

QuickReport 2.0b  
Registre 1997 QuSoft COMO  
QuSoft COMO  
Fred Olsensgt. 1

Doc aceleram 1.2

1 INTRODUÇÃO 1-8  
PREFÁCIO 1-9  
SOBRE QUICKREPORT 1-11  
SOBRE ESTA DOCUMENTAÇÃO 1-12  
2 QUICKREPORT 2-13 USANDO  
O PRIMEIRO RELATÓRIO 2-14  
Um relatório 2-14 simples  
A faixa diferente digita 2-15  
Dados 2-16 conectando  
Filtrando e Ordenando os Dados 2-16  
Texto somando e Dados Fields 2-17  
RELATÓRIOS 2-19 CRIANDO  
Grid, Unidades e Zune 2-19  
Tamanho de papel e margens 2-20  
Selecionando um Manancial 2-21  
Título de relatório e Descrição 2-22  
Faixas 2-22 somando  
Classificando segundo o tamanho as Faixas 2-24  
Faixa que Imprime Ordem 2-25  
Faixas virando de tempo em tempo 2-26  
Componentes Imprimíveis somando para o Relatório 2-27  
Sobre o Texto Componentes 2-28  
Texto 2-30 Estático imprimindo  
Banco de dados imprimindo Fields 2-31  
Formas 2-32 imprimindo  
Imagens 2-32 imprimindo  
RichText Formatted imprimindo Memorandos 2-33  
EXPRESSÕES 2-35 USANDO  
Expressões 2-35 comondo  
TQRExpr 2-38 usando  
Empilhe Tamanho 2-39  
GRUPOS 2-40 CRIANDO  
3 TÉCNICAS 3-43 AVANÇADAS  
RELATÓRIOS CRIANDO ATRAVÉS DE CÓDIGO 3-44  
QRPRINTER 3-48 USANDO  
4 QUICKREPORT 4-49 SE EXPANDINDO  
SOBRE QUICKREPORT SOMAR-ONS 4-50  
  
ESCREVENDO UM COMPONENTE 4-52 IMPRIMÍVEL  
Uma Amostra CheckBox Componente 4-52  
Dados Componentes 4-54 Atentos  
AMPLIANDO A EXPRESSÃO EVALUATOR 4-55  
5 QUICKREPORT REFERÊNCIA 5-57  
ADDPRINTABLE MÉTODO 5-58  
EVENTO de AFTERPREVIEW PARA TQUICKREP 5-60  
EVENTO de AFTERPRINT PARA TQUICKREP 5-61  
EVENTO de AFTERPRINT PARA ATA 5-62  
ALINHAMENTO PROPRIEDADE 5-63  
ALIGNTOBAND PROPRIEDADE 5-64

ALIGNBOTTOM PROPRIEDADE 5-65  
ALLDATASETS PROPRIEDADE 5-66  
APPLYSETTINGS MÉTODO 5-67  
AUTOSIZE PROPRIEDADE 5-68  
AUTOSTRETCH PROPRIEDADE 5-69  
PROPRIEDADE 5-70 DISPONÍVEL  
BANDLIST PROPRIEDADE 5-71  
PROPRIEDADE de FAIXAS PARA TQRSUBDETAIL 5-72  
FAIXAS PROPRIEDADE 5-73  
BANDTYPE PROPRIEDADE 5-74  
EVENTO de BEFOREPRINT DE TQUICKREP 5-76  
EVENTO de BEFOREPRINT PARA ATA 5-77  
ESCOVE PROPRIEDADE 5-78  
LEGENDA PROPRIEDADE 5-79  
CENTRE PROPRIEDADE 5-80  
CHILDBAND PROPRIEDADE 5-81  
COLORA PROPRIEDADE 5-82  
COLUMNTOPPOSITION PROPRIEDADE 5-83  
CÓPIAS PROPRIEDADE 5-84  
CURRENTCOLUMN PROPRIEDADE 5-85  
CURRENTX PROPRIEDADE 5-86  
CURRENTY PROPRIEDADE 5-87  
DADOS PROPRIEDADE 5-88  
DATAFIELD PROPRIEDADE 5-89  
DATASET PROPRIEDADE 5-90  
PROPRIEDADE de DATASET PARA TQRDBTEXT 5-91  
DESCRIÇÃO PROPRIEDADE 5-92  
DESENHISTA PROPRIEDADE 5-93  
DISPOSITIVO PROPRIEDADE 5-94  
MOTORISTA PROPRIEDADE 5-95  
PROPRIEDADE 5-96 HABILITADA  
EXPORTFILTER PROPRIEDADE 5-97  
PROPRIEDADE 5-98 EXPORTANDO  
EXPORTTOFILTER MÉTODO 5-99  
PROPRIEDADE de EXPRESSÃO PARA TQREXPR 5-100  
PROPRIEDADE de EXPRESSÃO PARA TQRGROUP 5-101  
MANANCIAL PROPRIEDADE 5-102  
FOOTERBAND PROPRIEDADE 5-103  
FORCENWCOLUMN PROPRIEDADE 5-104  
FORCENWPAGE PROPRIEDADE 5-105  
HEADERBAND PROPRIEDADE 5-106  
LEFTOFFSET PROPRIEDADE 5-107  
LINHAS PROPRIEDADE 5-108  
MASCARE PROPRIEDADE 5-109  
DOMINE PROPRIEDADE PARA TQRCHILDBAND 5-110  
  
