eventos para o procedimento BuildEmployeeAddress

```

procedimento TfrmPrintToFile.btnPrintClick (Sender: TObject);
começar
  {carregue o arquivo de modelo e imprima na impressora}
  rbPrintData.Template.LoadFromFile;

  E se rbTabular.Checked então
    começar
      rbPrintData.AllowPrintToFile = True; rbPrintData.TextFileName =
        FPathName + 'Customer.txt'; rbPrintData.DeviceType = dtTextFile;

    fim
  senão
    começar
      rbPrintData.AllowPrintToFile = False;
      rbPrintData.TextFileName = '';
      rbPrintData.DeviceType = dtPrinter;

    fim;

  rbPrintData.Print;
fim;
  
```


Impressão de um arquivo de texto

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Imprimir um relatório de dados armazenados em um arquivo de texto

Criar uma nova aplicação

Observação: este tutorial usa o arquivo Customer.txt criado no tutorial anterior. Você pode concluir este tutorial ou usar o arquivo Customer.txt localizado no diretório RBuilder \ Tutoriais.

1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmPrintFromTextFile'.

3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbFrmTxt no diretório My RB Tutoriais.

Nota: É importante nomear o formulário e salvar a unidade do formulário usando os nomes fornecidos acima porque o relatório será usado em tutoriais posteriores.

4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

6 Salve o projeto com o nome rbFTPProj no diretório My RB Tutoriais.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie o componente Text DataPipeline

1 Selecione a guia RBuilder da paleta de componentes Delphi.

2 Coloque um componente TextPipeline no formulário.

3 Configure o componente TextPipeline:

Nome do arquivo	Customer.txt
Tipo de arquivo	ftTab
Nome	plCustomer

Defina os campos de dados para o pipeline de texto

1 Clique duas vezes no componente TextPipeline para exibir o Editor de Campos.

2 Adicione um novo campo clicando no botão Adicionar.

3 Defina as seguintes propriedades para o campo:

FieldName	Empresa
FieldLength	30

Nota: O TextPipeline permite que o ReportBuilder use dados de um arquivo de texto como se estivessem armazenados em um banco de dados. Os comprimentos de campo especificados aqui são comprimentos máximos. Por exemplo, os nomes reais das empresas armazenados no arquivo podem variar em comprimento, mas nenhum pode ter mais de 30 caracteres.

4 Adicione os seguintes campos:

FieldName	FieldLength
CustNo	10
Contato	20
Telefone	15
FAX	15
Cidade	15
Estado	20
País	20

5 Feche o Editor de Campos.

6 Selecione Arquivo | Salve no menu principal do Delphi.

Crie um relatório e conecte-o ao

Dados

1 Adicionar um componente de relatório



para o formulário.

2 Configure o componente Relatório:

DataPipeline	plCustomer
Nome	rbCustomerList

Teste a conexão de dados

1 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

2 Coloque um componente DBText na banda de detalhes e defina-o como AutoSize.

3 Use as duas listas suspensas no canto superior esquerdo do Report Designer para definir as propriedades DataPipeline e DataField:

DataPipeline	plCustomer
Campo de dados	Empresa

Nota: O relatório e o Report Designer se comportam da mesma maneira se o relatório estiver conectado a um DBPipeline, TextPipeline ou qualquer outro tipo de pipeline, incluindo um descendente de pipeline de dados personalizado que você ou outro desenvolvedor criou.

4 Visualize o relatório. O campo Empresa deve ser listado.

5 Selecione a guia Design para retornar à área de trabalho de design.

Carregar o modelo de relatório de estilo vertical

1 Selecione o arquivo | Abra ... no menu principal do Report Designer.

2 Abra o arquivo CustVert.rtm.

Observação: o arquivo CustVert.rtm foi criado no tutorial do Assistente de relatório. O relatório foi conectado a um DBPipeline denominado plCustomer. Agora temos um pipeline de texto denominado plCustomer, cujos campos definimos com os mesmos nomes dos campos usados na tabela Customer.db. Quando o relatório CustVert.rtm é carregado, os dados são reconectados automaticamente porque o pipeline e os nomes dos campos são idênticos.

3 Selecione Arquivo | Salve no menu principal do Delphi.

4 Selecione a guia Visualizar para ver o relatório.

5 Selecione a guia Design para retornar ao modo de design.

Codifique o nome do arquivo de texto

1 Adicione o seguinte código ao evento OnCreate do formulário:

procedimento

```
frmPrintFromTextFile.FormCreate (Sender:
TObject);
```

começar

```
plCustomer.FileName: =
ExtractFilePath (ParamStr (0)) +
'Customer.txt';
```

fim;

Nota: Este código permitirá que você implante o aplicativo sem se preocupar com o diretório em que o arquivo customer.txt está. Este código assume que o arquivo está localizado no mesmo diretório do aplicativo.

Visualize o relatório em tempo de execução

1 Selecione a guia Padrão da paleta de componentes Delphi.

2 Adicione um componente Botão ao formulário.

3 Configure o componente Botão:

Nome btnPreview

Antevisão da legenda

4 Adicione o seguinte código ao manipulador de eventos OnClick do botão:

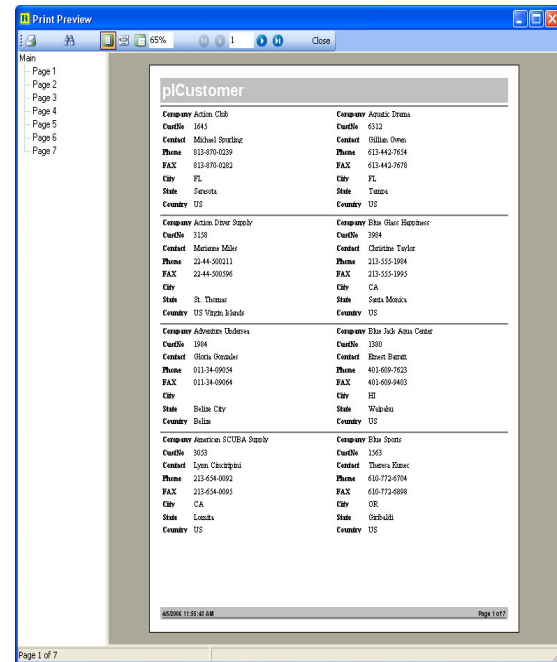
```
rbCustomerList.Print;
```

5 Selecione o projeto | Compile rbFTProj. Corrija quaisquer problemas de compilação.

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Execute o projeto.

8 Clique no botão Visualizar. O relatório deve ser exibido no formulário de visualização de impressão:

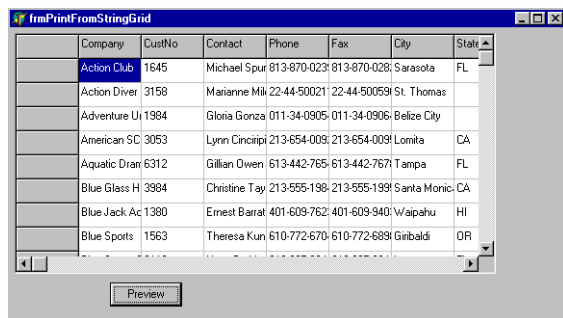


Usando o JITPipeline para imprimir de um StringGrid

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Imprimir um relatório de dados armazenados em uma grade de string ou lista de string



Criar uma nova aplicação

Observação: este tutorial se baseia no tutorial Impressão de um arquivo de texto. Você pode completar o tutorial Imprimir do arquivo de texto ou usar o formulário rbFrmTxt no diretório RBuilder \ Tutorials.

1 Selecione Arquivo | Novo aplicativo do menu Delphi.

Isso criará um novo projeto e um formulário em branco.

2 Feche o formulário em branco e sua unidade associada sem salvar.

3 Selecione Arquivo | Abra no menu principal do Delphi. Abra a unidade rbFrmTxt no diretório Tutorials.

4 Defina o nome do formulário como 'frmPrintFromStringGrid'.

5 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbFrmJIT no diretório My RB Tutorials.

6 Selecione Exibir | Gerente de projeto no menu principal.

7 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Adicionar.

8 Adicione rbFrmJIT ao projeto.

9 Clique com o botão direito sobre o nome do projeto no Project Manager e selecione a opção de menu Salvar.

10 Salve o projeto com o nome rbFJProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Renomear o TextPipeline

1 Renomeie o TextPipeline como plCustomerTxt.

Observação: para este exemplo, o pipeline de texto será usado para carregar os dados do arquivo customer.txt em uma grade de string. Um JITPipeline puxará os dados da grade da string e os fornecerá ao relatório.

Crie um StringGrid

1 Selecione a guia Adicional da paleta de componentes Delphi e adicione um StringGrid ao formulário.

2 Configure o StringGrid:

Deixou	5
Principal	5
Largura	500
Altura	250
Nome	grCustomer

3 Clique duas vezes na propriedade Options no Object Inspector para expandir a lista de opções. Defina goColSizing e goRowSelect como True.

Adicionar código para carregar o StringGrid

1 Adicione a seguinte declaração de procedimento ao final da seção de interface da unidade (logo abaixo do final da declaração de classe do formulário e logo acima da declaração var).

procedimento PipeDataToGrid (aPipeline: TppDataPipeline; aGrid: TStringGrid);

2 Copie a rotina PipeDataToGrid do formulário dm0139 em RBuilder \ Demos \ 1. Diretório de relatórios.

3 Cole o código na parte inferior da seção de implementação da unidade rbFrmJIT.

Nota: este procedimento é escrito de maneira genérica para que possa ser reutilizado conforme necessário. Você passa ao procedimento um pipeline de dados (TextPipeline, JITPipeline, DBPipeline, etc) e uma grade de string. O procedimento carrega todos os dados do pipeline na grade da string. Você pode armazenar rotinas como esta em uma unidade comum ao construir um aplicativo. Você pode então adicionar esta unidade comum à cláusula de usos de outras unidades onde as rotinas de utilidades são necessárias. Desta forma, você pode evitar repetir a mesma rotina em várias unidades.

Observação: este procedimento preenche uma grade de string com dados de pipeline. O pipeline de dados é aberto e todos os dados são percorridos para contar o número total de registros. O tamanho da linha da grade é definido como o número de linhas + 1 porque os nomes dos campos devem ser armazenados na primeira linha (ou seja, linha 0). O tamanho da coluna da grade é igualmente definido para o número de campos + 1 porque a primeira coluna contém a borda da grade. Em seguida, os nomes dos campos são carregados na primeira linha. Por fim, os dados do pipeline são percorridos mais uma vez e carregados nas células da grade.

4 Altere plCustomer para plCustomerTxt no manipulador de eventos OnCreate do formulário:

```
plCustomerTxt.FileName:= ExtractFilePath (ParamStr (0)) + 'Customer.txt'; PipeDataToGrid (plCustomerTxt, grCustomer);
```

Nota: Este código atribui o nome do arquivo de texto (assumindo que o arquivo de texto está no mesmo diretório que o aplicativo) e preenche a grade da string com dados.

5 Selecione o projeto | Compile rbFJProj. Corrija quaisquer problemas de compilação.

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Execute o projeto. Os dados do cliente devem ser exibidos na grade.

Crie o componente JIT DataPipeline

1 Selecione a aba RBuilder do Delphi compo-paleta atual.

2 Adicionar um componente JITPipeline  para o formulário.

Defina a propriedade Name para plCustomer.

3 Defina a propriedade DataPipeline do componente de relatório como plCustomer.

Defina os campos de dados para o JITPipeline

1 Clique duas vezes no componente JITPipeline para exibir o Editor de campo.

2 Clique no botão Adicionar para adicionar um novo campo.

3 Use o Inspetor de objetos para definir o seguinte propriedades para o campo:

FieldName	Empresa
FieldLength	30

Nota: O JITPipeline permite que o ReportBuilder acesse os dados como se estivessem armazenados em um banco de dados. Os comprimentos de campo especificados aqui são comprimentos máximos.

4 Repita as etapas 2 e 3 para adicionar os seguintes campos:

FieldName	FieldLength
CustNo	10
Contato	20
Telefone	15
FAX	15
Cidade	15
Estado	20
País	20

5 Feche o Editor de campo.

6 Selecione o formulário e adicione o seguinte código ao evento OnCreate (abaixo da chamada para 'PipeDataToGrid'):

```
plCustomer.InitialIndex: = 1; plCustomer.RecordCount: =
grCustomer.RowCount-1;
```

7 Selecione Arquivo | Salve no menu principal do Delphi.

Adicione uma função para retornar os valores do campo da grade

1 Copie a declaração da função GetGridFieldValue do formulário 0139 na RB \ Demos \ 1. Diretório de relatórios.

2 Cole-o diretamente acima da declaração PipeDataToGrid.

3 Acesse o formulário 0139 e copie a rotina GetGridFieldValue.

4 Cole o código na parte inferior da seção de implementação da unidade rbFrmJIT.

Nota: Esta rotina retorna o valor do campo de dados para o registro atual. Os nomes dos campos são armazenados na primeira linha da grade e são usados para determinar o número da coluna do nome do campo fornecido. O número da coluna e o índice da linha atual são usados para recuperar o valor do campo atual.

Adicionar manipuladores de eventos JITPipeline para retornar os valores de campo

1 Volte para o formulário e selecione o pipeline de dados JIT plCustomer e, em seguida, clique na guia Eventos no Inspetor de objetos.

2 Crie o seguinte manipulador de eventos para OnGetFieldValue:

```
Resultado := GetGridFieldValue (grCustomer,
    plCustomer.RecordIndex, aFieldName,
    plCustomer.GetFieldDataType (aFieldName));
```

Nota: O manipulador de eventos acima simplesmente chama a rotina GetGridFieldValue e passa o nome do campo solicitado.

3 Selecione o projeto | Compile rbFJProj. Você deve obter um erro relacionado a TppDataType no manipulador de eventos GetGridFieldValue.

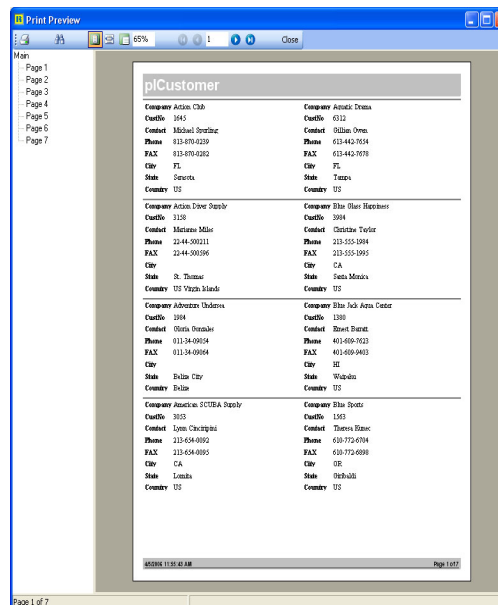
4 Adicione 'ppTypes' à cláusula 'usa' na parte superior da unidade.

5 Compile e selecione Arquivo | Salve no menu principal do Delphi.

6 Selecione o componente de relatório e defina DeviceType como 'Screen'.

7 Execute o projeto.

8 Clique no botão Visualizar.



Usando o componente Rich Text para mala direta / mesclagem

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Gere uma carta modelo com informações de endereço de um banco de dados
- Edite o componente padrão de Rich Text
- Use o recurso de correio / mesclagem

Jennifer Davis
100 Cranberry St.
Wellesley, MA 02181

Dear Jennifer,
Greetings and welcome to the new standard for Delphi Reporting!

Over the past several months we have received many questions about Digital Metaphors and about the history behind ReportBuilder. When we first starting getting into Delphi, we were amazed at what could be accomplished using this slick object-oriented tool. It is a tribute to creators of Delphi that they were able to design something so powerful that it could actually be used to write itself! But then we hit the wall - there were no integrated tools available for building the type of professional quality reports to which we, and more importantly our *clients*, were accustomed. We started by taking the best features from the report designers we had worked with in the past. We then combined object-oriented design principles and our relentless pursuit of quality. The result is a product that has quickly become the report tool of choice among Delphi novices and experts alike.

Jennifer, we would like to invite you to download a demo of our software from www.digital-metaphors.com. Afterwards, drop us a line at info@digital-metaphors.com. We always love to hear from fellow Delphi enthusiasts.

Sincerely,

The Digital Metaphors Team

Criar uma nova aplicação

- 1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.
- 2 Defina o nome do formulário como 'frmMailMerge'.
- 3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbMailM no diretório My RB Tutorials.
- 4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.
- 5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.
- 6 Salve o projeto com o nome rbMMProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie uma tabela, uma fonte de dados e um componente DataPipeline

- 1 Selecione a guia Data Access da paleta de componentes Delphi.
- 2 Adicione um componente Tabela ao formulário.
- 3 Configure o componente Tabela:

DatabaseName DBDEMOS

Nome tblClient

TableName clients.dbf

4 Adicione um componente DataSource ao formulário.

5 Configure o componente DataSource:

DataSet	tblClient
Nome	dsClient

6 Selecione a aba RBuilder do Delphi compo-
paleta atual.

7 Adicionar um componente DBPipeline  para o formulário.

8 Configure o componente DBPipeline:

DataSource	dsClient
Nome	plClient

Crie um relatório e conecte-o ao

Dados

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

DataPipeline	plClient
Nome	rbMailMerge

Chame o Report Designer e defina o
layout da página

1 Clique duas vezes no componente Relatório para exibir
o Designer de Relatórios.

2 Dimensione e mova a janela Report Designer para
que o Object Inspector fique visível.

3 Selecione Arquivo | Configuração da página no menu
principal do Report Designer.

4 Selecione a guia Margens.

5 Defina todas as margens para 1,25.

6 Clique no botão OK para fechar a caixa de diálogo.

Modifique as bandas

1 Selecione Relatório | Cabeçalho do menu principal
do Report Designer. A banda do cabeçalho será
removida do relatório.

2 Selecione o relatório | Opção de rodapé no menu
principal. A faixa de rodapé será removida do
relatório.

3 Clique com o botão direito sobre o espaço em branco da banda
de detalhes e selecione Posição. Defina PrintCount como 1 e Height
como 4. Isso permitirá que apenas uma banda de detalhes seja
impressa por página, criando assim o efeito de uma carta padrão
por página.

4 Coloque um componente RichText na banda de detalhes.

5 Clique com o botão direito do mouse no componente
RichText e selecione a opção de menu Esticar. Isso forçará o
componente RichText a redimensionar com base no tamanho
da letra.

6 Clique com o botão direito sobre o componente RichText e
defina a posição e o tamanho:

Deixou	0
Principal	0
Largura	6
Altura	4

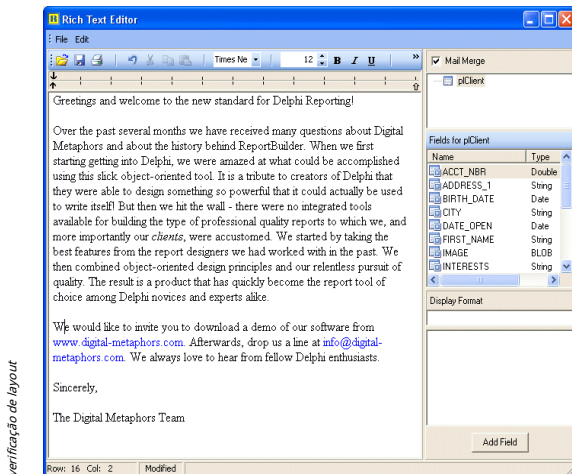
Observação: posicionamos o controle RichText de forma
que preencha toda a banda de detalhes. Isso nos permite
usar as margens do relatório para controlar o
posicionamento da carta, em vez de posicionar o
componente RichText dentro da banda. A altura de 4 é
arbitrária: ela simplesmente nos permitirá ler o

todo o conteúdo da carta durante o projeto. Quando o relatório for impresso, o componente RichText calculará sua altura com base no comprimento da carta.



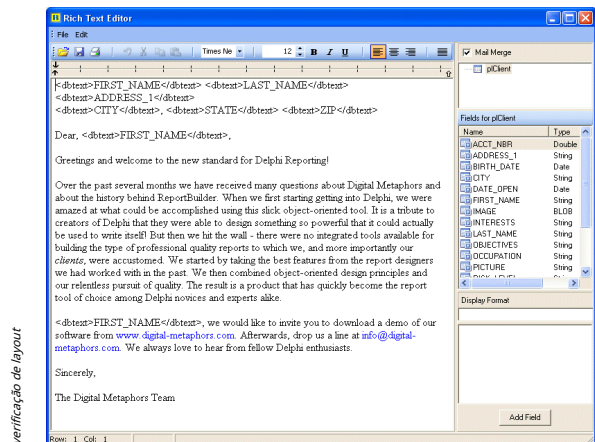
Carregar MMLetter no componente RichText

- 1 Clique com o botão direito do mouse no componente RichText e selecione a opção de menu MailMerge.
- 2 Clique com o botão direito do mouse no componente RichText e acesse a opção de menu Editar. O Editor de Rich Text será exibido.
- 3 Selecione Arquivo | Abra no menu do Editor de Rich Text.
- 4 Localize o arquivo MMLetter.RTF no diretório RBuilder \ Tutorials. Abra este arquivo.