DOMINE PROPRIEDADE PARA TQREXPR 5-111  
NEWCOLUMN MÉTODO 5-112  
NEWPAGE MÉTODO 5-113  
ONENDPAGE EVENTO 5-114  
ONNEEDDATA EVENTO 5-115  
ONPREVIEW EVENTO 5-116  
ONSTARTPAGE EVENTO 5-117  
OPÇÕES PROPRIEDADE 5-118  
ORIENTAÇÃO PROPRIEDADE 5-119  
OUTPUTBIN PROPRIEDADE 5-120

CHAME PROPRIEDADE 5-121  
PAGENUMBER PROPRIEDADE 5-122  
PAPERLENGTH PROPRIEDADE 5-123  
PAPERSIZE PROPRIEDADE 5-124  
PAPERSIZESUPPORTED PROPRIEDADE 5-125  
PAPERWIDTH PROPRIEDADE 5-126  
PARENTFONT PROPRIEDADE 5-127  
PIXELSPERX PROPRIEDADE 5-128  
PIXELSPERY PROPRIEDADE 5-129  
PREPARE MÉTODO 5-130  
PREVEJA MÉTODO 5-131  
IMPRIMA MÉTODO 5-132  
IMPRESSORA PROPRIEDADE 5-133  
PRINTERSETTINGS PROPRIEDADE 5-134  
PRINTERSETUP MÉTODO 5-135  
PRINTBACKGROUND MÉTODO 5-136  
PRINTIFEMPTY PROPRIEDADE 5-137  
QRPRINTER PROPRIEDADE 5-138  
RECORDCOUNT PROPRIEDADE 5-139  
RECORDNUMBER PROPRIEDADE 5-140  
REPORTTITLE PROPRIEDADE 5-141  
RESETAFTERPRINT PROPRIEDADE 5-142  
RESETPAGEFOOTERSIZE MÉTODO 5-143  
SHOWPROGRESS PROPRIEDADE 5-144  
ESTIRE PROPRIEDADE 5-145  
TEXTO PROPRIEDADE 5-146  
TPRINTERSETTINGS CLASS 5-147  
TQRABOUTBOX CLASS 5-149  
TQRBAND COMPONENTE 5-150  
TQRCHILDBAND COMPONENTE 5-152  
TQRDBIMAGE COMPONENTE 5-154  
TQRDBTEXT COMPONENTE 5-155  
TQREXPR COMPONENTE 5-157  
TQRIMAGE COMPONENTE 5-159  
TQRLABEL COMPONENTE 5-160  
TQRMEMO COMPONENTE 5-162  
TQRSHAPE COMPONENTE 5-163  
TQRSHAPETYPE TYPE 5-164  
TQRSUBDETAIL COMPONENTE 5-165  
TQRSYSDATA COMPONENTE 5-167  
TQUICKREP COMPONENTE 5-168  
PROPRIEDADE 5-170 TRANSPARENTE  
AMOLDE PROPRIEDADE 5-171  
CLASSIFIQUE SEGUNDO O TAMANHO PROPRIEDADE PARA ATA 5-172  
UNIDADES PROPRIEDADE 5-173  
WORDWRAP PROPRIEDADE 5-174  
ZUNA PROPRIEDADE 5-175  
6 APÊNDICES 6-176  
Um - CLASSE de QUICKREPORT HIERARQUIA 6-177  
B - ARQUIVO DESCRIÇÕES 6-179  
C - CONVERTENDO DE QUICKREPORT 1 6-180

1 introdução

Prefácio e informação geral sobre QuickReport e esta documentação

PREFÁCIO

Quando eu comecei escrevendo para QuickReport isto primeiro era criar uma solução de impressão mínima para um pouco projeto escrito em Delphi. Evoluiu logo em um projeto separado se querido ficar mais familiar com o Delphi VCL e escreve componentes e programação de Windows geral. Pensando que talvez outra pessoa pudesse usar isto que eu lancei uma versão muito cedo como uma utilidade de impressão livre na Internet em 1995 de junho.

Julgando da resposta de outros fomentador de Delphi eu não estava a favor o único com a necessidade por este tipo de ferramentas e desde então ninguém mais estava na ocasião disponível a decisão foi tomada ampliar isto um pouco mais e lança isto como shareware.

A beleza de QuickReport 1 era a simplicidade. Podendo criar quase qualquer tipo de relatório com um gerador de relatório realmente escrito em menos de 5.000 linhas de código mostra a força da ferramenta subjacente. QuickReport usado o desenhista de forma de Delphi como um desenhista de relatório, as conexões de banco de dados de Delphi para adquirir dados e até mesmo o deixar usam Delphi codificam para manipular o relatório diretamente durante preparação.