Adicionar campos à carta

- 1 Selecione Editar | Inserir campo. Uma lista de campos da tabela do cliente será exibida.
- 2 Selecione o campo FIRST_NAME e clique em OK.
- 3 Digite um espaço e insira o campo LAST_NAME.
- 4 Pressione Enter para iniciar uma nova linha.
- 5 Insira a linha ADDRESS_1.
- 6 Pressione Enter para iniciar uma nova linha.
- 7 Insira o campo CIDADE. Digite uma vírgula e um espaço.
- 8 Insira o campo ESTADO.
- 9 Digite um espaço e insira o campo ZIP.
- 10 Pressione Enter duas vezes para criar uma linha em branco abaixo dos campos de endereço.
- 11 Digite 'Caro', depois um espaço e, em seguida, insira o campo FIRST_NAME.
- 12 Digite uma vírgula e pressione Enter duas vezes.
- 13 Localize o início do segundo parágrafo (começa com as palavras 'Gostaríamos ...').
- 14 Altere para '<FIRST_NAME>, gostaríamos de ...'



15 Feche o Editor de Rich Text. Clique no botão Sim quando a caixa de diálogo Salvar alterações for exibida.

16 Selecione Arquivo | Salve no menu principal do Delphi.

17 Visualize o relatório. Os dados da tabela de clientes devem ser exibidos.

Visualize o relatório em tempo de execução

1 Feche o Report Designer.

2 Selecione a guia Padrão da paleta de componentes Delphi.

3 Adicione um componente Botão ao formulário.

4 Configure o componente Botão:

Nome btnPreview

Antevisão da legenda

5 Adicione o seguinte código ao manipulador de eventos OnClick do botão:

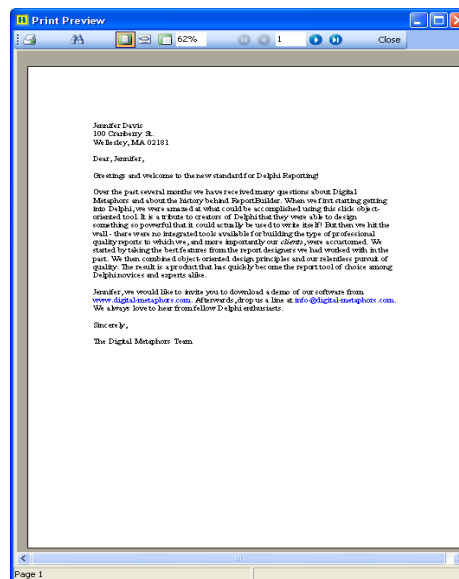
```
rbMailMerge.Print;
```

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Selecione o projeto | Compile rbMMPProj. Corrija quaisquer problemas de compilação.

8 Execute o projeto.

9 Clique no botão Visualizar. O relatório deve ter cinco páginas, com uma letra por página.



Criando uma Crosstab

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Resuma os dados em formato de grade
- Crie um relatório de crosstab

Annual Sales

		Sales Month							
State	Date	1	2	3	4	5	6	7	8
AL	Sales of Automobiles			\$11,269.30		\$19,652.00		\$39,739.70	
	Average Annual Sales			\$11,269.30		\$19,652.00		\$39,739.70	
	Count of Automobiles			1		2		1	
BC	Sales of Automobiles	\$17,517.00			\$7,395.00	\$714.90	\$6,955.00		
	Average Annual Sales	\$17,517.00			\$7,395.00	\$714.90	\$6,955.00		
	Count of Automobiles	1			1	1	2		
CA	Sales of Automobiles		\$22,047.00	\$7,463.90				\$42,675.05	\$571.70
	Average Annual Sales		\$11,423.50	\$3,731.95				\$21,337.53	\$285.85
	Count of Automobiles		2	2				2	1
Conn	Sales of Automobiles	\$6,097.00	\$15,092.00				\$0.00		\$7,009.90
	Average Annual Sales	\$6,097.00	\$15,092.00				\$0.00		\$7,009.90
	Count of Automobiles	1	1				1		1
FL	Sales of Automobiles		\$12,736.00	\$36,629.00	\$124,695.00	\$16,041.00	\$1,103.00	\$35,149.90	\$10,712.00
	Average Annual Sales		\$12,736.00	\$12,210.33	\$41,531.67	\$5,347.00	\$368.67	\$11,716.63	\$3,570.67
	Count of Automobiles		1	3	2	3	2	3	2
GA	Sales of Automobiles			\$0.00		\$1,195.00	\$479.00		\$15,532.00
	Average Annual Sales			\$0.00		\$1,195.00	\$239.50		\$7,766.00
	Count of Automobiles			1		1	2		1
HI	Sales of Automobiles	\$17,677.35	\$0.00		\$1,990.05	\$7,590.00		\$22,174.80	\$13,945.00
	Average Annual Sales	\$17,677.35	\$0.00		\$1,990.05	\$2,530.00		\$7,391.60	\$4,648.33
	Count of Automobiles	2	1		1	3		5	2

9:40:22 AM 10 PM Page 1 of 15

Criar uma nova aplicação

Observação: o componente crosstab está disponível apenas nas edições Professional e Enterprise.

1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmSaleCross'.

3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbCross no diretório My RB Tutorials.

4 Selecione Exibir | Gerente de projeto no menu principal.

5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

6 Salve o projeto com o nome rbCrossProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie um componente Query, DataSource e DataPipeline

1 Selecione a guia BDE da paleta de componentes Delphi.

2 Adicione um componente Consulta ao formulário.

3 Configure o componente Query:

Nome do banco de dados	DBDEMOS
Nome	qryOrder

SQL

SELECT AmountPaid, State,
EXTRACT (MONTH FROM SaleDate) AS
SaleMonth
DO cliente, pedidos
WHERE Orders.CustNo = Customer.CustNo

4 Clique duas vezes na propriedade Active no Object Inspector para defini-la como True.

5 Adicione um componente DataSource ao formulário.

6 Configure o componente DataSource:

DataSet	qryOrder
Nome	dsOrder

7 Selecione a aba RBuilder do Delphi compo-paleta atual.

8 Adicionar um componente DBPipeline  para o formulário.

9 Configure o componente DBPipeline:

Fonte de dados	dsOrder
Nome	plOrder

Crie um relatório

1 Adicionar um componente de relatório  para o formulário.

2 Configure o componente Relatório:

Nome rbOrder

3 Clique duas vezes no componente Relatório para exibir o Designer de Relatórios.

4 Posicione o Report Designer de forma que o Object Inspector também fique visível.

5 Coloque um componente de crosstab barra  (de ferramentas avançada) na banda de detalhes.

6 Use a barra de ferramentas Editar para atribuir a crosstab ao pipeline de dados plOrder.

7 Selecione Arquivo | Configurações da página. Acesse o Tamanho do papel e defina a orientação como Paisagem. Clique OK.

8 Clique com o botão direito sobre a crosstab e selecione Configurar. O Crosstab Designer será exibido.

9 Selecione o campo SaleMonth (na parte inferior da lista) e arraste-o sobre a nova célula da linha. Os indicadores pretos triangulares devem mostrar onde a nova célula será criada.

10 Quando esses indicadores aparecerem à esquerda da nova célula da linha, solte o botão do mouse.

11 Clique OK.

12 Selecione Arquivo | Salve no menu principal do Dephi.

13 Pré-visualizar. Uma página em branco é exibida porque nenhuma dimensão de Valor foi criada para a crosstab.

Projete a Crosstab

1 Retorne à área de trabalho do projeto, clique com o botão direito do mouse na crosstab e selecione Configurar.

2 Arraste o campo Valor pago sobre a nova célula de valor. Quando os indicadores aparecerem, solte o botão do mouse.

verificação do diagrama

		new column
SaleMonth	new row	Sum of AmountPaid 1000
		new value
Grand Total		1000

Nota: O número 1000 representa o formato do valor calculado. O Total geral indica que a última linha da crosstab mostrará o valor total pago em todos os meses.

Adicionar valores adicionais à tabela de referência cruzada

1 Arraste o campo Valor pago sobre a nova célula de valor e solte-o.

2 Selecione o segundo valor de Soma do Valor Pago.

3 Localize a caixa de listagem suspensa na barra de ferramentas.

4 Selecione Média na lista suspensa.

5 Mais uma vez, arraste o campo Valor pago sobre a nova célula de valor e solte-o.

6 Selecione a segunda Soma do Valor Pago (abaixo da média).

7 Selecione Contagem na lista suspensa.

verificação do diagrama

		new column
SaleMonth	new row	Sum of AmountPaid 1000
		Average AmountPaid 1000
		Count of AmountPaid 1000
		new value
Grand Total (Sum of AmountPaid)		1000
Grand Total (Average AmountPaid)		1000
Grand Total (Count of AmountPaid)		1000

8 Clique OK.

9 Selecione Arquivo | Salve no menu principal do Delphi.

10 Pré-visualizar. A crosstab inclui novos valores. Avance para a segunda página. Os totais gerais estão nesta página. O relatório nos informa a soma, média e contagem do valor pago por mês.

Defina o formato dos valores

1 Acesse o Crosstab Designer.

2 Selecione 1000 em Soma do valor pago.

3 Clique com o botão direito e selecione Formato de exibição. Selecione a primeira opção do menu com um cifrão.

4 Selecione 1000 em Valor médio pago.

5 Clique com o botão direito e selecione Formato de exibição. Selecione a primeira opção do menu com um cifrão.

Calcular totais por estado

- 1 Mova a célula SaleMonth sobre a nova célula da coluna e solte.
- 2 Arraste o Estado da lista de campos, coloque-o sobre a célula da nova linha e solte.

verificação do diagrama

		SaleMonth	Grand Total
		new column	
State	new row	Sum of AmountPaid	
		\$1,000.00	\$1,000.00
		Average AmountPaid	
		\$1,000.00	\$1,000.00
		Count of AmountPaid	
		1000	1000
		new value	
		Grand Total (Sum of AmountPaid)	\$1,000.00
		Grand Total (Average AmountPaid)	\$1,000.00
		Grand Total (Count of AmountPaid)	1000

- 3 Clique OK.

- 4 Selecione Arquivo | Salve no menu principal do Delphi.
- 5 Pré-visualizar. Selecione Página inteira na barra de ferramentas de visualização. A crosstab nos mostra a soma, a média e a contagem do valor pago por mês para cada estado. Observe que os meses aparecem na parte superior do relatório, em vez de na lateral. Isso ocorre porque SaleMonth agora é uma dimensão de coluna.

Esquematize a faixa de cabeçalho

- 1 Retorne à área de trabalho de design e coloque uma etiqueta no canto superior esquerdo da faixa de cabeçalho.
- 2 Defina a legenda como Vendas mensais por estado.
- 3 Configure o rótulo:

Nome da fonte	Arial
Tamanho da fonte	12
Estilo de fonte	Negrito
Cor da fonte	Preto

- 4 Coloque uma SystemVariable no canto esquerdo inferior da faixa de rodapé.
- 5 Defina-o como PrintDateTime.
- 6 Coloque outra SystemVariable no canto inferior direito da faixa de rodapé.
- 7 Defina-o como PageSetDesc.
- 8 Alinhe a parte superior das SystemVariables.
- 9 Configure as variáveis de sistema:

Nome da fonte	Arial
Tamanho da fonte	8
Estilo de fonte	Regular
Cor da fonte	Preto

- 10 Selecione Arquivo | Salve no menu principal do Delphi.
- 11 Pré-visualizar. Observe que o número da página é impresso em cada página.

Definir Paginação

- 1 Clique com o botão direito do mouse no componente crosstab e selecione Paginação. Observe que a configuração padrão é Abaixo e na horizontal.
- 2 Selecione transversalmente e depois para baixo.
- 3 Pré-visualizar. As páginas são impressas transversalmente e depois para baixo.

Use legendas repetidas

- 1 Clique com o botão direito sobre o componente crosstab.
- 2 Expanda a opção de menu Estilo. Observe que a configuração padrão é Padrão.
- 3 Selecione Legendas repetidas.
- 4 Visualize e imprima todas as quatro páginas da crosstab. Observe como as legendas repetidas esclarecem o contexto da crosstab.

RAP TUTORIAIS

Campos de concatenação 359

Componentes de codificação de cores

363 Duplexação dinâmica 367

Adicionando Novas Funções ao RAP

371 Estendendo o RAP RTTI 377

Imprimindo uma Descrição dos Critérios de AutoSearch

387 Exibindo Formulários Delphi do RAP 391

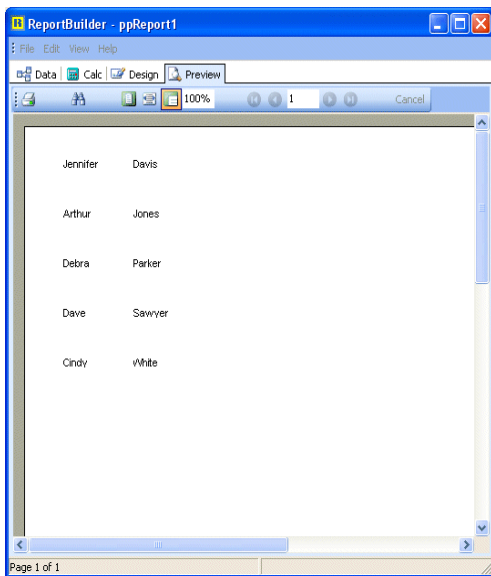
RAP TUTORIAIS

Campos de Concatenação

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Crie um relatório básico em tempo de design
- Crie um relatório que concatene dois campos usando RAP em vez de manipuladores de eventos Delphi.



Crie uma pasta de tutorial

- 1 Acesse o Windows Explorer.
- 2 Crie uma pasta na raiz do seu disco rígido.
- 3 Nomeie a pasta como Meus Tutoriais RAP.

Criar uma nova aplicação

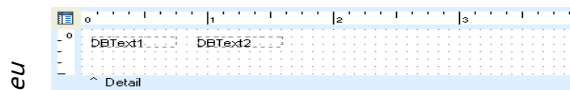
- 1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um novo formulário.
 - 2 Na guia RBuilder na Paleta de componentes, selecione um componente de relatório e solte-o o formulário em branco.
 - 3 Clique com o botão direito do mouse no componente e selecione Report Designer no menu de contexto.
- ### Crie um DataView
- 1 No Report Designer aberto, clique na guia Dados.
 - 2 Selecione Arquivo | Novo no menu. A caixa de diálogo Novos itens aparecerá.
 - 3 Clique duas vezes no Assistente de consulta para começar a criar uma visualização de dados. Na lista de tabelas disponíveis, clique duas vezes na tabela Clientes.
 - 4 Selecione Avançar em todas as páginas a seguir até chegar à última página e clique em Concluir.

Layout do relatório

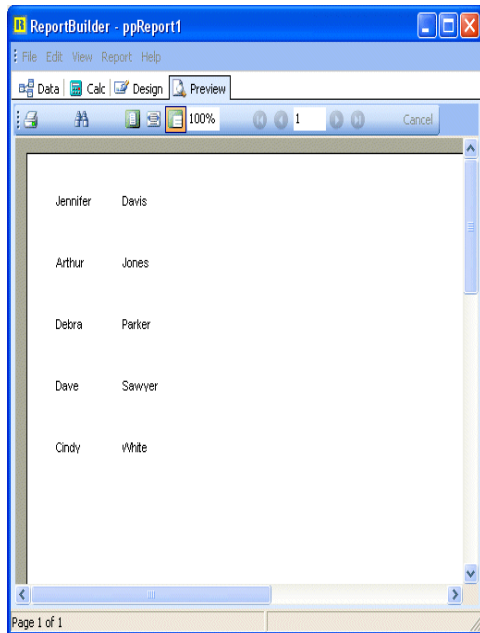
1 Clique na guia Design e desative as bandas de cabeçalho e rodapé do relatório selecionando esses itens em o menu Relatório.

2 Adicione duas bandas de controle  ao detalhe DBText.

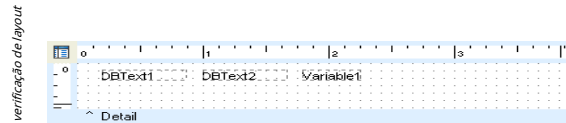
3 Clique em DBText1 e use a lista suspensa Campo de dados para selecionar Nome.



4 Clique em DBText2 e use a lista suspensa Campo de dados para selecionar Sobrenome. Se você visualizar o relatório agora, deverá ver cinco registros de detalhes com nomes e sobrenomes exibidos.



5 Coloque um componente variável  Para a direita dos componentes DBText na faixa de detalhes. A variável exibirá os campos concatenados.



6 Selecione todos os componentes clicando em Shift, e selecione o ícone Alinhar Superior . Alinhar ou

Barra de ferramentas de espaço.

Navegue pelo Calc Workspace

Usaremos o evento OnCalc da Variável para concatenar os dois campos.

1 Clique na guia Calc para exibir a área de trabalho do Calc.

2 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Eventos.

3 Clique em Variable1.

4 Clique com o botão direito do mouse no evento OnCalc e selecione Novo.

Adicione o código de concatenação

A guia Dados da Caixa de ferramentas de código deve estar ativa, se não estiver, selecione-a. Na janela superior da caixa de ferramentas de código, você deve ver o pipeline que adicionamos, plClients. Abaixo disso, você deve ver uma entrada para cada campo no pipeline. Esses itens podem ser arrastados.

Vamos inserir a seguinte linha de código, mas vamos construí-la arrastando e soltando.

1 Clique na entrada Nome na caixa de ferramentas e arraste-o para o Editor de código, logo à direita da linha "Valor: =".

2 Arraste a entrada Sobrenome à direita da linha de código.

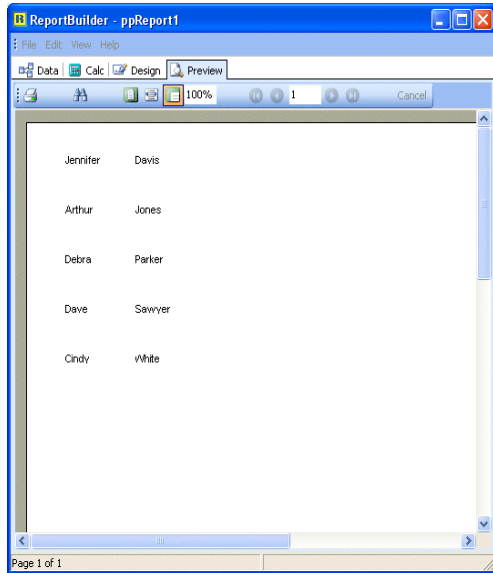
3 Digite os caracteres restantes da linha conforme mostrado abaixo.

Código	Evento OnCalc Variável1
	<pre>procedimento Variable1OnCalc (var Valor: variante); começar Valor: = Clientes ['Nome'] + " + Clientes ['Sobrenome']; fim;</pre>

Compilar e visualizar

1 Para compilar seu código, clique com o botão direito no Editor de Código e selecione Compilar.

2 Para ver os resultados, clique na guia Visualizar.



Parabéns - você concatenou com sucesso os campos do banco de dados usando RAP!

Salve o relatório

1 Clique na guia Design e selecione Salvar como no menu Arquivo.

2 Navegue até a pasta Meus Tutoriais RAP e salve o relatório como RAP Tutor 1.rtm.

Observe que o Código RAP, o DataView e o layout do relatório foram salvos juntos.

Carregue o relatório

1 Selecione Novo Relatório  no menu Arquivo.

2 Volte ao menu Arquivo e abra o relatório que acabou de salvar.

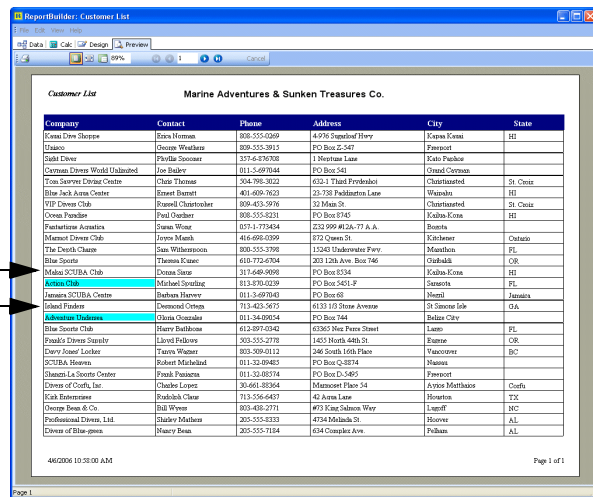
Se você visualizar o relatório, verá que o código RAP está em execução e não foi necessário reconectar nenhum manipulador de eventos.

Componentes de codificação de cores

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Use RAP em tempo de execução
- Adicionar variáveis locais
- Adicione o código do manipulador de eventos para codificar um componente com cores.



Abra o projeto de demonstração

- 1 Seleccione Arquivo | Aberto | RBuilder | Demos |
3. EndUser | 1. Report Explorer | EndUser.dpr

- 2 Habilite o RAP removendo o “X” da frente do \$ DEFINE RAP na interface MyEurpt.pas.

- 3 Compile e execute o projeto de demonstração.

- 4 Seleccione a pasta Clientes.

- 5 Clique duas vezes na Lista de clientes para abri-la.

Ativar DetailBeforePrint

- 1 Clique na guia Calc para ativar a área de trabalho Calc.

- 2 Clique com o botão direito na visualização em árvore do painel esquerdo no Code Explorer e seleccione Eventos.

- 3 Percorra a árvore de relatórios até encontrar a banda de detalhes e seleccione-a.

- 4 Clique com o botão direito do mouse no evento BeforePrint no painel direito e seleccione Novo.

Adicionar Variáveis Locais

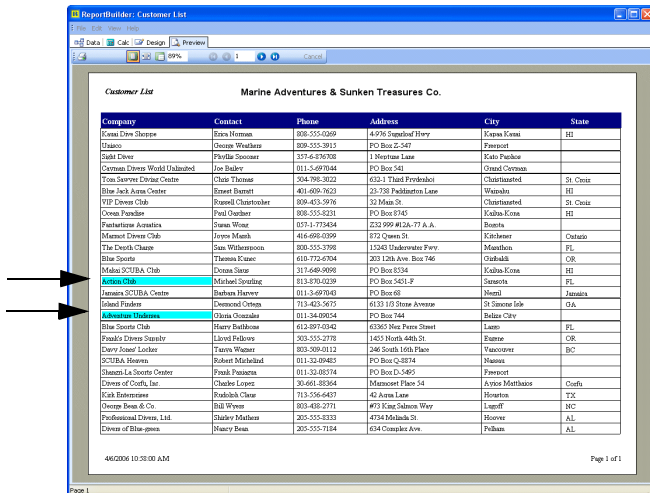
1 Assim como faria no Delphi, coloque o cursor um pouco antes do início e adicione as variáveis locais.

2 Coloque o cursor entre o início e o fim e insira o código para o manipulador de eventos conforme mostrado abaixo.

Código	Detalhe do evento antes da impressão
<pre>procedimento DetailBeforePrint; var lsString: String; liLength: Inteiro; começar lsString := plCustomer ['Company']; liLength: = Comprimento (lsString); Excluir (lsString, 2, liLength - 1); E se lsString = 'A' então ppCustomerListDBText2.Color := clAqua senão ppCustomerListDBText2.Color := clWhite; fim;</pre>	

Visualize e salve

1 Clique na guia Visualizar para ver o relatório. Você deverá ver alguns dos nomes de empresas destacados no aqua.



ReportBuilder - Customer List

Marine Adventures & Sunken Treasures Co.

Company	Contact	Phone	Address	City	State
Kean Dive Shoppe	Bora Roman	808-555-0269	4-976 Sandford Hwy	Kona Kona	HI
Ukalo	Christie Whelan	808-555-3912	PO Box 2-547	Princeton	
Solo Dive	Phyllis Spencer	352-6-270308	1 Maritime Lane	Kala Palooka	
Captain Dives World Unlimited	Joe Badier	011-3-607044	PO Box 543	Grand Cayman	
Tom Service Dives Centre	Chae Thomas	304-795-3022	612-1 Third Freindrick	Charleston	St. Croix
Blue Jack Arms Centre	Robert Emmet	403-609-7623	23-738 Paddleton Lane	Windsor	HI
VIP Dives Club	Barrell Chantabher	808-453-5976	32 Main St	Charleston	St. Croix
Ocean Paradise	Paul Ouelber	808-555-8231	PO Box 1745	Kailua Kona	HI
Paradise Divers	James West	079-1-774844	232 599 #26-77 & A	Buenos	
Marine Dives Club	James Marsh	416-698-0399	872 Queens St	Kitchener	Ontario
The Diver's Choice	Sam Witherspoon	808-555-3798	15243 Underwater Ferry	Mauldin	FL
Blue Sports	Thomas Christ	610-772-6704	203 12th Ave Box 746	Orlando	OR
Marine SCUBA Club	Cynthia Stone	313-648-8095	PO Box 1534	Kailua Kona	HI
Aqua Club	Michael Spaulding	813-870-0239	PO Box 5451-F	Seminole	FL
Jamaica SCUBA Centre	Burton Brown	011-3-607043	PO Box 68	Miami	Jamaica
Edward Fisher	Emmett O'Brien	712-423-5672	613-110 Stone Avenue	St. George Isle	GA
Adventure Divers	Chris Chandler	011-3-649854	PO Box 744	Bahia City	
Blue Sports Club	Henry Bullhouse	612-897-0342	63802 Main Penn Street	Lamo	FL
Frank's Dives Supply	Larry Polk	502-555-2778	1455 North 44th St	Tempe	OR
Cherry Island Lashes	Thane Walker	802-300-0112	246 South 10th Place	Yamoussoukro	BO
SCUBA Heaven	Robert MacLeland	011-32-09483	PO Box Q-3874	Shannon	
Shannon La Source Centre	Frank Pansera	011-32-08374	PO Box D-5485	Princeton	
Divers of Choice, Inc	Charles Lyons	30-661-88344	Marionette Place 34	Ayres Matthews	Chiefs
Kick International	Paulaiah Chase	713-556-6487	42 Anna Lane	Houston	TX
Ocean Dive & Co	Dill Wayne	803-418-2771	#73 King Salmon Way	Lagoff	NC
Professional Divers, Ltd.	Shirley Mullens	202-555-8333	4734 Melinda St	Hoover	AL
Divers of Blue jeans	Henry Ross	202-555-7184	634 Computer Ave	Pulaski	AL

4/6/2006 10:28:00 AM Page 1 of 1

É possível salvar seu código como uma entidade separada de seu relatório, exportando-o. Você também pode importar o código salvo.

1 Clique na guia Calc e selecione o menu Arquivo.

2 Selecione Exportar e dê um nome ao seu código.

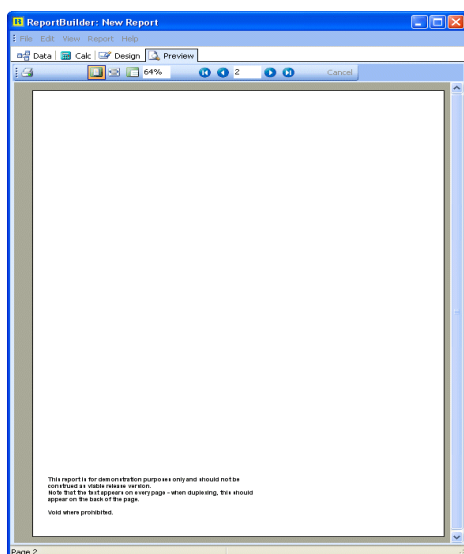
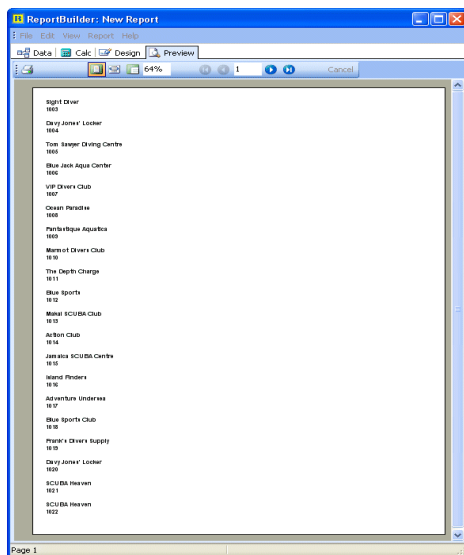
3 Feche o Report Designer. Você deve ver seu código na pasta onde o salvou.

Duplex Dinâmico

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Use a duplexação para imprimir um aviso de isenção de responsabilidade no verso de cada página de um relatório.
- Imprima os dados apenas nas páginas pares e o aviso de isenção de responsabilidade nas páginas ímpares.



Habilitar RAP

- 1 Selecione Arquivo | Aberto | RBuilder | Demos | 3. EndUser | 1. Report Explorer | EndUser.dpr.
- 2 Habilite o RAP removendo o "X" na frente de \$ DEFINE na interface MyEURpt.pas.
- 3 Construa e execute o projeto de demonstração.
- 4 Navegue até a pasta de sua escolha e clique no botão Novo relatório. O Report Designer será aberto automaticamente.

Crie um novo DataView

- 1 Clique na guia Dados.
- 2 Selecione Arquivo | Novo e escolha o Assistente de consulta na caixa de diálogo Novos itens.
- 3 Na primeira página do Assistente, adicione Cliente à lista de Tabelas Seleccionadas.
- 4 Adicione pedidos à lista de tabelas seleccionadas. Isso deve fazer com que a caixa de diálogo Juntar Tabela apareça.
- 5 Clique na linha TaxRate na caixa de listagem Campos unidos e clique no botão Remover.
- 6 Clique em OK para entrar nas mesas.
- 7 Aceite os padrões em todo o resto das páginas e clique em Concluir para criar o DataView.
- 8 Feche a visualização quando ela aparecer.

Criar layout de relatório

- 1 Clique na guia Design para iniciar o layout do relatório.
- 2 Desative as bandas de cabeçalho e rodapé selecionando esses itens no menu Relatório.
- 3 Selecione Relatório | Grupos para exibir a caixa de diálogo Grupos.
- 4 Selecione Customer.CustNo na lista suspensa Grupos e clique em Adicionar para criar o grupo.
- 5 Certifique-se de que a opção Reimprimir cabeçalhos de grupo nas páginas subsequentes esteja marcada e clique em OK para criar o grupo.
- 6 Clique com o botão direito do mouse no divisor de banda rotulado ^ Cabeçalho do grupo [0]: CustNo e selecione Posição no menu de contexto.
- 7 Precisamos fazer com que o Cabeçalho do Grupo ocupe toda a página física, portanto, supondo que Report.Units seja definido como Polegadas e suas margens sejam definidas como 0,25 "cada, defina Altura como 10,5" e clique em OK.

- 8 Coloque um memorando  na banda do cabeçalho do grupo para conter sua isenção de responsabilidade.
- 9 Clique com o botão direito no memorando e selecione Linhas.
- 10 Edite o controle de memorando e adicione algum texto oficial.

Nota: O cabeçalho do grupo e o controle de memo serão impressão no verso de cada página.

- 11 Em seguida, coloque dois controles DBText  no faixa de detalhes e anexe um ao campo Empresa e um ao campo OrderNo.

Nota: O cabeçalho do grupo funcionará para todas as páginas, exceto a última. Como os dados sempre terminam em uma página ímpar, precisaremos providenciar algo para imprimir o aviso de isenção de responsabilidade no verso da última página.

- 12 Posicione o componente DBText da empresa:

Esquerdo 0,0313

Principais 0,0208
- 13 Posicione o componente OrderNo DBText:

Esquerdo 0,0313

Top 0,2083
- 14 Clique com o botão direito nos componentes DBText e selecione AutoSize.
- 15 Selecione o item Resumo no menu Relatório para ativar a banda Resumo.
- 16 Assim como você fez para a faixa do cabeçalho do grupo, defina a altura da faixa do resumo para 10,5 ".
- 17 Copie o controle Memo do cabeçalho do grupo para a faixa de resumo e coloque-o na mesma posição na faixa.
- 18 Clique na guia Calc.
- 19 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Eventos.
- 20 Clique com o botão direito do mouse no evento OnStartPage do relatório na visualização em lista e selecione Novo.

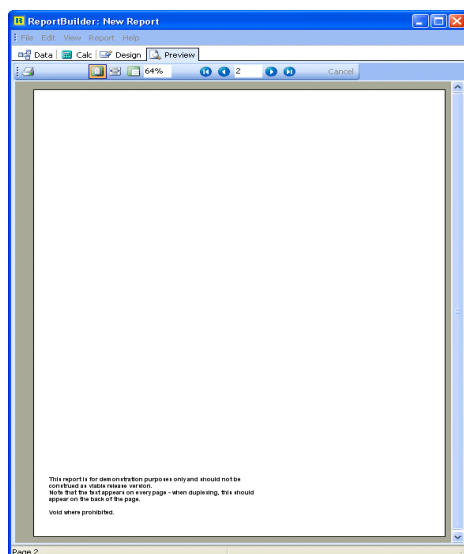
- 21 No Editor de código, insira o código abaixo:

Código	Reportar evento OnStartPage
procedimento ReportOnStartPage; começar GroupHeaderBand1.Visible: = (Report.AbsolutePageNo mod 2 = 0); fim;	

Compilar e visualizar

1 Para compilar seu código, clique com o botão direito no Editor de Código e selecione Compilar.

Para ver os resultados, clique na guia Visualizar



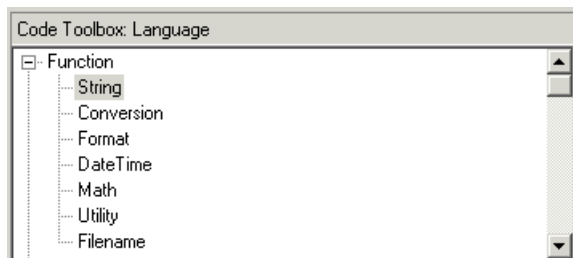
Conforme você percorre o relatório, notará que todas as outras páginas contêm o aviso de isenção. A faixa de resumo deve certificar-se de que a última página do relatório também é o aviso de isenção. Se você imprimir este relatório no modo duplex, a isenção de responsabilidade deve aparecer no verso de cada página.

Adicionando funções ao RAP

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Adicione duas funções e uma categoria à lista de funções na caixa de ferramentas de código.
- Adicione uma função de passagem para recuperar o nome do arquivo do aplicativo, uma função para expor a função ExtractFilename Delphi e uma categoria chamada "Filename".



Criar uma nova unidade

- 1 Abra o Delphi e selecione Fechar tudo no menu de arquivo.
- 2 Selecione Arquivo | Novo | Unidade no menu Delphi.
- 3 Selecione Arquivo | Salvar como | C:\Meus Tutoriais RAP
- 4 Salve o arquivo como myRapFuncs.pas. Esta unidade conterá nossas novas funções de passagem. Esta unidade conterá nossas novas funções de passagem.

Editar cláusula de uso

- 1 Adicione uma cláusula de uso entre as seções de interface e implementação.
- 2 Adicione Forms, raFunc e ppRTTI à cláusula. O código deve ser semelhante a este:

unidade myRapFuncs;

interface

usa

Formulários, raFunc, ppRTTI;

implementação

Observação: raFunc contém TraSystemFunction (de onde descendem todas as funções de passagem) e raRegisterFunction, que é usado para registrar suas funções de passagem com RAP.

Adicionar TraSystemFunction Descendent para Categoria

1 Precisamos de um novo descendente de TraSystemFunction para definir uma nova categoria. Adicione uma seção de tipo sob a cláusula de uso de sua nova unidade. Vamos adicionar um descendente para substituir TraSystemFunction.Category. Adicione a seguinte declaração de classe à seção de tipo:

Código	Declaração de classe TmyFilenameFunction
modelo	<pre>TmyFilenameFunction = aula(TraSystemFunction) público função de classe Categoria: Fragmento; sobrepor; fim;</pre>

2 Na seção de implementação da unidade, adicione o seguinte:

Código	Método de Categoria
implementação	<pre>função de classe TmyFilenameFunction.Category: String; começar resultado = 'Nome do arquivo'; fim;</pre>

Agora, qualquer coisa que herde de TmyFilenameFunction aparecerá na categoria Filename na caixa de ferramentas de código.

Esta classe não será registrada com RAP, pois ela não implementa realmente uma função de passagem.

Adicionar nova classe de função

Na declaração de classe TmyFilenameFunction,
adicione a seguinte declaração de classe:

Código	Declaração de classe TmyApplicationFilenameFunction
	<pre>TmyApplicationFilenameFunction = aula (TmyFilenameFunction) público procedimento ExecuteFunction (aParams: TraParamList); sobrepôr; função de classe GetSignature: String; sobrepôr; função de classe HasParams: Boolean; sobrepôr; fim;</pre>

Implementar ExecuteFunction

Na seção de implementação da unidade, digite o
seguinte:

Código	Método ExecuteFunction
	<pre>procedimento TmyApplicationFilenameFunction.ExecuteFunction (aParams: TraParamList); var IsResult: Fragmento; começar IsResult := Application.Exename; SetParamValue (0, IsResult); fim;</pre>

Implementar GetSignature

Na seção de implementação da unidade, digite o
seguinte:

Código	Método GetSignature
	<pre>função de classe TmyApplicationFilenameFunction.GetSignature: Fragmento; começar Resultado := 'função ApplicationFilename: fragmento'; fim;</pre>

Implementar HasParams

Na seção de implementação da unidade, digite o seguinte:

Código	Método HasParams
	função de classe TmyApplicationFilenameFunction.HasParams: Boolean; começar resultado: = Falso; fim;

Adicionar inicialização

1 Na parte inferior da unidade, antes do fim final, adicione uma seção de inicialização.

2 Nesta seção, digite o seguinte:

Código	Registro da função ApplicationFilename
	inicialização raRegisterFunction ('ApplicationFilename', TmyApplicationFilenameFunction); fim.

Esta linha realmente registra a função de passagem com RAP.

Adicionar myRapFuncs a um projeto

Agora precisamos ver sua nova função em ação. Adicionaremos a unidade a uma das demonstrações do usuário final.

1 Abra o RBuilder | Demos | 3. EndUser |

1. Report Explorer | Projeto EndUser.dpr.

2 Abra o formulário principal, myEURpt.pas.

3 Role para baixo até ver uma série de `{ $ DEFINE }` afirmações. Procure aquele que define RAP e remova o 'x' da frente do `$ DEFINE`.

Agora, precisamos adicionar myRapFuncs ao projeto.

4 Selecione o projeto | Adicionar ao projeto.

5 Navegue até Meus Tutoriais RAP e selecione myRapFuncs.pas.

6 Vá para o final da cláusula Uses e adicione myRapFuncs à cláusula.

7 Compile e execute o projeto.

8 Quando o formulário principal aparecer, clique em Iniciar.

9 Navegue até a pasta Clientes no Report Explorer.

10 Clique duas vezes no relatório Lista de clientes.

Nota: Você deve ver uma guia Calc exibida no Report Designer. Caso contrário, volte e certifique-se de habilitar o RAP e selecione Build All in Delphi.

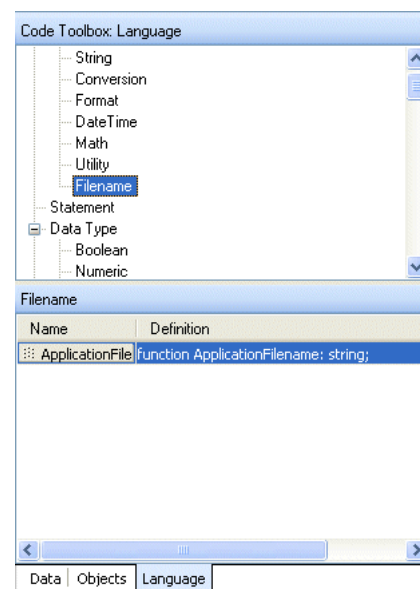
Consulte ApplicationFilename na caixa de ferramentas de código

1 Clique na guia Calc.

2 Clique na guia Idioma na Caixa de Ferramentas de Código e role até o topo da visualização em árvore da Caixa de Ferramentas para localizar o nó Função.

3 Clique no nó Nome do arquivo.

Na lista de funções, você deve ver os frutos de seus esforços.



Agora vamos ver em ação ...

Adicionar ApplicationFilename ao código

1 Clique com o botão direito na visualização em árvore do Report Explorer e selecione Eventos.

2 Selecione o nó Relatório, se ainda não estiver selecionado, e clique no evento BeforePrint listado no painel direito.

3 Clique no Editor de Código para ativar o stub do manipulador de eventos e adicionar uma variável local conforme mostrado abaixo.

4 Adicione o código para o evento ReportBeforePrint.

5 Clique com o botão direito no Editor de código e selecione Compilar.

6 Supondo que o código seja compilado, clique na guia Visualizar.

Parabéns! Você deve ver uma caixa de diálogo de mensagem com o nome do arquivo do aplicativo.

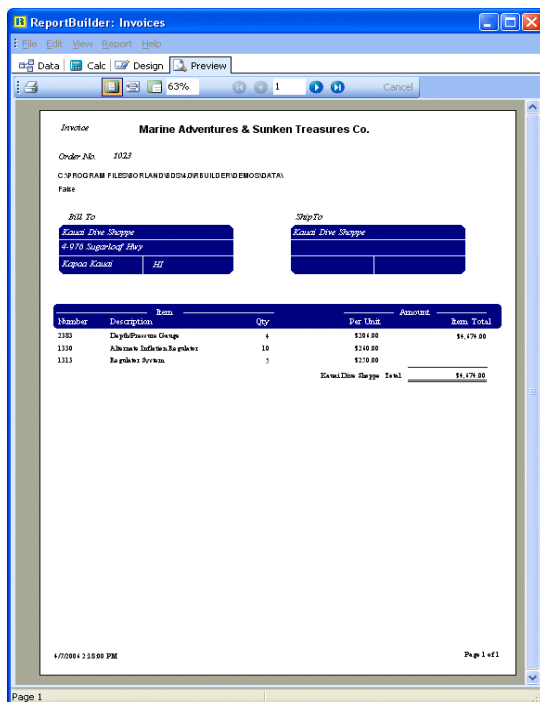
Código	Reportar evento BeforePrint
<pre>var IString: String; começar IString = ApplicationFilename; ShowMessage (IString); fim;</pre>	

Estendendo o RAP RTTI

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Informe o RAP sobre um novo componente.
- Registre uma nova classe para exibir TDataBase no RAP.
- Adicionar suporte para a propriedade pública, Directory e o método público, ValidateName.



Criar uma nova unidade

1 Seleccione Arquivo | Aberto | RBuilder | Demos |

3. EndUser | 1. Report Explorer | EndUser.dpr.

2 Seleccione Arquivo | Novo | Unidade, para criar uma nova unidade.

3 Salve a unidade como myRapClass.pas no C: | Meu diretório de tutoriais RAP. Esta unidade conterá nosso novo descendente TraRTTI.

4 Na nova unidade, adicione uma cláusula uses entre as seções de interface e implementação e adicione Classes, Forms, raFunc, ppRTTI, db, dbTables e ppUtils à cláusula.

unidade myRapClass;

interface

usa

Classes, Forms, raFunc, ppRTTI, db, dbTables, ppUtils

Declare TmyTDataBaseRTTI

Em seguida, devemos encontrar o descendente TraRTTI adequado do qual descender essa classe. TDataBase desce de TCustomConnection que desce de TComponent. Não há nenhum descendente de TraRTTI para TCustomConnection, mas há um para TComponent (TraTComponentRTTI), então usaremos esse.

1 Adicione uma seção de tipo abaixo da cláusula Uses para conter nosso novo componente TmyTDataBaseRTTI.

2 Adicione a seguinte declaração à seção de tipo:

Código	Declaração de classe TmyTDataBaseRTTI
modelo	<pre> TmyTDataBaseRTTI = aula(TraTComponentRTTI) público procedimento de classe GetPropList (aClass: TClass; aPropList: TraPropList); sobrepor; função de classe RefClass: TClass; sobrepor; função de classe GetPropRec (aClass: TClass; const aPropName: String; var aPropRec: TraPropRec): Boolean; sobrepor; função de classe CallMethod (aObject: TObject; const aMethodName: String; aParams: TraParamList; aGet: Boolean): Boolean; sobrepor; função de classe GetParams (const aMethodName: Fragmento); TraParamList; sobrepor; função de classe GetPropValue (aObject: TObject; const aPropName: Fragmento; var aValue): Boolean; sobrepor; função de classe SetPropValue (aObject: TObject; const aPropName: Fragmento; var aValue): Boolean; sobrepor; fim; </pre>

Implementar GetPropList

Implemente o método substituído, GetPropList, com o seguinte código:

Código	Método GetPropList
	<pre> procedimento de classe TmyTDataBaseRTTI.GetPropList (aClass: TClass; aPropList: TraPropList); começar herdado GetPropList (aClass, aPropList); aPropList.AddProp ('Directory'); aPropList.AddMethod ('ValidateName'); fim; </pre>

Implementar RefClass

Adicione o seguinte código à seção de implementação:

Código	Método RefClass
<pre> função de classe TmyTDataBaseRTTI.RefClass: TClass; começar Resultado: = TDataBase; fim; </pre>	

Implementar GetPropRec

Implemente o método substituído, GetPropList, com o seguinte código:

Código	Método GetPropRec
<pre> função de classe TmyTDataBaseRTTI.GetPropRec (aClass: TClass; const aPropName: Fragmento; varaPropRec: TraPropRec): Boolean; começar Resultado: = verdadeiro; E se ppEqual (aPropName, 'Directory') então PropToRec (aPropName, daString, False, aPropRec) mais se ppEqual (aPropName, 'ValidateName') então MethodToRec (aPropName, True, aPropRec) senão Resultado: = GetPropRec herdado (aClass, aPropName, aPropRec); fim; </pre>	

Implementar CallMethod

Implemente o método substituído, CallMethod,
com o seguinte código:

Código	Método CallMethod
	<pre> função de classe TmyTDataBaseRTTI.CallMethod (aObject: TObject; const aMethodName: Fragmento; aParams: TraParamList; aGet: Boolean): Boolean; var IsName: Fragmento; IbResult: Boolean; IDatabase: TDataBase; começar Resultado: = verdadeiro; IDatabase: = TDataBase (aObject); E se ppEqual (aMethodName, 'ValidateName') então começar aParams.GetParamValue (0, IsName); tentar IDatabase.ValidateName (IsName); IbResult: = True; exceto Sobre EDatabaseError Faz IbResult: = False; fim; aParams.SetParamValue (1, IbResult); fim senão Resultado: = herdado CallMethod (aObject, aMethodName, aParams, aGet); fim; </pre>

Implementar GetParams

Implemente o método substituído, GetParams,
com o seguinte código:

Código	Método GetParams
	<pre> função de classe TmyTDataBaseRTTI.GetParams (const aMethodName: Fragmento): TraParamList; começar E se ppEqual (aMethodName, 'ValidateName') então começar Resultado: = TraParamList.Create; Result.AddParam ('Name', daString,nada, ", Verdadeiro falso); Result.AddParam ('Result', daBoolean,nada, ", Falso, falso); fim senão Resultado: = herdado GetParams (aMethodName); fim; </pre>

Implementar GetPropValue

Implemente o método substituído, GetPropValue, com o seguinte código:

Código	Método GetPropValue
	<pre> função de classe TmyTDataBaseRTTI.GetPropValue (aObject: TObject; const aPropName: String; var aValue): Boolean; começar Resultado: = verdadeiro; E se ppEqual (aPropName, 'Directory') então Fragmento(aValue): = TDataBase (aObject) .Directory senão Resultado: = herdado GetPropValue (aObject, aPropName, aValue); fim; </pre>

Implementar SetPropValue

Implemente o método substituído, SetPropValue, com o seguinte código:

Código	Método SetPropValue
	<pre> função de classe TmyTDataBaseRTTI.SetPropValue (aObject: TObject; const aPropName: Fragmento; vara aValue): Boolean; começar Resultado: = verdadeiro; E se ppEqual (aPropName, 'Directory') então TDataBase (aObject) .Directory: = Fragmento(um valor) senão Resultado: = herdado SetPropValue (aObject, aPropName, aValue); fim; </pre>

Registre a aula

Na seção de inicialização da unidade, adicione a seguinte chamada:

```
raRegisterRTTI (TmyTDataBaseRTTI);
```

Na seção de finalização, adicione a seguinte chamada:

```
raUnregisterRTTI (TmyTDataBaseRTTI);
```

Criar função de acesso

Para nossos propósitos, precisamos ser capazes de obter um TDataBase para que possamos testar a nova classe RTTI. Poderíamos simplesmente criar um, mas vamos obter um que já exista. Para fazer isso, vamos criar uma nova função de passagem que nos permitirá chegar a um objeto Delphi por nome. A criação de funções de passagem é abordada em outro tutorial, então vamos acelerar isso.

1 Adicione um novo descendente de TraSystemFunction, TmyDevelopersFunction.

Código	TmyDevelopersFunction
	<pre>modelo TmyDevelopersFunction = aula (TraSystemFunction) público função de classe Categoria: Fragmento; sobrepor; fim; implementação função de classe TmyDevelopersFunction.Category: Fragmento; começar Resultado: = 'Desenvolvedor'; fim;</pre>

2 Adicione um novo TmyDevelopersFunction,
TmyGetDelphiComponentFunction.

Código	TmyGetDelphiComponentFunction
	<pre> TmyGetDelphiComponentFunction = aula (TmyDevelopersFunction) público procedimento ExecuteFunction (aParams: TraParamList); sobrepor; função de classe GetSignature: Fragmento; sobrepor; fim; implementação procedimento TmyGetDelphiComponentFunction.ExecuteFunction (aParams: TraParamList); var IsString: Fragmento; IResult: TComponent; começar GetParamValue (0, IsString); IResult := Application.MainForm.FindComponent (IsString); SetParamValue (1, IResult); fim; função de classe TmyGetDelphiComponentFunction.GetSignature: String; começar Resultado := 'function GetDelphiComponent (const aComponentName: String): TComponent;'; fim; </pre>

3 Para registrar a nova função de passagem,
adicione o seguinte à seção de inicialização

```
raRegisterFunction ('GetDelphiComponent', TmyGetDelphiComponentFunction);
```

Adicionar myRapClass a um projeto

Adicionaremos a unidade a uma das demonstrações do usuário final.

1 Abra o RBuilder | Demos | 3. EndUser |

1. Report Explorer | Projeto EndUser.dpr.

2 Abra o formulário principal, myEURpt.pas.

3 Role para baixo até ver uma série de instruções
{ \$ DEFINE }. Procure aquele que define RAP e
remova o 'x' da frente de \$ DEFINE.

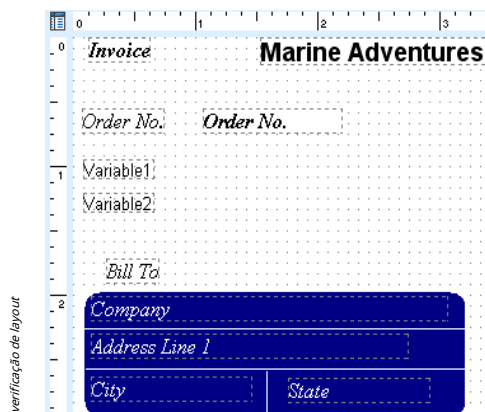
4 Vá para o final da cláusula uses e adicione
myRapClass à cláusula.

5 Compile e execute o projeto.

6 Navegue até a pasta Faturas no Relatório
Explorador.

7 Clique duas vezes no relatório Faturas.

8 Na guia Design, adicione duas variáveis  para o
Faixa de cabeçalho, uma abaixo da outra. Usaremos essas
variáveis para exibir informações de nosso TDataBase.



verificação de layout

0 Invoice Marine Adventures

Order No. Order No.

Variable1

Variable2

Bill To

2 Company

Address Line 1

City State

Escreva o código RAP**1** Clique na guia Calc.

2 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Módulo.

3 Selecione o nó Declarações, clique com o botão direito no item Variáveis e selecione Novo.

4 Insira o seguinte no Editor de Código:

gDataBase: TDataBase;

gbValidName: Boolean;

5 Agora selecione o nó Eventos, clique com o botão direito em OnCreate e selecione Novo.

6 No manipulador de eventos, insira o código para o evento Global OnCreate.

7 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Eventos.

8 Selecione Variable1 na visualização em árvore.

9 Clique com o botão direito do mouse no evento OnCalc e selecione Novo.

10 Insira o seguinte código no manipulador de eventos:

Valor: = gDataBase.Directory;

11 Selecione Variable2 na visualização em árvore.

12 Clique com o botão direito do mouse no evento OnCalc e selecione Novo.

13 Insira o seguinte código no manipulador de eventos:

E se gbValidName então

Valor: = 'Verdadeiro'

senão

Valor: = 'Falso';

Código **Evento Global OnCreate**

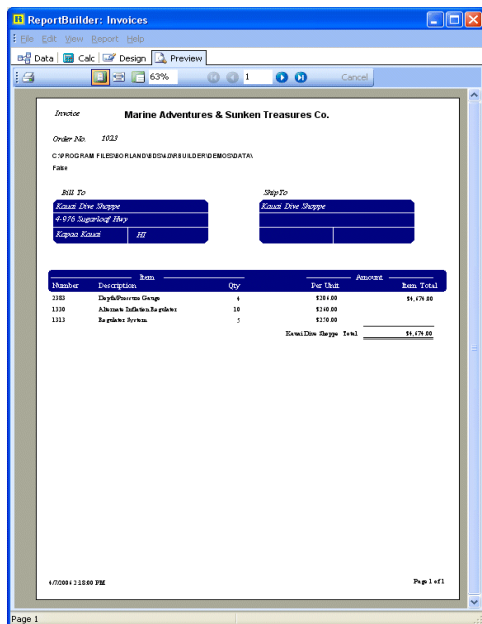
```
gDataBase: = TDataBase (GetDelphiComponent ('euDatabase'));  
E se (gDataBase <> nulo) então  
    começar  
  
        gbValidName: = gDataBase.ValidateName ('DBDEMOS');  
  
    fim;
```

Visualize o relatório

Agora vamos visualizar o relatório. Observe que quando chamamos `ValidateName`, uma exceção será levantada e a execução do programa será interrompida no depurador do Delphi. No entanto, lembre-se de que se você estivesse executando este aplicativo fora do Delphi, a exceção seria tratada e o usuário não veria nenhum problema.

1 Clique na guia Visualizar.

Você deve ver suas duas variáveis, uma indicando um caminho e a outra um valor booleano.



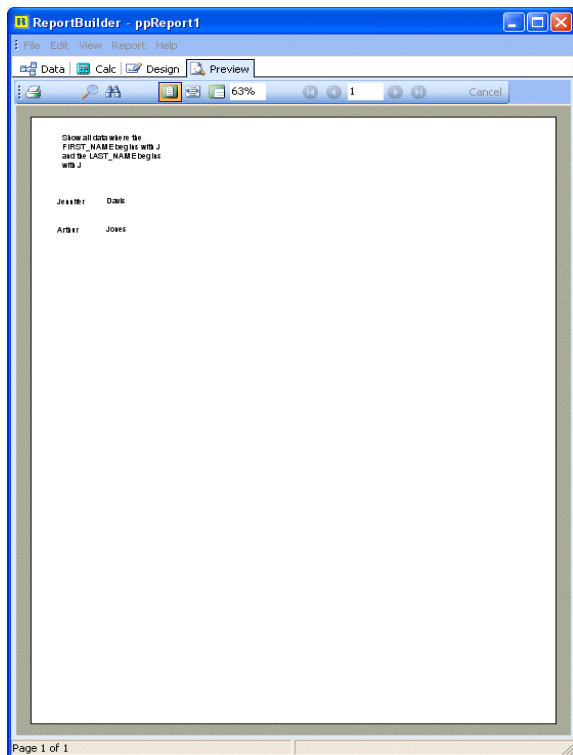
Parabéns! Você adicionou com sucesso sua primeira classe ao RAP.

Imprimindo uma descrição dos critérios de pesquisa automática

Visão geral

Este tutorial irá guiá-lo através do seguinte

- Obtenha acesso às descrições dos campos de AutoSearch via RAP
- Crie um novo relatório e visualização de dados para este tutorial.



Crie um novo relatório

1 Abra o Delphi e selecione Fechar tudo no menu Arquivo.

2 Selecione Arquivo | Novo projeto.

3 Coloque um formulário do componente  no novo Relatório.

Nota: RAP pode ser usado em tempo de design Delphi ou em tempo de execução. Obviamente, seus usuários usarão o Calc Workspace em tempo de execução, mas você pode usá-lo para criar seus relatórios em qualquer ambiente. Este tutorial será feito em tempo de design.

4 Abra o ReportDesigner clicando duas vezes no componente Relatório.

5 Clique na guia Dados.

6 Selecione Arquivo | Novo.

7 Selecione Designer de consulta na caixa de diálogo Novos itens e clique em OK.

8 Na guia Tabelas, clique duas vezes em Clientes.

9 Na guia Campos, marque a caixa de seleção Todos os campos.

A seguir, criaremos os critérios de pesquisa ...

Criar Critérios de Pesquisa

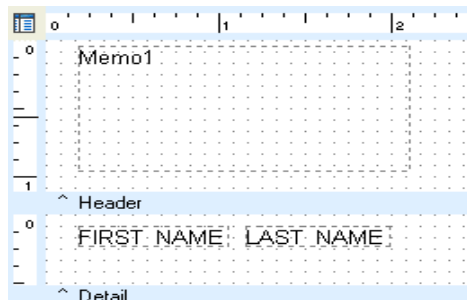
- 1 Na guia Pesquisar, clique duas vezes em Nome para adicionar um critério.
- 2 Para os novos critérios, defina Operator como "Like" e defina Value como "J."
- 3 Marque a caixa de seleção AutoSearch.
- 4 Clique duas vezes em Sobrenome para adicionar um critério.
- 5 Para esses critérios, defina Operator como "Like" e defina Value como "J."
- 6 Marque a caixa de seleção AutoSearch.
- 7 Clique com o botão direito no segundo critério e selecione "Inserir OU".
- 8 Clique no botão OK no Query Designer. Agora você deve ver o DataView.
- 9 Clique no botão Visualizar em DataView. Você deverá ver dois registros exibidos, Jennifer Davis e Arthur Jones.

Layout do relatório

- 1 Clique na guia Design.
- 2 Desative a banda de rodapé selecionando Rodapé no menu Relatório.
- 3 Solte duas bandas de controles  no detalhe DBText.
- 4 Atribua DBText1 ao Nome e DBText2 ao Sobrenome.

- 5 Deixe um Memorando  na banda de cabeçalho.

- 6 Defina o memorando para esticar - isso manterá as descrições dos campos de AutoSearch.



Escreva o código

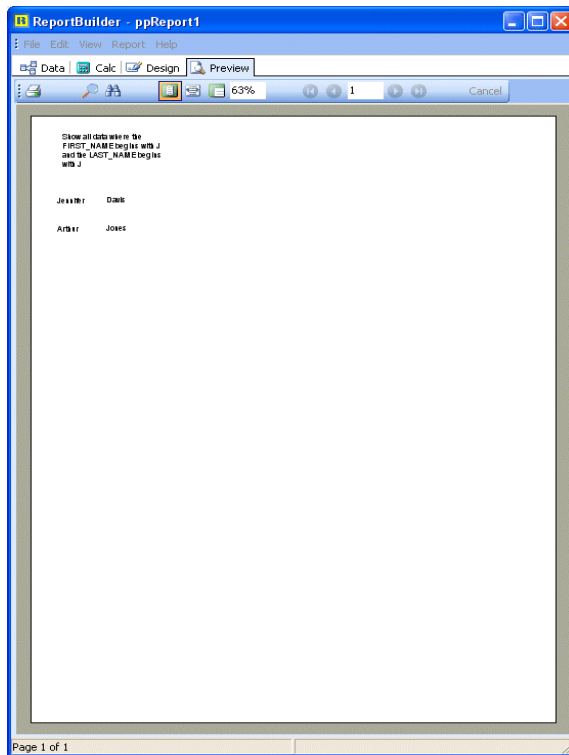
- 1 Clique na guia Calc.
- 2 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Eventos.
- 3 Clique no nó Relatório.
- 4 Clique com o botão direito do mouse no evento OnStartPage na lista e selecione Novo.
- 5 No stub do manipulador de eventos no Editor de código, insira o código a seguir (observe que você pode digitá-lo ou arrastar e soltar o código da caixa de ferramentas de código):

```
Report.GetAutoSearchDescription- Lines
(Memo1.Lines);
```

Compilar e visualizar

1 Para compilar seu código, clique com o botão direito no Editor de Código e selecione Compilar.

2 Para ver os resultados, clique na guia Visualizar.



Parabéns! Você deve ver uma descrição dos campos de AutoSearch no Memo.

Exibindo Formulários Delphi do RAP

Visão geral

A funcionalidade de AutoSearch é bem adequada para pesquisar o usuário final quanto aos critérios de pesquisa; entretanto, às vezes é necessário solicitar ao usuário final informações que não estão relacionadas ao banco de dados. No Delphi isso seria fácil, mas o usuário final, sem o Delphi, não seria capaz de mostrar os formulários personalizados, a menos que você os expusesse por meio de funções de passagem.

Você aprenderá o seguinte com este tutorial:

- O processo de disponibilizar formulários personalizados para o uso final no RAP.
- Crie formulários personalizados e, em seguida, disponibilize-os no RAP.
- Crie um relatório que acesse esses formulários.

Criar formulário de entrada do usuário

1 Abra o Delphi e selecione o arquivo | Feche tudo.

2 Selecione Arquivo | Nova forma.

3 Configure as propriedades do formulário:

Nome	frmMyUserInput
Altura	211
Largura	387
Estilo de borda	bsDialog
Posição	poScreenCenter
Rubrica	'Entrada do usuário'

4 Salve a nova unidade no RBuilder | Demos |

3. EndUser | 1. Diretório do Report Explorer como myUserInputForm.pas.

Adicionar componentes

1 Adicione um componente PageControl ao formulário e defina suas propriedades:

Nome	pgCtlMain
Altura	141
Alinhar	alTop
Estilo	tsFlatButtons

2 Coloque um botão no formulário e defina-o

laços:

Nome	bnOK
Deixou	110
Principal	148
Rubrica	'OK'
Padrão	Verdadeiro
ModalResult	mrOK

3 Coloque outro botão no formulário e defina suas propriedades:

Nome	bnCancelar
Deixou	194
Principal	148
Rubrica	'Cancelar'
ModalResult	mrCancel

4 Com o PageControl existente, adicione duas novas TabSheets. Defina os nomes:

Name = tabStartup
Name = tabMasterRecord

Observação: a propriedade TabVisible para ambas as planilhas será definida como False no evento OnCreate do formulário

5 Em tabStartup adicione um Label e defina suas propriedades:

Nome	lblHeading
Deixou	8
Principal	16
Largura	43
Rubrica	'Cabeçalho:'

6 Coloque uma edição na tabStartup e defina a sua laços:

Nome	ebHeading
Deixou	68
Principal	12
Largura	281
Texto	'Digite um cabeçalho de relatório-'
ing '	

7 Agora, coloque um RadioGroup em tabStartup e defina suas propriedades:

Nome	rgDestination
Deixou	8
Principal	44
Largura	345
Altura	61
Rubrica	'Destino'
Colunas	2
ItemIndex	0
Items.Strings	'Marketing'
	'CEO'
	'Vice-presidentes'
	'Arquivos'

8 Em tabMasterRecord adicione 2 Labels e defina suas propriedades:

Nome	lblCompany
Deixou	12
Principal	8
Largura	47
Rubrica	'Empresa:'

Nome	lblCompanyName
Deixou	68
Principal	8
Largura	82
Rubrica	'Nome da empresa'

9 Em seguida, adicione uma CheckBox a tabMasterRecord e defina suas propriedades:

Nome	cbIncludeAddress
Deixou	32
Principal	40
Largura	181
Rubrica	'Incluir Informação de Endereço no relatório '

Métodos de Formulário de Código

1 Em seguida, adicione um método OnCreate para o formulário e insira o seguinte código:

```
tabStartup.TabVisible: = False;

tabMasterRecord.TabVisible: = False;
```

2 Adicione o seguinte método público ao formulário:

Código	Método InitDialog
	<pre>procedimento TfrmMyUserInput.InitDialog (aStartupPage: Boolean; const aDefaultHeading, aCompanyName: Fragmento); começar E se aStartupPage então pgCtlMain.ActivePage: = tabStartup senão pgCtlMain.ActivePage: = tabMasterRecord; ebHeading.Text: = aDefaultHeading; lblCompanyName.Caption: = aCompanyName; fim;</pre>

Criar uma nova unidade

1 Crie uma nova unidade e salve-a no RBuilder | Demos | 3. EndUser | 1. Diretório do Report Explorer como myRapFuncsForm.pas.

2 Adicione a seguinte cláusula de uso:

Código	Cláusula de Uso
	<pre>usa SysUtils, Windows, Mensagens, Classes, Gráficos, Controles, Formulários, Diálogos, raFunc, ppRTTI, myUserInputForm;</pre>

3 Adicione a seguinte declaração de classe:

Código	Declaração de classe TmyUserInputFormFunction
	<pre>TmyUserInputFormFunction = aula (TraSystemFunction) público função de classe Categoria: Fragmento; sobrepor; fim;</pre>

4 Implemente a função Categoria com este código.

Código	Método de Categoria
<pre>função de classe TmyUserInputFormFunction.Category: Fragmento; começar resultado: = 'Formulários de entrada do usuário'; fim;</pre>	

5 Adicione o seguinte descendente de

TmyUserInputFormFunction:

Código	Declaração de classe TmyShowStartupFormFunction
<pre>TmyShowStartupFormFunction = aula(TmyUserInputFormFunction) procedimento ExecuteFunction (aParams: TraParamList); sobrepor; função de classe GetSignature: Fragmento; sobrepor; fim;</pre>	

6 Implemente TmyShowStartupFormFunction com este código.

Código	Método ExecuteFunction
	<pre> procedimento TmyShowStartupFormFunction.ExecuteFunction (aParams: TraParamList); var IString: Fragmento; IInteger: Integer; IbResult: Boolean; IInputForm: TfrmMyUserInput; começar GetParamValue (0, IString); GetParamValue (1, IInteger); IInputForm := TfrmMyUserInput.Create (aplicativo); tentar IInputForm.InitDialog (True, IString, ""); E se IInputForm.ShowModal = mrOK então começar IString := IInputForm.ebHeading.Text; IInteger := IInputForm.rgDestination.ItemIndex; IbResult := True; fim senão IbResult := False; finalmente IInputForm.Free; fim; SetParamValue (0, IString); SetParamValue (1, IInteger); SetParamValue (2, IbResult); fim; função de classe TmyShowStartupFormFunction.GetSignature: String; começar Resultado := 'função ExecuteStartupForm (var aReportHeading: String; var aDestination- Index: Integer): Boolean;'; fim; </pre>

7 Adicione o seguinte descendente de TmyUserInputFormFunction:

Código	Declaração de classe TmyShowMasterRecordFormFunction
	<pre> TmyShowMasterRecordFormFunction = aula(TmyUserInputFormFunction) procedimento ExecuteFunction (aParams: TraParamList); sobrepor; função de classe GetSignature: Fragmento; sobrepor; fim; </pre>

8 Implemente TmyShowMasterRecordFormFunction com este código.

Código	Método ExecuteFunction
	<pre> procedimento TmyShowMasterRecordFormFunction.ExecuteFunction (aParams: TraParamList); var IString: Fragmento; IBoolean: Boolean; IbResult: Boolean; IInputForm: TfrmMyUserInput; começar GetParamValue (0, IString); GetParamValue (1, IBoolean); IInputForm := TfrmMyUserInput.Create (aplicativo); tentar IInputForm.InitDialog (False, '', IString); IInputForm.cbIncludeAddress.Checked := IBoolean; E se IInputForm.ShowModal = mrOK então começar IBoolean := IInputForm.cbIncludeAddress.Checked; IbResult := True; fim senão IbResult := False; finalmente IInputForm.Free; fim; SetParamValue (1, IBoolean); SetParamValue (2, IbResult); fim; função de classe TmyShowMasterRecordFormFunction.GetSignature: Fragmento; começar resultado := 'function ExecuteMasterRecordForm (const aCompanyName: String; var aDisplay- Endereço: Boolean): Boolean;'; fim; </pre>

9 Por fim, adicione uma seção de inicialização à unidade e adicione as duas linhas a seguir:

Código	Registro de função
	<pre> inicialização raRegisterFunction ('ExecuteStartupForm', TmyShowStartupFormFunction); raRegisterFunction ('ExecuteMasterRecordForm', TmyShowMasterRecordFormFunction); </pre>

Adicionar unidades ao projeto**1** Abra o RBuilder | Demos | 3. EndUser |

1. Report Explorer | Projeto EndUser.dpr.

2 Selecione o projeto | Adicione ao projeto e adicione as duas novas unidades, myUserInputForm.pas e myRapFuncsForm.pas.

3 Compile e execute o projeto.**4** Crie um novo relatório.**Crie um DataView**

1 Depois que o Report Designer estiver aberto, clique na guia Dados.

2 Selecione Arquivo | Novo | Designer de consulta e clique em OK.

3 Na guia Tabelas do Designer de Consultas, clique duas vezes na tabela Cliente.

4 Clique duas vezes na tabela Pedidos para abrir a caixa de diálogo Tabela de junção e remover a junção TaxRate da lista Campos unidos. Em seguida, clique em OK.

5 Na guia Campos, marque a caixa de seleção Todos os campos.

6 Clique em OK para criar o DataView.**Layout do relatório****1** Clique na guia Design

2 No menu Relatório, selecione Grupos ... e crie um grupo com base em plCustomer.CustNo

3 No menu Relatório, selecione os itens Título, Cabeçalho e Rodapé para ativar a banda Título e virar fora das bandas de cabeçalho e rodapé.

4 Na banda do cabeçalho do grupo, adicione um DBText  , uma

Região  e duas variáveis  dentro de região.

5 Atribua o DBText ao campo Company e defina-o como AutoSize.

6 As duas variáveis dentro da região manterão as informações de endereço, portanto, coloque a região e as variáveis abaixo do componente DBText.

7 Desligue a linha ao redor da região.

8 No título do relatório, adicione 3 variáveis .

9 Variable3 será o título do relatório. Centralize-o no topo da banda e ajuste-o para 16pt Arial.

10 Variable4 será nossa saudação. Coloque-o meia polegada abaixo da Variável3 contra a margem esquerda. Defina-o para 10pt Times New Roman.

11 Variable5 é o corpo da mensagem. Coloque-o logo abaixo da Variável 4 contra a margem esquerda. Defina sua largura para 7,5 "e sua altura para 0,5". Definir WordWrap

para verdadeiro.

12 Na banda de detalhes, adicione 3 DBText  controles.

13 Coloque os controles DBText da esquerda para a direita, atribuídos a OrderNo, SaleDate e AmountPaid.

14 Defina a banda de detalhes como altura estática e defina sua altura como 0,25 ".

Declarar Variáveis Globais**1** Clique na guia Calc.

2 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Módulo.

3 Selecione o nó Declarações.

4 Clique com o botão direito no item Variáveis na lista e selecione Novo.

5 No Editor de código, adicione as seguintes declarações:

```
myHeaderString: Fragmento;
myRecipientString: Fragmento;
myDescriptionString: Fragmento;
```

Essas variáveis correspondem às variáveis que você criou na faixa de título.

Implementar Global OnCreate

1 Clique no nó Eventos.

2 Clique com o botão direito no item OnCreate na lista e selecione Novo.

3 No Editor de Código, adicione as duas variáveis locais a seguir ao manipulador de eventos:

var

```
liIndex: Integer;
lbExecuted: Boolean;
```

4 Insira o código na página seguinte para o manipulador de eventos.

Basicamente, neste código estamos executando a versão de inicialização do formulário onde solicitamos um cabeçalho de relatório e um destinatário. liIndex nos informa qual botão de opção foi selecionado no formulário, portanto, formulamos cuidadosamente nossa saudação de acordo.

Código	Evento Global OnCreate
	<pre> começar liIndex: = 0; myHeaderString: = 'Relatório Mensal de Vendas'; E se ExecuteStartupForm (myHeaderString, liIndex) então começar caso liIndex de 0: começar myRecipientString: = 'À atenção de: Equipe de Marketing,'; myDescriptionString: = 'Aqui estão os números mensais de alguém lá pediu. Passe este relatório de mesa em mesa até que alguém se lembre de que o solicitou. '; fim; 1: começar myRecipientString: = 'Minhas humildes desculpas, Vossa Alteza,'; myDescriptionString: = 'Lamento interrompê-lo, mas este humilde pro- grammer traz para você o relatório de vendas mensal que você não vai gostar. '; fim; 2: começar myRecipientString: = 'Caro Brogan, Ethan, Jake, Rebekah e Kathleen,'; myDescriptionString: = 'Visto que em breve estarei me juntando a você como VP, este será meu último relatório mensal. '; fim; 3: começar myRecipientString: = 'CÓPIA DE ARQUIVO'; myDescriptionString: = 'Relatório Mensal de Vendas'; fim; fim; fim senão começar myRecipientString: = 'À atenção de: Equipe de Marketing,'; myDescriptionString: = 'Aqui estão os números mensais de alguém lá pediu. Passe este relatório de mesa em mesa até que alguém se lembre de que o solicitou. '; fim; fim; </pre>

Cabeçalho de grupo BeforePrint Event

1 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Eventos.

2 Role para baixo na visualização em árvore e encontre o nó Cabeçalho do Grupo.

3 Clique nesse nó e, a seguir, clique com o botão direito do mouse no evento BeforePrint na exibição de lista.

4 Implemente este manipulador de eventos com o código conforme mostrado abaixo.

5 Clique com o botão direito na visualização em árvore do Code Explorer e selecione Variáveis.

6 Clique no nó da faixa de título.

Na lista, você verá as três variáveis que residem na faixa de título. Na próxima etapa, iremos atribuir valores para essas variáveis

Código	Cabeçalho de grupo BeforePrint Event
	<pre>var IDisplayAddress: Boolean; começar IDisplayAddress := True; ExecuteMasterRecordForm (plCustomer ['Company'], IDisplayAddress); Region1.Visible := IDisplayAddress; fim;</pre>

7 Para as variáveis, faça as seguintes atribuições:

Código	Eventos OnCalc de Variável de Banda de Título
	Variável 3: Valor: = MyHeaderString; Variável 4: Valor: = MyRecipientString; Variável 5: Valor: = MyDescriptionString;

8 Clique no nó Cabeçalho do grupo na exibição em árvore do Explorador de código.

Faça as seguintes atribuições digitando o código conforme mostrado abaixo.

Código	Eventos OnCalc de Variável de Banda de Título
	Variável 1: Valor: = plCustomer ['Addr1'] + " + plCustomer ['Addr2']; Variável 2: Valor: = plCustomer ['City'] + ',' + plCustomer ['Country'] + " + plCustomer ['Zip'];

Compilar e visualizar

1 Compile o código clicando com o botão direito do mouse no Editor de código e selecionando Compilar.

2 Agora clique na guia Visualizar.

Antes que qualquer coisa apareça, você deve ver o Formulário de inicialização solicitando um título e um destino (destinatário).

3 Insira um valor para um título, selecione um destinatário e clique em ok.

Agora, esta próxima caixa de diálogo será um pouco chata, mas esta é uma demonstração para mostrar um ponto, então tenha paciência.

4 A caixa de diálogo Registro mestre será exibida para cada grupo de registro mestre. Clique na caixa de seleção para indicar se deseja exibir o endereço da empresa.

Isso é tudo. Não se esqueça de que existe uma versão finalizada deste tutorial no projeto de demonstração RAP como o relatório # 35.

UMAPPLICATION TUTORIAIS

Construindo um aplicativo de relatório 405

Construindo um aplicativo de relatório do usuário final 413

Adicionando suporte de dicionário de dados ao aplicativo do usuário final 419

Personalizando o formulário 425 do Report Explorer

Construindo um aplicativo de relatório usando InterBase 427

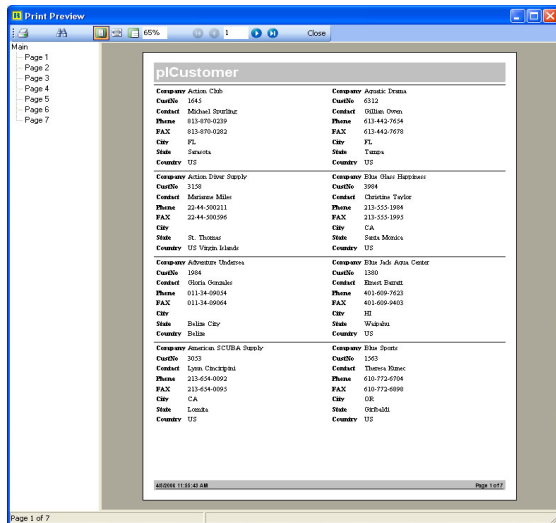
UMAPPLICATION TUTORIAIS

Construindo um aplicativo de relatório

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Construir uma arquitetura para criar formulários de relatório dinamicamente
- Use seu próprio formulário de visualização de impressão personalizado



Criar uma nova aplicação

1 Selecione Arquivo | Novo aplicativo do menu

Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmMain'.

3 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbMain no diretório My RB Tutorials (estabelecido na página 181).

4 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

5 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

6 Salve o projeto com o nome rbMainProj no diretório My RB Tutorials.

Crie uma caixa de listagem e um botão de visualização no formulário principal

1 Selecione a guia Padrão na paleta de componentes Delphi.

2 Adicione uma ListBox ao canto superior esquerdo do formulário Delphi.

3 Configure o ListBox:

Nome	lbrReports
Deixou	8
Principal	8
Largura	280
Altura	300

4 Crie um botão abaixo do ListBox.**5 Configure o botão:**

Nome	btnPreview
Rubrica	Antevisão
Deixou	208
Principal	312

6 Defina o tamanho do formulário:

Altura	368
Largura	303

7 Selecione Arquivo | Salve no menu principal do Delphi.**8 Feche a unidade rbMain.****Adicionar um formulário ao projeto**

1 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

2 Clique com o botão direito sobre rbMainProj no Project Manager e selecione a opção de menu Adicionar

3 Localize a unidade rbViaWiz no diretório Tutoriais. Se você não completou este tutorial, você pode encontrar esta unidade em RBuilder \ Tutoriais. Adicione este formulário ao seu projeto.

4 Clique com o botão direito do mouse no projeto

rbMainProj e acesse o item de menu Opções A caixa de diálogo Opções do projeto será exibida.

5 Selecione a guia Formulários. 'frmViaReportWizard' deve estar listado na lista de formulários de criação automática. Selecione este formulário e clique no botão '>' para movê-lo para a lista Formulários disponíveis. Feche a caixa de diálogo.

Nota: Quando um formulário é adicionado a um projeto Delphi, ele é automaticamente adicionado à lista de criação automática. Os formulários que aparecem nesta lista são criados quando o aplicativo é executado. Estaremos instanciando este formulário manualmente usando Object Pascal e não queremos que o aplicativo o crie para nós.

Criar uma classe de formulário ancestral

1 Selecione Arquivo | Novo ... no menu principal do Delphi.

2 Selecione o ícone Unidade e clique no botão OK. Uma nova unidade intitulada 'Unit1' será criada.

3 Selecione Arquivo | Salvar como no menu principal do Delphi e salvar a unidade com o nome 'rbClass' no diretório My RB Tutoriais.

4 Adicione o seguinte código entre a interface e a seção de implementação da unidade:

usa

Formulários, ppReport;

modelo

TrbReportForm = **aula**(TForm)

protegido

função GetReport: TppReport; **virtual;**
abstrato;

público

propriedade Relatório: TppReport leia
GetReport;

fim;

Nota: Um formulário Delphi é apenas outra classe dentro do VCL. Esta declaração cria um descendente não visual de um TForm. Em seguida, declara uma propriedade de relatório que pode ser usada por qualquer rotina que faça referência à classe. A seguir, modificaremos uma forma visual para nos tornar um descendente de TrbReportForm. Em seguida, implementaremos o procedimento GetReport retornando o objeto de relatório contido no formulário. Esta técnica nos permite trabalhar facilmente com os objetos de relatório em descendentes de TrbReportForm.

5 Selecione Arquivo | Salve no menu principal do Delphi.

Tornar o formulário de relatório do tutorial um descendente de TrbReportForm

1 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

2 Clique duas vezes na unidade rbViaWiz para abrir o formulário.

3 Acesse a unidade rbViaWiz e maximize o Editor de Código.

4 Adicione rbClass à cláusula uses na parte superior da unidade. O código deve ser semelhante a este:

unidade rbViaWiz;

interface

usa

Janelas, Mensagens, SysUtils, Classes, Gráficos, Controles, Formulários, Diálogos, StdCtrls, ppBands, ppCtrls, ppPrnabl, ppClass, ppDB, ppProd, ppReport, ppComm, ppCache, ppDBPipe, ppDBBDE, Db, DBTables, ppVar, rbClass;

Nota: Devemos adicionar a unidade rbClass à cláusula uses para que possamos descer do TrbReportForm.

5 Altere o nome do formulário herdado de TForm para TrbReportForm. A primeira linha da declaração da classe do formulário deve ser semelhante a esta:

```
modelo
  TfrmViaReportWizard =
    aula(TrbReportForm)
```

6 Clique na guia rbClass no Editor de código.

7 Copie a seção protegida da classe (incluindo a palavra-chave protegida) em sua área de transferência. Aqui está o código que você precisa copiar:

```
protegido
  função GetReport: TppReport; virtual;
  abstrato;
```

8 Clique na guia rbViaWiz no Editor de código.

9 Role para baixo até localizar a palavra-chave pública na parte inferior da declaração de classe do formulário.

10 Insira o conteúdo da sua área de transferência acima da palavra-chave pública. A declaração agora deve ter a seguinte aparência:

```
protegido
  função GetReport: TppReport; virtual;
  abstrato;
público
  {Declarações públicas}
fim;
```

11 Substitua as palavras-chave virtuais e abstratas por override para que a declaração tenha a seguinte aparência:

```
protegido
  função GetReport: TppReport;
sobrepôr;
público
  {Declarações públicas}
fim;
```

12 Copie a declaração da função GetReport para a área de transferência e role para baixo até a seção de implementação da unidade.

13 Cole a declaração na seção de implementação, remova a palavra-chave override e adicione o código a seguir para que a função tenha a seguinte aparência:

```
implementação
  {$ R *.DFM}

função TfrmViaReportWizard.GetReport: TppReport;
```

```
começar
  Resultado := rbCustomerList;
fim;
```

Nota: Esta rotina será chamada sempre que a propriedade rbReportForm.Report for referenciada.

14 Selecione Arquivo | Salve no menu principal do Delphi.

15 Feche o Editor de código.

Preencher a caixa de listagem no evento OnCreate do formulário principal

1 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

2 Clique duas vezes em rbMain para abrir este formulário.

3 Selecione a guia Eventos do Inspetor de objetos.

4 Clique duas vezes no evento OnCreate. O shell do manipulador de eventos será gerado e o Editor de Código será exibido.

5 Adicione a seguinte linha de código ao manipulador de eventos:

```
lbrReports.Items.AddObject ('Criação de um relatório  
por meio do assistente de relatório',  
  TObject (TfrmViaReportWizard));
```

Nota: Este código armazena uma descrição do relatório e o tipo de classe para o formulário que contém o relatório na lista. O tipo de classe será usado para instanciar uma instância do formulário.

Codifique o procedimento LaunchReport no formulário principal

1 Adicione a declaração do procedimento LaunchReport à seção privada do formulário principal. O código deve ser semelhante a este:

```
privado
  procedimento LaunchReport;
público
  {Declarações públicas}
fim;
```

2 Adicione as seguintes unidades à cláusula de uso na parte superior da unidade:

```
ppReport, rbViaWiz, rbClass
```

3 Adicione o procedimento LaunchReport à seção de implementação. O código deve ser semelhante ao código abaixo.

4 Selecione Arquivo | Salve no menu principal do Delphi.

Observação: essa rotina recupera o tipo de classe da caixa de listagem, instancia um formulário desse tipo, recupera o relatório do formulário e, em seguida, imprime o relatório. Esta rotina assume duas coisas: que a propriedade Report.ModalPreview seja True e que o tipo de classe no ListBox seja descendente de TrbReportForm. Como você pode ver, a classe abstrata TrbReportForm é muito útil. À medida que avançamos adicionando mais relatórios a este projeto, podemos tornar os formulários descendentes desta classe e adicionar a descrição e o tipo de classe à caixa de listagem, e os relatórios funcionarão 'automaticamente'.

Conecte o procedimento LaunchReport ao ListBox e ao botão Visualizar

1 Clique no ícone Toggle Form / Unit na barra de ferramentas Delphi View (ou pressione a tecla F12). Você deve ver o formulário principal.

2 Selecione o controle ListBox e clique duas vezes no evento OnDblClick no Inspetor de objetos.

3 Adicione uma chamada para LaunchReport a este manipulador de eventos. O código deve ser semelhante a este:

```
procedimento TfrmMain.lbxReportsDlClick (Remetente:
    TObject);
começar
    LaunchReport;
fim;
```

4 Selecione o botão Visualizar no formulário e clique duas vezes no evento OnClick no Inspetor de Objetos.

5 Adicione uma chamada para LaunchReport a este manipulador de eventos.

6 Selecione Arquivo | Salve no menu principal do Delphi.

7 Selecione o projeto | Compile rbMain. Corrija quaisquer problemas de compilação.

8 Feche o Editor de Código e o formulário rbMain.

```
procedimento TfrmMain.LaunchReport;
var
    IFormClass: TFormClass;
    liIndex: Integer;
    IForm: TForm;
    IReport: TppReport;
começar

    liIndex := lbxReports.ItemIndex;

    E se (liIndex = -1) então Saída;

    IFormClass := TFormClass (lbxReports.Items.Objects [liIndex]); IForm :=
    IFormClass.Create (aplicativo);

    IReport := TrbReportForm (IForm) .Report;

    IReport.Print;

    IForm.Free;

fim;
```

9 Execute o aplicativo. A caixa de listagem deve conter um item. Se você clicar duas vezes no item ou selecionar o item e clicar no botão Visualizar, deverá ver o relatório na janela Visualizar impressão.

Crie um formulário de visualização de impressão personalizado

1 Feche o aplicativo e retorne ao ambiente de desenvolvimento Delphi.

2 Selecione Arquivo | Abra no menu principal do Delphi. Localize a unidade ppPrvDlg no diretório RBuilder \ Source. Abra esta unidade.

3 No Object Inspector, altere o nome do formulário de ppPrintPreview para rbPrintPreview. (Pode ser necessário usar a caixa de listagem suspensa na parte superior do Object Inspector para localizar o formulário ppPrintPreview).

4 Salve o formulário no diretório My RB Tutorials com o nome rbPrvDlg.

5 Selecione o painel na parte superior do formulário (denominado pnlPreviewBar).

6 Defina a cor para clAqua.

7 Traga o Editor de Código para a frente e role até a parte inferior da unidade e verifique a seção de inicialização da unidade. O código deve ser semelhante a este:

inicialização

```
ppRegisterForm (TppCustomPreviewer,  
TrbPrintPreview);
```

finalização

```
ppUnRegisterForm (TppCustomPreviewer);
```

fim.

Nota: Esta chamada de registro substitui o formulário integrado ReportBuilder Print Preview por esta versão personalizada. A seção de inicialização dispara quando a unidade é carregada. A unidade é carregada apenas se constar da cláusula de uso de outra unidade ou se estiver incluída no projeto. Por esse motivo, você deve sempre adicionar seu formulário de visualização ao seu projeto ou adicionar o nome da unidade do formulário à cláusula de uso da unidade principal de seu aplicativo.

8 Selecione Arquivo | Salve no menu principal do Delphi.

9 Feche o Editor de código.

Adicione o formulário de visualização de impressão personalizado ao seu projeto

1 Selecione a vista | Opção Project Manager no menu principal do Delphi.

2 Clique com o botão direito sobre rbMainProj e selecione a opção de menu Adicionar

3 Localize rbPrvDlg.pas no diretório My RB Tutorials e adicione-o ao projeto.

4 Clique com o botão direito sobre rbMainProj e selecione o item de menu Opções

5 Selecione a guia Formulários e mova o formulário rbPrintPreview da lista Criação automática para a lista Formulários disponíveis.

6 Selecione o projeto | Compile rbMainProj. Corrija quaisquer problemas de compilação.

7 Selecione Arquivo | Salve no menu principal do Delphi.

8 Execute o aplicativo. Ao visualizar o relatório, você deve ver o formulário personalizado em vez do formulário padrão de visualização de impressão do ReportBuilder.

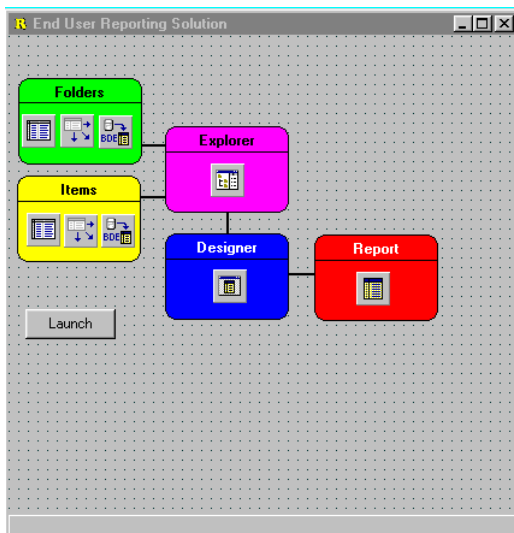
Nota: Você pode adicionar todos os relatórios do tutorial a este projeto da mesma forma que adicionamos este. Para ver uma versão deste aplicativo que inclui todos os relatórios do tutorial, abra o projeto rbMainProj no diretório RBuilder \ Tutorials.

Criação de um aplicativo de relatório de usuário final

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Combine os componentes do ReportBuilder em uma solução de relatório de usuário final
- Configurar objetos de acesso a dados Delphi para uso com o Report Explorer
- Use os componentes Report Designer e Report Explorer



Adicionar tabelas ReportExplorer ao banco de dados

O ReportExplorer fornece uma interface fácil de usar que permite ao usuário organizar relatórios em pastas hierárquicas. Nos bastidores, o ReportExplorer utiliza duas tabelas de banco de dados: rbFolder e rbItem.

Para este exemplo, estamos usando tabelas Paradox. No entanto, o ReportExplorer pode funcionar com qualquer produto de banco de dados. Consulte RBuilder \ Demos \ Enduser Databases para obter exemplos adicionais.

Localize a pasta RBuilder \ Demos \ Data \ EmptyDBs. Copie as tabelas rbItem e rbFolder para a pasta que contém os dados de demonstração do Delphi. Em uma instalação padrão, será algo como C: \ Arquivos de programas \ Arquivos comuns \ Borland Shared \ Data. Certifique-se de copiar rbFolder * e rbItem *, pois há vários arquivos necessários para suportar cada tabela.

Criar uma nova aplicação

- 1 Selecione Arquivo | Novo aplicativo do menu Delphi. Isso criará um novo projeto e um formulário em branco.
- 2 Defina o nome do formulário como 'myEndUserSolution'.
- 3 Defina a legenda do formulário como Solução de relatórios do usuário final.
- 4 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbEndUsr no diretório My RB Tutorials (estabelecido na página 181).
- 5 Selecione Arquivo | Salvar projeto como.
- 6 Salve o projeto com o nome rbEUProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie componentes de acesso a dados para a tabela de pastas

- 1 Selecione a guia BDE da paleta de componentes Delphi.
- 2 Adicione um componente Tabela ao formulário.
- 3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblFolder
Nome da tabela	rbFolder.db
- 4 Adicione um componente DataSource ao formulário.

5 Configure o componente DataSource:

DataSet	tblFolder
Nome	dsFolder

- 6 Selecione a aba RBuilder do Delphi compo-
paleta atual.

- 7 Adicionar um componente DBPipeline  para o formulário.

8 Configure o componente DBPipeline:

Fonte de dados	dsFolder
Nome	plFolder
Visível	Falso

Nota: A propriedade Visible de um DataPipeline controla se o pipeline de dados pode ser selecionado no Report Designer e atribuído a um relatório ou banda de detalhes. Como todos os pipelines criados neste tutorial serão usados para armazenar informações de suporte, nenhum deve estar visível no Report Designer.

Crie componentes de acesso a dados para a tabela de itens

1 Adicione um componente Tabela ao formulário.

2 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblItem
Nome da tabela	rbItem.db

3 Adicione um componente DataSource ao formulário.

4 Configure o componente DataSource:

DataSet	tblItem
Nome	dsItem

5 Selecione a aba RBuilder do Delphi compo-paleta atual.

6 Adicionar um componente DBPipeline  para o formulário.

7 Configure o componente DBPipeline:

DataSource	dsItem
Nome	plItem
Visível	Falso

Crie os componentes ReportBuilder

1 Adicionar um componente de relatório  para o formulário.

2 Adicionar um componente Designer  para o formulário.

3 Defina a propriedade Report do Designer como ppReport1.

4 Adicione um formulário de componente  para o ReportExplorer.

5 Defina a propriedade Designer do ReportExplorer como ppDesigner1.

6 Configure o componente ReportExplorer:

FolderPipeline	plFolder
ItemPipeline	plItem

Configure o Componente Designer

1 Selecione o componente Designer.

2 Expanda a propriedade DataSettings no Inspetor de objetos.

3 Defina a propriedade DatabaseName como DBDEMOS.

Nota: esta propriedade especifica o banco de dados que será consultado.

4 Defina o SQLType como sqBDELocal.

Observação: como estamos usando o SQL local por meio do BDE, precisamos definir o tipo de SQL. Se você não estiver usando o SQL local para acessar as tabelas Paradox e dBase, você deve usar um SQLType de sqSQL1, uma vez que o formato SQL local não é compatível com a maioria dos outros bancos de dados.

5 Adicione daDatMan, daDBBDE à cláusula uses da unidade do formulário.

6 Selecione Arquivo | Salve no menu principal do Delphi.

Nota: A unidade daDatMan contém uma classe de gerenciador de dados que disponibiliza o espaço de trabalho de dados no Report Designer. A unidade daDBBDE associa uma classe de visualização de dados baseada em BDE com o Query Wizard e o Query Designer. Portanto, essas ferramentas criarão instâncias dessa visualização de dados quando chamadas. Este design permite que outras classes de visualização de dados de consulta sejam criadas e associadas a essas ferramentas.

Compilar e executar o aplicativo

1 Selecione a guia Win32 da paleta de componentes Delphi.

2 Adicione um StatusBar ao formulário.

3 Configure o StatusBar:

Nome	stbStatusBar
SimplePanel	Verdadeiro

4 Adicione um componente Botão ao formulário.

5 Configure o botão:

Nome	btnLaunch
Rubrica	Lançar

6 Adicione o seguinte código ao evento OnClick do botão:

```
se não(ppReportExplorer1.Execute) então
  stbStatusBar.SimpleText: =
    ppReportExplorer1.ErrorMessage;
```

Observação: esse código chama o método Execute para invocar o Report Explorer. Se não configuramos uma das propriedades do Report Explorer corretamente, o formulário não será exibido, False será retornado e o motivo da falha será exibido em nossa barra de status.

7 Selecione o projeto | Compile rbEUProj. Corrija quaisquer problemas de compilação.

8 Selecione Arquivo | Salve no menu principal do Delphi.

9 Execute o projeto.

10 Clique no botão Iniciar. O Report Explorer deve ser exibido. Se o Report Explorer não for exibido, uma explicação será exibida no painel na parte inferior da janela.

Criar novas pastas

1 Selecione Arquivo | Novo | Pasta no menu principal do Report Explorer.

2 Digite Marketing no nome da pasta e pressione a tecla Enter.

3 Selecione a pasta principal (intitulada Todas as pastas).

4 Clique com o botão direito do mouse no espaço em branco da Árvore de pastas e selecione a opção Nova pasta no menu pop-up.

5 Digite Contabilidade no nome da pasta e pressione a tecla Enter.

6 Clique com o botão direito do mouse no espaço em branco da Árvore de pastas e selecione a opção Nova pasta. Uma pasta deve ser criada na pasta Contabilidade.

7 Digite 2006 nesta pasta e pressione a tecla Enter.

8 Clique com o botão direito do mouse na pasta 1998 e selecione a opção de menu Novo Relatório.

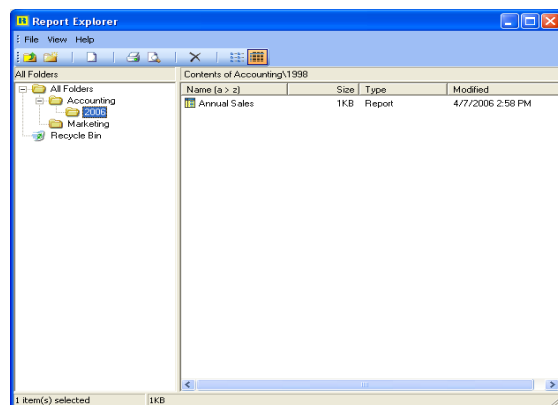
9 Selecione o arquivo | Salvar como no menu principal do Report Designer e salve o relatório com o nome Vendas anuais.

10 Feche o Report Designer.

11 O relatório 'Vendas anuais' deve ser exibido na pasta de 1998.

Nota: Você pode criar e configurar relatórios a partir deste ponto, assim como faria no Delphi. E como você pode ver, o Report Explorer é bastante fácil de usar devido à sua semelhança com o Windows Explorer.

O aplicativo em execução deve ter a seguinte aparência:

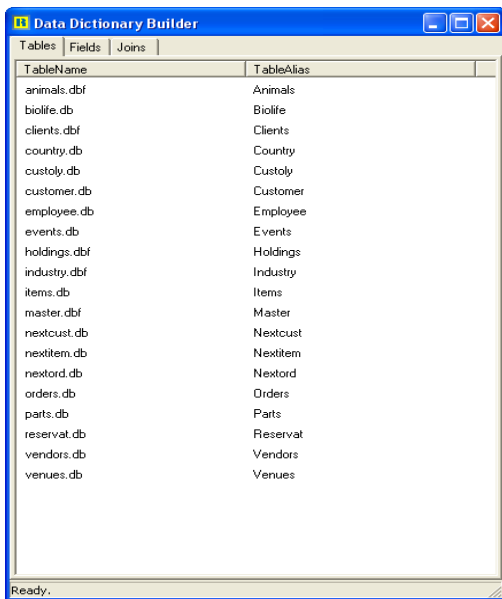


Adicionando suporte de dicionário de dados ao aplicativo do usuário final

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configurar objetos de acesso a dados Delphi para uso com o Dicionário de Dados
- Configure o componente Dicionário de Dados
- Use o Query Wizard dentro do Report Designer



Adicionar tabelas DataDictionary ao banco de dados

O DataDictionary permite que você forneça uma visão personalizada do banco de dados para seus usuários finais. Você pode decidir quais tabelas e campos o usuário final pode ver; definir aliases de tabela e campo; e predefinir relações de junção. Nos bastidores, o DataDictionary utiliza três tabelas de banco de dados: rbTable e rbField e rbJoin.

Para este exemplo, estamos usando tabelas Paradox. No entanto, o DataDictionary pode funcionar com qualquer produto de banco de dados. Consulte RBuilder \ Demos \ Enduser Databases para obter exemplos adicionais.

Localize a pasta RBuilder \ Demos \ Data \ EmptyDBs. Copie as tabelas rbTable, rbField e rbJoin para a pasta que contém os dados de demonstração do Delphi. Em uma instalação padrão, será algo como C: \ Arquivos de programas \ Arquivos comuns \ Borland Shared \ Data. Certifique-se de copiar rbTable *, rbField * e rbJoin *, pois há vários arquivos necessários para suportar cada tabela.

Abra o aplicativo do usuário final

1 Selecione Arquivo | Abra o Projeto a partir do menu Delphi.

2 Abra o projeto rbEUProj do tutorial anterior. Se você não concluiu este tutorial, pode criar um novo projeto e adicionar o formulário rbEndUsr do diretório RBuilder \ Tutoriais.

3 Abra o formulário rbEndUsr e salve-o com o nome 'rbEUDD' no diretório My RB Tutoriais (estabelecido na página 181).

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

4 Selecione Arquivo | Salve o projeto como no menu Delphi e salve o projeto com o nome rbEUDDProj no diretório My RB Tutoriais.

Crie componentes de acesso a dados para as tabelas

1 Selecione a guia Data Access da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblTable
Nome da tabela	rbTable.db

4 Adicione um componente DataSource ao formulário.

5 Configure o componente DataSource:

DataSet tblTable
Nome dsTable

6 Selecione a aba RBuilder do Delphi compo-
paleta atual.

7 Adicionar um componente DBPipeline  para o formulário.

8 Configure o componente DBPipeline:

Fonte de dados	dsTable
Nome	plTable
Visível	Falso

Crie componentes de acesso a dados para a tabela de campo

1 Adicione um componente Tabela ao formulário.

2 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblField
Nome da tabela	rbField.db

3 Adicione um componente DataSource ao formulário.

4 Configure o componente DataSource:

DataSet tblField
Nome dsField

5 Selecione a guia RBuilder da paleta de componentes Delphi.

6 Adicione um componente DBPipeline ao formulário.

7 Configure o componente DBPipeline:

Fonte de dados	dsField
Nome	plField
Visível	Falso

Crie componentes de acesso a dados para a tabela de junção

1 Adicione um componente Tabela ao formulário.

2 Configure o componente Tabela:

Nome do banco de dados	DBDEMOS
Nome	tblJoin
Nome da tabela	rbJoin.db

3 Adicione um componente DataSource ao formulário.

4 Configure o componente DataSource:

DataSet	tblJoin
Nome	dsJoin

5 Selecione a aba RBuilder do Delphi compo-
paleta atual.

6 Adicionar um DBPipeline  componente do formulário.

7 Configure o componente DBPipeline:

Fonte de dados	dsJoin
Nome	plJoin
Visível	Falso

Crie o componente de dicionário de dados

1 Adicione um formulário  componente para o
DataDictionary.

2 Configure o componente DataDictionary:

AutoJoin	True
BuilderSettings.DatabaseName	DBDEMOS
FieldPipeline	plField
JoinPipeline	plJoin
TablePipeline	plTable

3 Selecione Arquivo | Salve no menu principal do Delphi.

Preencher as tabelas

1 Clique duas vezes no componente DataDictionary para iniciar o Data Dictionary Builder.

2 Clique com o botão direito no espaço em branco da guia Tabelas e selecione Gerar. As tabelas serão geradas.

3 Clique na guia Campos e gere as
entradas.

4 Clique na guia Associações. Clique com o botão direito e selecione a
opção de menu Adicionar. O painel de junção aparecerá.

5 Selecione Customer.db na lista da tabela à
esquerda.

6 Selecione Left Outer na lista suspensa Join Type.

7 Selecione Pedidos na lista da tabela à direita. Os
campos de junção sugeridos aparecerão na seção
Campos Juntos do painel.

8 Clique duas vezes em TaxRate para removê-lo da
seleção de junção.

9 Clique em Salvar.

10 Clique com o botão direito e selecione Adicionar. O painel de junção
aparecerá.

11 Selecione Orders.db na lista da tabela à esquerda.

12 Selecione Left Outer na lista suspensa Join
Type.

13 Selecione Funcionário na lista com tabela à
direita. Os campos de junção sugeridos aparecerão
na seção Campos Juntos do painel.

14 Clique em Salvar.

15 Feche o Data Dictionary Builder.

Configure o Componente Designer

1 Selecione o formulário do componente  no Designer.

2 Expanda a propriedade DataSettings no Inspetor de objetos.

3 Defina a propriedade DatabaseName como DBDEMOS.

Nota: O DatabaseName indica o banco de dados que contém os dados a serem recuperados pelos relatórios. Para manter este tutorial simples, usamos o mesmo alias de banco de dados para o dicionário de dados e as tabelas de dados do relatório. Isso não é um requerimento. Você pode armazenar o dicionário de dados em tabelas locais e construir consultas em um banco de dados remoto.

4 Defina a propriedade DataDictionary como ppDataDictionary1.

5 Defina a propriedade UseDataDictionary como True.

Nota: Definir essas duas propriedades permite que o Designer exiba os valores TableAlias e FieldAlias das tabelas do dicionário de dados em vez dos nomes dos campos brutos. Além disso, os valores dos campos Seleccionável, Pesquisável e Classificável serão usados para determinar se um determinado campo pode aparecer nas páginas Selecionar Campo, Critérios de Pesquisa e Definir Pedido do Assistente de Consulta e Designer de Consulta.

6 Selecione Arquivo | Salve no menu principal do Delphi.

Compilar e executar o aplicativo

1 Selecione o projeto | Compile rBEUDDProj. Corrija quaisquer problemas de compilação.

2 Selecione Arquivo | Salve no menu principal do Delphi.

3 Execute o projeto.

Crie uma consulta simples

1 Crie um novo relatório.

2 Selecione a guia Dados do Report Designer.

3 Selecione Arquivo | Novo.

4 Selecione o Assistente de consulta.

5 Selecione a tabela Cliente na lista de tabelas disponíveis. Role para baixo e selecione a tabela Pedidos.

6 Clique no botão Avançar.

7 Clique na opção Escolher Campos.

8 Selecione Empresa, Contato, Estado, N° do pedido e Valor pago e mova-os para a lista Campos selecionados.

9 Pule as próximas quatro páginas clicando em Avançar até chegar à última página.

10 Clique no botão Concluir. Os dados serão exibidos.

11 Feche a caixa de diálogo Visualização de dados.

12 O Dataview aparecerá na área de trabalho da guia Dados.

13 Clique na guia Design.

Crie um relatório simples

1 Selecione Arquivo | Novo no menu principal do Report Designer.

2 Selecione o Assistente de Relatório.

3 Selecione todos os campos disponíveis.

4 Selecione as opções restantes com base em suas preferências.

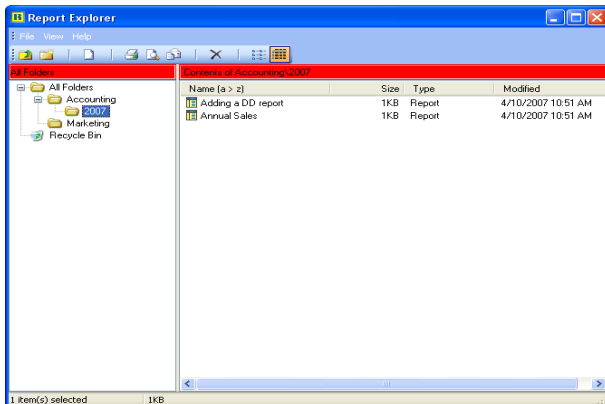
5 Depois de clicar no botão Concluir e retornar à guia Design, salve seu relatório com o nome 'Pedidos'.

Personalizando o formulário do Report Explorer

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Substitua o formulário Report Explorer por um formulário próprio
- Use o procedimento ppRegisterForm



Abra o aplicativo do usuário final

1 Selecione Arquivo | Abra o Projeto a partir do menu Delphi.

2 Abra o projeto rbEUProj do tutorial “Construindo um aplicativo de usuário final”. Se você não concluiu este tutorial, pode criar um novo projeto e abrir o formulário rbEndUsr do diretório RBuilder \ Tutorials.

3 Feche o formulário rbEndUsr.

4 Selecione Arquivo | Save Project As a partir do menu Delphi e salve o projeto no diretório My RB Tutorials (estabelecido na página 181) sob o nome rbREProj.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Criar um novo formulário do Report Explorer

1 Selecione Arquivo | Abra o menu ... no menu principal do Delphi.

2 Localize a unidade ppExpFrm.pas no diretório RBuilder \ Source.

3 Abra este formulário.

4 Altere o nome do formulário de ppReportExplorerForm para rbReportExplorerForm.

5 Selecione o painel na parte superior do formulário (denominado `tbrExplorer`) e use o Inspetor de objetos para definir a cor para `clMaroon`.

6 Role até a parte inferior da unidade de formulário e inspecione a seção de inicialização. Deve conter o seguinte:

inicialização

```
ppRegisterForm (TppCustomReportExplorer,  
TrbReportExplorerForm);
```

finalização

```
ppUnRegisterForm (TppCustomReportExplorer);
```

fim.

Nota: O Delphi Code Editor mudou automaticamente o nome da classe do formulário de `ppReportExplorerForm` para `rbReportExplorerForm` quando você alterou o nome do formulário no Object Inspector. Esta chamada irá substituir o formulário explorador 'oficial' pelo novo formulário que criamos.

Nota: Para que a seção de inicialização desta nova unidade de formulário seja acionada (chamando assim a rotina `ppRegisterForm`), devemos incluir esta unidade no projeto. Se não incluirmos esta unidade no projeto, o explorador de relatório padrão será usado.

7 Salve o formulário com o nome `rbExpFrm` no diretório My RB Tutorials.

8 Feche o formulário.

9 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

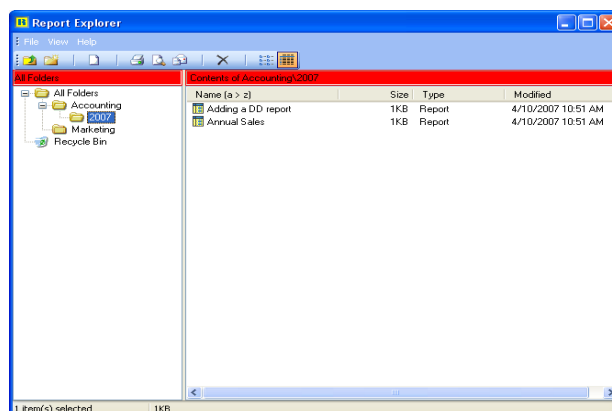
10 Adicione `rbExpFrm` ao projeto e remova o formulário da lista de criação automática do projeto (guia Opções | Formulários).

11 Selecione Arquivo | Salve no menu principal do Delphi.

12 Execute o aplicativo. Ao clicar no botão Iniciar, você verá uma barra de ferramentas marrom em vez de prateada (indicando que o formulário do explorer personalizado está sendo usado).

Observação: como você pode ver, agora você tem controle total sobre a aparência, o comportamento e o comportamento do formulário Report Explorer. Se você revisar o código-fonte deste formulário, verá que ele está chamando várias funções no componente Report Explorer. Essas funções estão documentadas na ajuda.

O aplicativo de tempo de execução personalizado deve ter a seguinte aparência:

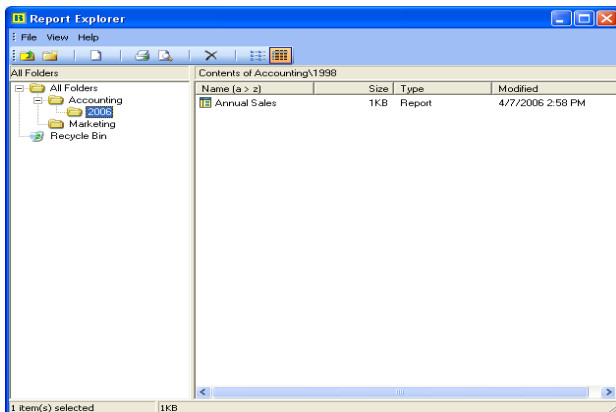


Construindo um aplicativo de relatório usando o InterBase

Visão geral

Este tutorial mostrará como fazer o seguinte:

- Configurar objetos de acesso a dados Delphi para uso pelo Report Explorer
- Crie as tabelas InterBase exigidas pelo Report Explorer
- Use os componentes Report, Report Designer e Report Explorer
- Crie as tabelas InterBase exigidas pelo Dicionário de Dados (opcional)



Revise o script SQL Create Tables para o Report Explorer

Nota: Este tutorial requer que você tenha o InterBase instalado e saiba como usá-lo. Os conceitos abordados neste tutorial podem ser adaptados a outros bancos de dados baseados em SQL. Os scripts SQL usados para criar as tabelas InterBase também podem ser usados como ponto de partida para a criação de scripts para outros tipos de bancos de dados SQL. No entanto, exemplos de vários produtos de banco de dados populares estão incluídos no ReportBuilder. Esses exemplos podem ser encontrados em RBuilder \ Demos \ EndUser Databases.

Nota: O Report Explorer está conectado às tabelas InterBase neste tutorial. Assim, as pastas e relatórios criados pelo usuário final são armazenados em tabelas do InterBase. O Report Designer também está conectado ao InterBase para permitir que o usuário final use o Query Wizard para construir visualizações de dados. Também é possível conectar o Report Explorer e o Report Designer a bancos de dados separados. Para alguns aplicativos, pode fazer sentido conectar o Report Explorer às tabelas do banco de dados local e conectar o Report Designer a um servidor de banco de dados como o InterBase. Nesse cenário, cada usuário final teria seu próprio conjunto de pastas e relatórios.

1 Selecione Arquivo | Abra no menu principal do Delphi para acessar a caixa de diálogo Abrir.

2 Selecione a opção Qualquer arquivo (*.*) Na lista Arquivos do tipo na parte inferior da caixa de diálogo.

3 Localize o diretório RBuilder \ Tutorials e abra o arquivo ExploreTables.sql.

4 Encontre a instrução Connect na parte superior do arquivo e analise os parâmetros de banco de dados, nome de usuário e senha. O nome do banco de dados é configurado para acessar o banco de dados de demonstração de funcionários instalado com o InterBase. Modifique os parâmetros conforme necessário para refletir sua instalação do InterBase.

```
CONNECTAR "C: \ Arquivos de programas \ Arquivos
comuns \ Borland Shared \ Data \ employee.gdb"
USUÁRIO "SYSDBA" SENHA "masterkey";
```

5 Revise as instruções de criação para a tabela de pastas. A tabela é criada primeiro, seguida por dois índices. Em seguida, um gerador e um gatilho são criados para o valor folder_id.

Nota: Não modifique as definições de tabela ou índice. Não adicione restrições de campo, como NOT NULL ou UNIQUE. Não defina uma chave primária. O Report Explorer foi projetado para oferecer suporte a bancos de dados de desktop e cliente / servidor. Este requisito limita a flexibilidade das definições da tabela. Adicionar restrições de campo, como as descritas acima, resultará em erros de tempo de execução Delphi em seu aplicativo de relatório de usuário final.

Código *O SQL para criar a tabela de pastas:*

```
/* criar tabela de pastas */

CRIAR A TABELA rb_folder
(folder_id INTEIRO,
nome VARCHAR (60),
parent_id INTEGER);

CRIAR ÍNDICE folder_idx ON rb_folder (folder_id);

CRIAR ÍNDICE parent_idx ON rb_folder (parent_id);

COMPROMETER-SE;

/* criar gerador para id da pasta */

CRIAR GERADOR folder_id_gen;
DEFINIR GERADOR folder_id_gen para 1;

/* criar gatilho para adicionar id de pasta exclusivo */

DEFINIR PRAZO !!;
CRIAR ACIONADOR set_folder_id PARA rb_folder
ANTES DE INSERIR
COMO
COMEÇAR
new.folder_id = gen_id (folder_id_gen, 1);
FIM !!
SET TERM; !!

COMPROMETER-SE;
```

6 Revise as instruções de criação para a tabela de itens. A tabela é criada primeiro, seguida por dois índices. Em seguida, um gerador e um gatilho são criados para o valor item_id.

Execute o script SQL para criar tabelas

1 Selecione Banco de Dados | Explore a partir do menu principal do Delphi para acessar o Database Explorer

2 Dê um clique duplo no alias do banco de dados correspondente ao seu banco de dados Interbase. O alias para uma instalação padrão do servidor InterBaser local é IBLOCAL.

3 Digite 'masterkey' como senha e clique em OK.

4 Clique com o botão direito para acessar o menu pop-up do banco de dados. Selecione 'ISQL' para acessar o utilitário InterBase ISQL.

5 Selecione 'Arquivo | Execute um script ISQL 'a partir do menu principal do utilitário InterBase ISQL. Abra o arquivo ExploreTables.sql. A 'Salvar saída em um arquivo?' a caixa de diálogo será exibida.

6 Clique no botão Não.

Código *O SQL para criar a tabela de itens*

```

/* criar tabela de itens */
CRIAR A TABELA rb_item
(item_id INTEIRO,
folder_id INTEGER,
nome VARCHAR (60),
item_size INTEGER,
item_type INTEIRO,
FLOAT modificado,
excluiu FLOAT,
modelo BLOB SUB_TYPE 0 SEGMENT SIZE 400);

CRIAR ÍNDICE item_idx SOBRE rb_item (item_id);

CRIAR ÍNDICE folder_item_name_idx SOBRE rb_item (folder_id, item_type, nome);

COMPROMETER-SE;

/* criar gerador para id do item */

CRIAR GERADOR item_id_gen;
DEFINIR GERADOR item_id_gen para 1;

/* criar gatilho para adicionar id de item exclusivo */

DEFINIR PRAZO !! ;
CRIAR ACIONADOR set_item_id PARA rb_item
ANTES DE INSERIR
COMO
COMEÇAR
    new.item_id = gen_id (item_id_gen, 1);
FIM !!
SET TERM; !!

COMPROMETER-SE;

```

Observação: o script SQL deve se conectar ao banco de dados e criar os objetos do banco de dados. Se você receber uma mensagem indicando que o script foi concluído com erros, revise as informações de saída - as mensagens de erro são exibidas após a instrução em que o erro ocorreu.

7 Feche a janela InterBase ISQL.

8 Feche o Database Explorer.

Criar uma nova aplicação

1 Selecione Arquivo | Novo aplicativo do menu principal do Delphi. Isso criará um novo projeto e um formulário em branco.

2 Defina o nome do formulário como 'frmEndUserInterBase'.

3 Defina a legenda do formulário como Solução de relatórios do usuário final.

4 Selecione Arquivo | Salvar como no menu Delphi e salvar o formulário com o nome rbEUIB no diretório My RB Tutorials (estabelecido na página 181).

5 Selecione Exibir | Gerente de projeto no menu principal do Delphi.

6 Clique com o botão direito sobre o nome do projeto no Project Manager (geralmente Project1.exe) e selecione a opção de menu Salvar.

7 Salve o projeto com o nome rbEUIBProj no diretório My RB Tutorials.

Nota: Salve seu trabalho com frequência para evitar a perda de alterações caso seu computador perca energia ou trave inesperadamente.

Crie componentes de acesso a dados para a tabela de pastas

1 Selecione a guia Data Access da paleta de componentes Delphi.

2 Adicione um componente de banco de dados ao formulário.

3 Configure o componente Banco de dados:

AliasName	IBLocal
Nome do banco de dados	rbData
Nome	rbData
LoginPrompt	Falso
Params	NOME DE USUÁRIO = SYSDBA PASSWORD = masterkey

4 Teste a conexão do banco de dados definindo a propriedade Connected do componente Database como True. Resolva quaisquer erros e redefina a propriedade para False.

5 Adicione um componente Tabela ao formulário.

6 Configure o componente Tabela:

DatabaseName	rbData
Nome	tblFolder
TableName	RB_FOLDER

7 Adicione um componente DataSource ao formulário.

8 Configure o componente DataSource:

DataSet	tblFolder
Nome	dsFolder

9 Selecione a aba RBuilder do Delphi compo-paleta atual.

10 Adicionar um componente DBPipeline



para o formulário.

11 Configure o componente DBPipeline:

Fonte de dados	dsFolder
Nome	plFolder
RefreshAfterPost	Verdadeiro
Visível	Falso

Observação: definimos a propriedade DBPipeline.Visible como False para que o Report Designer não exiba esse pipeline de dados.

Crie componentes de acesso a dados para a tabela de itens

1 Selecione a guia Data Access da paleta de componentes Delphi.

2 Adicione um componente Tabela ao formulário.

3 Configure o componente Tabela:

Nome do banco de dados	rbData
Nome	tblItem
Nome da tabela	RB_ITEM

4 Adicione um componente DataSource ao formulário.

5 Configure o componente DataSource:

DataSet tblItem
Nome dsItem

6 Selecione a aba RBuilder do Delphi compo-paleta atual.

7 Adicionar um componente DBPipeline  para o formulário.

8 Configure o componente DBPipeline:

Fonte de dados	dsItem
Nome	plItem
RefreshAfterPost	Verdadeiro
Visível	Falso

Crie os componentes ReportBuilder

1 Adicionar um componente de relatório  para o formulário.

2 Adicionar um componente Designer  para o formulário.

3 Defina a propriedade Report do Designer como ppReport1.

4 Configure o componente Designer:

DataSettings.DatabaseName rbData
DataSettings.SQLType sqSQL2
Tipo de banco de dados dtInterbase

Observação: a propriedade DataSettings do Designer contém várias propriedades usadas para controlar a conexão do banco de dados e o acesso aos dados. O SQLType deve ser definido como sqSQL2 para servidores de banco de dados e como sqlBDELocal para bancos de dados de desktop como Paradox e dBase.

5 Adicione um componente ReportExplorer ao formulário.

6 Defina a propriedade Designer do ReportExplorer como ppDesigner1.

7 Configure o componente ReportExplorer:

FolderPipeline	plFolder
FolderFieldNames.FolderId	FOLDER_ID
FolderFieldNames.Name	NOME
FolderFieldNames.ParentId	PARENT_ID
ItemPipeline	plITEM
ItemFieldNames.Deleted	EXCLUÍDO
ItemFieldNames.FolderId	FOLDER_ID
ItemFieldNames.ItemId	ITEM_ID
ItemFieldNames.ItemType	TIPO DE ITEM
ItemFieldNames.Modified	MODIFICADO
ItemFieldNames.Name	NOME
ItemFieldNames.Size	ITEM_SIZE
ItemFieldNames.Template	MODELO

Nota: Para simplificar, definimos nossas tabelas de banco de dados para que os nomes dos campos da tabela correspondam estreitamente às propriedades ItemFieldName e FolderFieldName. A exceção é o campo item_size, que é atribuído à propriedade ItemFieldNames.Size. 'Tamanho' é uma palavra reservada no InterBase e portanto não pode ser usada como um nome de campo.

Observação: embora seja aceitável nomear os campos de maneira diferente na tabela do banco de dados, não é aceitável alterar os tipos de dados ou outros atributos de campo, pois o Report Explorer depende desses nomes.

Execute o aplicativo

1 Selecione a guia Win32 da paleta de componentes Delphi.

2 Adicione um StatusBar ao formulário.

3 Configure o componente StatusBar:

Nome stbStatusBar
SimplePanel True

4 Adicione um componente Botão ao formulário.

5 Configure o componente Botão:

Nome btnLaunch
Rubrica Lançar

6 Adicione o seguinte código ao manipulador de eventos OnClick do botão:

```
Session.SQLHourGlass: = False;
```

```
se não(ppReportExplorer1.Execute) então  
  stbStatusBar.SimpleText: =  
  ppReportExplorer1.ErrorMessage;
```

Nota: Primeiro, definimos a propriedade SQLHourGlass como False. Não queremos que o cursor seja exibido ao usar o Report Explorer para criar pastas e relatórios. Em seguida, chamamos o método Execute para invocar o Report Explorer. Se não configuramos uma das propriedades do Report Explorer corretamente, o formulário não será exibido e False será retornado. O motivo da falha será exibido na 'barra de status'.

7 Adicione as seguintes unidades à cláusula 'usa' na parte superior da unidade de formulário: daDatMan, daDBBDE.

Nota: A unidade daDatMan contém uma classe de gerenciador de dados que disponibiliza o espaço de trabalho de dados no Report Designer. A unidade daDBBDE associa uma classe de visualização de dados baseada em BDE com o Query Wizard e o Query Designer. Portanto, essas ferramentas criarão instâncias dessa visualização de dados quando chamadas. Este design permite que outras classes de visualização de dados de consulta sejam criadas e associadas a essas ferramentas.

8 Selecione o projeto | Compile rbEUIBProj. Corrija quaisquer problemas de compilação.

9 Selecione Arquivo | Salve no menu principal do Delphi.

10 Execute o projeto.

11 Clique no botão Iniciar. O Report Explorer deve ser exibido.

Nota: Se o Report Explorer não for exibido, uma explicação será fornecida no painel na parte inferior da janela.

Criar novas pastas

1 Selecione Arquivo | Novo | Pasta no menu principal do Report Explorer.

2 Digite Marketing no nome da pasta e pressione a tecla Enter.

3 Selecione a pasta principal (intitulada Todas as pastas).

4 Clique com o botão direito do mouse no espaço em branco da Árvore de pastas e selecione a opção Nova pasta no menu pop-up.

5 Digite Contabilidade no nome da pasta e pressione a tecla Enter.

6 Clique com o botão direito do mouse no espaço em branco da Árvore de pastas e selecione a opção Nova pasta. Uma pasta deve ser criada na pasta Contabilidade.

7 Digite 1998 nesta pasta e pressione a tecla Enter.

8 Clique com o botão direito do mouse na pasta 1998 e selecione a opção de menu Novo Relatório.

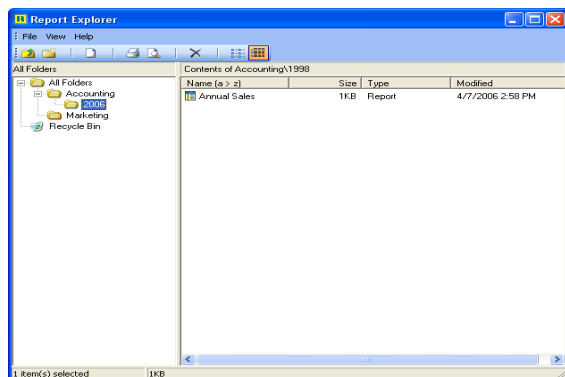
9 Selecione Arquivo | Salvar como no menu principal do Report Designer e salve o relatório com o nome Vendas anuais.

10 Feche o Report Designer. O relatório de vendas anuais deve ser exibido na pasta 1998.

O aplicativo deve ficar assim depois de salvar o relatório:

Nota: Você pode criar e configurar relatórios a partir deste ponto, assim como faria no Delphi. E como você pode ver, o Report Explorer é bastante fácil de usar devido à sua semelhança com o Windows Explorer.

Observação: o uso do dicionário de dados é opcional. O script SQL para adicionar as tabelas do Dicionário de Dados é denominado DataDictTables.sql e está localizado no diretório RBuilder \ Tutoriais. Você pode consultar o tutorial chamado "Adicionando suporte a dicionário de dados ao aplicativo de usuário final" para obter informações adicionais sobre como configurar os componentes e propriedades relevantes.



UMAPPENDIX

Para onde eu vou daqui?

O ReportBuilder tem muitos outros recursos que podem ajudar iniciantes e usuários experientes.

The Digital Metaphors Website

O material abaixo, com exceção dos arquivos de ajuda, pode ser encontrado em:

www.digital-metaphors.com

Learning ReportBuilder

Depois de construir seu aplicativo de usuário final e estar pronto para distribuição, você pode enviar seus usuários finais ao nosso site para baixar o Learning ReportBuilder, que é um sistema de aprendizado completo desenvolvido para usuários finais. Ele vem com um aplicativo independente, um banco de dados, uma série de tutoriais de 125 páginas em formato PDF e um arquivo de ajuda completo com um glossário.

Ajuda do ReportBuilder

A Ajuda da RB pode ser acessada no menu principal do Delphi ou no menu principal do Report Designer. Você também pode selecionar um objeto e pressionar F1 para obter ajuda sobre esse objeto. O arquivo de ajuda da RB inclui informações de instalação e documenta a Paleta de componentes Delphi, bandas e grupos, e a Paleta de componentes RB. Ele também fornece dicas de solução de problemas e um histórico de versões de RB.

Ajuda RAP

A Ajuda RAP pode ser acessada no arquivo de Ajuda RB. Ele oferece uma explicação de RAP, um conjunto de tutoriais, informações sobre programação com RAP e uma seção de referência de linguagem. Ele também fornece dicas sobre como dimensionar o RAP para atender às necessidades dos usuários e estender o RAP.

Grupos de notícias ReportBuilder

A seguinte lista de grupos de notícias pode ser acessada na seção de Suporte do nosso site:

digital-metaphors.public.reportbuilder.tech-tips

dicas de tecnologia postadas por metáforas digitais

digital-metaphors.public.reportbuilder.general

construção de relatório geral

digital-metaphors.public.reportbuilder.rap

ambiente de desenvolvimento Pascal em tempo de execução

digital-metaphors.public.reportbuilder.subreports

usando sub-relatórios de formato livre

digital-metaphors.public.reportbuilder.end-user

desenvolver soluções de relatórios para o usuário final

digital-metaphors.public.reportbuilder.datapipelines

relatório de acesso a dados: banco de dados, arquivos de texto, listas de sequência ...

digital-metaphors.public.reportbuilder.devices

saída de relatório: impressora, tela, arquivo, arquivo ...

Grupos de notícias ReportBuilder - cont.digital-metaphors.public.reportbuilder.component-writing

componentes de relatórios personalizados, linhas de dados e dispositivos

digital-metaphors.public.reportbuilder.beta

programa beta para os próximos lançamentos

Aproveite as vantagens desses grupos de notícias: eles oferecem a você a oportunidade de examinar e / ou ingressar na comunidade ReportBuilder. Por favor, leia o conjunto de diretrizes antes de postar.

Suporte adicional

Os newsgroups são um excelente recurso de suporte técnico, mas se você preferir outros métodos, eles estão disponíveis:

- Mail: support@digital-metaphors.com

Inclua uma descrição concisa do problema e especifique as seguintes informações: versão Delphi, versão ReportBuilder, sistema operacional e modelo da impressora.

- Telefone: 214.239.9471

O suporte pago por minuto de um engenheiro da Digital Metaphors está disponível por US \$ 3,00 por minuto. Os engenheiros da Digital Metaphors estão disponíveis das 9h00 às 16h30 (horário central), de segunda a sexta-feira. Observe que o uso de e-mail é o método preferido para relatar bugs. O suporte por telefone é cobrado em uma taxa fixa de US \$ 3,00, independentemente do assunto.

- Fax: 267.501.5740

eUNDEX

UMA

Endereço

exemplo de código para 69

Advanced Component Palette 98

Align 101

Alinhar ou Barra de Ferramentas de Espaço

descrição de 101

uso de 231, 250 propriedade

AllowDataSettingsChange 153 propriedade

AllowEditSQL 181

Propriedade AllowPrintToArchive 123

Propriedade AllowPrintToFile 125

Propriedade AllowSaveToFile 153

implantação do aplicativo 131, 225

aplicativos

Adicionando suporte de dicionário de dados ao usuário final

App 415

Construindo um aplicativo de relatório. usando InterBase 423

Construindo um Aplicativo de Relatórios 401 Construindo um

Aplicativo de Relatórios de Usuário Final. Arquivo 409

imprimir relatório para a

propriedade 123 ArchiveFileName

123 ArchiveReader Component

descrição de 27, 151

ASCII

cerca de 125

ave 197

Média 19

B

bandas 14

Evento BeforePrint de 20

detalhe 4, 8

20 dinâmicos

rodapé 4, 234

cabeçalho 4, 7

Propriedade PrintHeight de 20

redimensionamento 17

resumo 265

título 325

Propriedade visível de 20

BarCode

descrição de 30, 97

alternativas BDE

Advantage 55

Opus 55

BDEPipeline 49

cerca de 49

descrição da propriedade

27, 151 BinName 119

Campo BLOB 132

favoritos

limitar a passagem de dados usando 51

a propriedade BottomOffset 33

uso de 275

método de caixa delimitadora 233

C

Calc Workspace 148, 193

Editor de Código 196

Code Explorer 194

Caixa de ferramentas de código 197

visão geral 193

Propriedade CalcOrder 73

Propriedade CalcType 75

campos calculados 19

cálculos

realizando 19, 73

CheckBox

descrição do 30, 97

Editor de código

compilar 197

deletar 197

novo 197

reverter 197

salvar 197

Code Explorer

visualização de eventos 194

escopo global 196

visualização do módulo 195

visualização de variáveis 194

Relatórios de código

como 83

Caixa de ferramentas de código

- Guia de dados 198
- Guia de idioma 199
- Aba Objetos 198

CodeSite

- compilando 213

Funções CodeSite

- usando 212

Propriedade de agrupamento 119

colunas

- usando para criar etiquetas de endereçamento 329
- componente
- seleção
 - Ctrl-clique método 232
 - Método Shift-clique 233
- visibilidade 20

editor de componente 17

Paleta de componentes 13

componentes

- ArchiveReader 27, 151
- Código de Barras 30, 97
- BDEPipeline 27, 151
- CheckBox 30, 97
- CrossTab 31, 98
- DataDictionary 152
- DBBarCode 31, 97
- DBCalc 31, 97
- DBCheckBox 31, 97
- DBImage 31, 97
- DBMemo 30, 97
- DBPipeline 27, 151
- DBRichText 31, 97
- DBTeeChart 31, 97
- DBText 30, 97
- Designer 152
- Imagem 30, 96
- JITPipeline 27, 151
- Rótulo 8, 96
- Memo 29, 96
- Região 31, 98
- Relatório 27, 151
- ReportExplorer 152
- RichText 29, 96
- Forma 30, 96
- Sub-relatório 31, 98
- SystemVariable 30, 96
- TeeChart 30, 96
- TextPipeline 27, 151
- Variável 30, 96
- Visualizador 27, 151

Concatenação

- código de exemplo para 68

Grande Total Condicional

- cerca de 80

Total condicional do grupo

- cerca de 79

Configurar propriedade

- uso de 351

Configurando Relatórios 67

Grupo Continuado

- código de exemplo para 70

Controlling Data Traversal

- Cerca de 50

Copia propriedade 119

Contar

- cerca de 74
- calculando 19
- Registros Mestre / Detalhe 75

Componente Crosstab

- descrição de 31, 98
- uso de 350

Crosstab Designer 350

Relatório de tabela cruzada

- criando 349

Soma Cumulativa

- cerca de 78

Classe CustomDataPipeline 61

Classe CustomPreviewer 117

D

classe daCustomDataView 183

classe daDataSet 184

classe daDataView 183

DADE

- cerca de 167
- arquitetura 183
- configurando 181
- estendendo 185

daQueryDataView classe 184

daSession classe 184

daSQL classe 184

dados

- filtrando 19

Acesso de dados

- cerca de 53
- tipos de pipeline de dados
- 49 descrição de 25
- objetos
 - configurando 7

Ambiente de desenvolvimento de acesso a dados

Ver DADE

Paleta de componentes de dados

- descrição de 97

Data Dialog 94

Dicionário de Dados 159

módulo de dados 132, 226

pipelines de dados

Propriedade BookmarkList de 51

visualizações de dados e 167

na árvore de dados 8

Propriedade RangeBegin de 51

Propriedade RangeEnd de 51 recuperando

valores de campo por meio de 59 Processo

de Dados 25

Guia de Dados 147

passagem de dados

controlando 50

controle por meio de pipelines de dados 49

Árvore de dados

cerca de 105

criação de um relatório por meio da guia

de dados 229 105

descrição de 95

configurações de arrastar e soltar da

guia 15 de layout 15, 105

uso do espaço de trabalho de

dados 232, 274, 293 167

componentes com reconhecimento de dados

criando através da árvore de dados 15

Ver DB

modelos de banco de dados 134

Propriedade DatabaseName 181

propriedade DataDictionary 181

propriedade DataSettings 153

DataSource Component

uso de 7

DataType 20

exibição de dados

cerca de 147, 167

classes 188

modelos

a visão do usuário final

185-186 a implementação 187

DBBarCode

descrição de 31, 97

DBCalc

cerca de 73

descrição de 31, 97

uso de 268

DBCheckBox

descrição de 31, 97

DBImage

descrição de 31, 97

uso de 274

DBMemo

descrição de 30, 97

uso de 275

DBPipeline 7

cerca de 49

descrição de 27, 151

uso de 7

DBRichText

descrição de 31, 97

DBTeeChart

descrição de 31, 97

DBText

descrição de 30, 97

uso de 8, 251

de 196

Componentes Delphi 27

Delphi Event Model 65

Delphi Objects 59

implantando

como um exe 131

como pacotes 131

solução de relatórios do usuário final e 225

relatórios de implantação

em uma tabela de banco de

dados 132 em um executável 131

em arquivos de modelo 132

design 4

Guia Design 89

área de trabalho de design 89,

149 Designer

descrição de 152

Propriedade de designer 157

Faixa de Detalhe

cerca de 4

uso de 8

Diálogos da propriedade DeviceType

123-125

Dados 94

Grupos 93

Configuração de página 92

Imprimir 91

Imprimir para configuração de arquivo

93 formatos de exibição

configuração 17

Propriedade DocumentName 119

recursos de arrastar e soltar

Ver Árvore de Dados

Barra de Ferramentas Desenhair

descrição de 103

uso de 250

dule 195

Propriedade duplex 119

Configuração Dinâmica 67

E

Editar Barra de Ferramentas
14, 99 EditMask
 como DisplayFormat 20
solução de relatório de usuário final 147, 161
ent 201
equação
 do paradigma de relatório 25, 147
ert 197
modelo de evento 65
eventos 65

F

Editor de Campo 57
valores de campo
 recuperando via pipelines de dados
59 Propriedade FieldFieldNames 159
Propriedade FieldPipeline 159 filtrando
dados 19
Propriedade FolderFieldNames 157
Propriedade FolderPipeline 157 Cor da
fonte
 exemplo de código para a
propriedade 67 Font
 configuração on-the-fly 67
Faixa de rodapé
 cerca de 4
 uso de 234
Barra de ferramentas de formatação
 cerca de 14
 descrição de 100
Emulação de formulários com um relatório de imagem WMF
 criando 281
Funções 209

G

g 202
geração 5, 115
total geral
 cerca de 77
 condicional 77
grupo
 criando um 257
banda de cabeçalho do grupo
 regiões condicionais em 71
 grupos
 Etiqueta 'Continuação ...' e 70 criando através
 do Assistente de Relatório 110 totais em 76

 dois totais de passagem em
81 Grupos de Diálogo 93
Grupos, cálculos e o relatório de faixa de resumo
 criando 255

H

Faixa de cabeçalho
 cerca de 4
 uso de 7
régua horizontal
 opções para unidades de medida 249
HTML 129

eu

Propriedade do ícone 153
Componente de imagem
 descrição de 30, 96
 InterBase
 criando um aplicativo de relatório.
usando 423 International Language
Support 139 ItemFieldNames propriedade
157 ItemPipeline propriedade 157

J

JITPipeline
 cerca de 49, 59
 descrição de 27, 151
 uso de 341

eu

Componente de etiqueta
 descrição de 96
 uso de 248
Assistente de modelo de etiqueta
 uso de 331
rótulos
 criado para mailing 329
suporte a idiomas
 cerca de 139
 traduções personalizadas
 141 idioma padrão 140
les 194
Componente de Linha
 uso de 250
tabelas de pesquisa / consultas 19

M

etiquetas de correspondência

criando 329

MailMerge

uso de 347

MaintainAspectRatio

uso de 274

Propriedade de margens 119

Relatório Mestre / Detalhe

criando um 289

Relatório Mestre / Detalhe / Detalhe

criando um 303

Máximo 19

Componente de Memorando

descrição da propriedade

29, 96 Overflow 34

Esticar propriedade 34

alongamento 20

Mínimo 19

Propriedade ModalForm 157

ms 201

MS Access

SQLType 181

N

Acesso nativo a dados proprietários 61 Barra de

ferramentas Nudge 102

O

OnCalc 20

Evento OnCalc 73

OnCalcFields evento 19

Ajuda On-Line 163

Evento OnPrint 67

OnReset evento 73

Propriedade de orientação 119

Propriedade OverflowOffset 33

P

pacotes 137

objetos de página 116

Diálogo Configuração de Página 92

estilo de página 43

PageLimit 18

Propriedade PaperHeight 119

Propriedade PaperName 119

Propriedade PaperWidth 119

Executando Cálculos 73 pilha

197

empilhando 202

Guia de visualização 89

Visualização da área de trabalho 149

visualizando 117

Imprimir Diálogo 91

método de impressão 123

formulário de visualização de impressão 115

405 personalizado

Imprimir para arquivo de texto ASCII 125

Caixa de diálogo de configuração de

imprimir para arquivo 93 Propriedade

PrinterName 119 Propriedade PrinterSetup

119 impressão

de um arquivo de texto 337 para

um arquivo de texto 333 para a

impressora 115

para a tela 115

configurações de impressão 119

processo 4

dados proprietários

acesso nativo a 61

Q

Designer de Consulta

adicionar critérios de pesquisa por meio de 173-175

campos de concatenação por meio de 178-179, criando

um SQL GROUP BY via 175-177, editando SQL gerado

por 180

uso de 173

Assistente de Consulta

cerca de 169

Crie um Dataview baseado em consulta simples via 169-
172

R

extensão de arquivo raf

123 RAP 201

cerca de 191

adicionando classes e propriedades 210

Adicionando funções a 367

adicionando funções a 209

Calc Workspace 193

procedimentos de chamada e funções 204

CodeSite

usando 212

Suporte para CodeSite 211

codificando um manipulador de eventos 201

Componentes de codificação de cores 359

compilando manipuladores de eventos 202

Concatenando Campos 355 configurando
205

depuração 211

declarando constantes globais 203

declarando procedimentos globais 203

declarando variáveis globais 202

declarando constantes locais 202

declarando variáveis locais 202

Exibindo Formulários Delphi de 387

Duplex Dinâmico 363

configurações de usuário final 205

estendendo 209

Estendendo RTTI 373

visão geral 191

funções de passagem 209

Descrição de impressão dos critérios de AutoSearch 373

procedimento e parâmetros de função 204 programas 201

Escrevendo código RAP 201

propriedade RAPInterface 153

propriedade RAPOptions 153

Region Component

descrição de 31, 98

exemplo de código para 71

usado para agrupar logicamente componentes dinâmicos 271

Relatório 25

Um relatório simples da maneira mais difícil 247

arquivamento 121

Criando um relatório por meio da árvore de dados 229 Criando

um relatório por meio do assistente de relatório 237 criando a

partir de uma classe ancestral 402

implantação 131, 225

emulação 127

Solução de relatório de usuário final

154 carregando e salvando 134

Propriedade PageLimit 18

método de impressão 9

tabular

criando 238

modelos 133

vertical

criando 240

Arquivo de arquivo de relatório 123

Componente de relatório

descrição de 27, 151

uso de 7

Propriedade visível de 20

componentes do relatório 29-31

criação do relatório

design 4

geração 5, 115

processo 4

selecione 3

Report Designer

cerca de 89

uso de 283

trabalhando com o mecanismo

de relatório 17 Report Emulation Text

File 127

passagem de dados e o 50

Report Explorer

cerca de 217

Personalizando o Report Explorer Formulário 421

descrição de 157

barra de ferramentas 217

layout do relatório

cerca de 25

salvando em um banco de dados 134

salvando em um arquivo 133

esboço do relatório

Ver Árvore de Relatório

resultado do relatório 25

Relatório de propriedade 153

modelos de relatório 133

Árvore de Relatório

cerca de 14

descrição de 95

Assistente de Relatório

criar um relatório por meio do 237

criar um relatório simples por meio do 107-109

criar grupos por meio do 110-111

Propriedade Report.Template.Format 133

ReportBuilder

Componentes Delphi de

27 fundamentos 25-141

ReportBuilder Enterprise

Componentes Delphi de 151

implantando 225

fundamentos 147-226

Componente ReportExplorer

descrição de 152

tabela de pastas de 217

itens tabela de 217

dispositivo ReportTextFile 127

Propriedade ReprintOnOverflow 33

Propriedade ResetGroup 75

Componente RichText
 descrição de 29, 96
 uso de 346

Editor RichText
 uso de 347

RTF 128

extensão do arquivo rtm

133 réguas 14

S

Escopo 196

selecione 3

Componente de forma 181 da

propriedade SessionType
 descrição de 30, 96
 uso de 293

formas

 alongamento 20

Propriedade ShiftRelativeTo 33

propriedade ShowComponents 153

propriedade ShowPrintDialog 123, 125

Size Toolbar

 descrição de 102
 uso de 268

SQL

 GRUPO POR
 criando 175
 com aplicações 425

Paleta de componentes padrão
 descrição de 96

Barra de ferramentas padrão

 descrição de 98

Barra de status 14

Esticamento 33 da propriedade

StopPosition

 memorandos e formas 20

Propriedade StretchWithParent 33

Opção de menu de estilo

 uso de 352

SubReport Component

 descrição de 31, 98
 exemplo de código para
 uso de 72 de 291

SubReports

 para ligar relatórios juntos 319
 seção-estilo
 uso de 319

Soma 19

Banda Resumida

 uso de 265

fornecendo dados

 cerca de 49

Componente SystemVariable

 descrição de 30, 96
 uso de 252

T

Componente de Tabela

 uso de 7

criação de mesa 410

Propriedade TableFieldNames 159

propriedade TablePipeline 159 tabelas

 criando 416

 povoar 417

Relatório tabular da propriedade

TabsVisible 153

 criando 238

Componente TeeChart

 descrição de 30, 96

modelos 133

arquivos de texto 57

Dispositivo TextFile 125

Componente TextPipeline

 cerca de 49

 descrição de 27, 151

objetos TField 19

Banda do Título

 uso de 325

folha de rosto

 exemplo de 325

barras de ferramentas 95-103

 Paleta de componentes avançados 98

 Alinhar ou espaçar 101, 231, 249 Paleta
 de componentes 13

 Paleta de componentes de dados 97

 Árvore de dados 95, 232

 Sorteio 103, 250

 Editar 99

 Formato 14, 100

 Nudge 102

 Report Explorer 217

 Tamanho 102, 268

 Standard 98

 Paleta de componentes padrão 96 modo

de duas passagens

 cerca de 240

você

Propriedade UseDataDictionary 181
Interface do usuário 13

V

Componente Variável
 cerca de 73
 Propriedade DataType de
 20 descrição de 30, 96
 evento OnCalc de 20
 uso de 309
relatório vertical
 criando 240
Componente de visualizador
 descrição de 27, 151

Z

ampliação
 no formulário de visualização de impressão 117