Até mesmo algumas pessoas a Borland gostaram de QuickReport e quando Delphi 2.0 foi lançado 1996 cedo incluiu a 32 versão de pedaço cheia de QuickReport. Este chumbo para um aumento enorme no número de usuários e QuickReport realmente foi empurrado a seus limites - se lembra foi projetado para fazer um trabalho muito simples originalmente e tinha superado quase seu desígnio interno e estrutura agora. Ficou claro logo que a próxima geração de QuickReport precisaria ser um total reescreva, não quebrar com as idéias de como QuickReport deveria trabalhar mas renovar um desígnio mais pensativo, prático e flexível de solo.

O resultado é o que você tem agora, QuickReport 2.0. consiste de cerca de 15.000 linhas de código mas não soma muito mais para uma aplicação que versão 1 fez! É mais rápido e mais poderoso que antes de e pode ser ampliado mais adiante em muitas formas. Este lançamento é só o primeiro passo em caminho que conduz um até mesmo mais poderoso, flexível e fácil usar informando ferramenta. As idéias e características para a próxima geração de QuickReport já são fixas e por favor estão preparado ser surpreendido!

Embora escrevendo para QuickReport fosse principalmente um um trabalho de homem há algumas pessoas que merecem muito crédito. Borland International foi muito útil, especial graças a Diane, Danny e Pancada. David Berneda, o autor de TeeChart e Magne Nilsen de Software de Vega também foi de grande ajuda.

Finalmente graças a todos os usuários de Delphi que apoiaram QuickReport, notificações de bicho enviadas e palavras encorajadoras.

Allan Lochert

SOBRE QUICKREPORT

QuickReport 2.0 inclui as características seguintes:

- Banded gerador de relatório
- desígnio de relatório Visual
- Delphi VCL Nativo escrito em 100% Objeto Pascal
- plano de relatório Extremamente preciso
- Compatible com Delphi 1.0, 2.0 e 3.0
- 100 - 350Kb pegada, dependendo de características usadas,
- Multithreaded
- pré-estréia Imediato
- relatórios de duração Ilimitados
- campos de memorando de duração Ilimitados
- RichText formatou memorandos

- Report de qualquer BDE fonte de dados compatível, inclusive ODBC,
- Report qualquer dados que usa evento dirigido relatórios
- Link número ilimitado de jogos de dados em relatórios de detalhe de mestre
- número Ilimitado de grupos
- Combinação relatórios
- End usuário relatório designio componente
- pré-estréia de Customizable
- Expandable com componentes imprimíveis novos como quadros, códigos de barra, formatos de gráficos avançados e mais
- evaluator de expressão Avançado com construtor de expressão
- Report o perito
- Report criação classifica para criar relatórios
- controle de impressora Cheio
- Preview MDI e SDI compatível, pré-estréia número ilimitado de relatórios simultaneamente
- Export relatórios para texto de ASCII, vírgula delimitou ou HTML. Escreva próprio adicione filtros de exportação
- que código de fonte Cheio incluiu
- Backed por Borland internacional e apoiou por Borland apoio Técnico

## SOBRE ESTA DOCUMENTAÇÃO

### 2 QuickReport usando

Esta parte dá uma explicação de como criar relatórios básicos e como prever e os imprimir em momento de designio e em momento de corrida de suas aplicações

#### O PRIMEIRO RELATÓRIO

Esta seção é uma explicação pequena de como montar um relatório básico. Os componentes de QuickReport mais comuns são descritos brevemente junto com como conectar seu relatório para uma mesa e chamar isto de sua aplicação.

Um relatório simples

QuickReport é um gerador de relatório atado. Seus relatórios são construídos de seções (faixas) com texto e componentes de gráficos neles. Estas faixas são reproduzidas de acordo com seus dados para criar o relatório final. Um relatório simples poderia incluir estes componentes colocados em uma forma:

- Dataset (TTable)
- que componente de QuickReport conectou ao dataset
- Detail faixa no componente de relatório
- texto Imprimível (TQRDBText) na faixa de detalhe

Siga estes passos para criar o relatório:

- Start um projeto novo
- Drop um componente de TTable na forma. Selecione o DBDemos pseudônimo como DatabaseName, Cliente como TableName e jogo Ativo Retificar
- Drop um componente de TQuickRep na forma principal e fixou a propriedade de DataSet para Table1
- Expand a propriedade de Faixas
- Set HasDetail True. Isto somará uma faixa de detalhe para o relatório
- Drop um componente de TQRDBText na faixa de detalhe, DataSet fixo para Table1 e DataField para Companhia

Estes são todos os componentes que você precisa criar um relatório básico.

Prever o relatório você pode corrigir faça tique-taque em algum lugar no componente de TQuickRep e Pré-estréia seletor do cardápio de popup. Se você fizesse tudo direito que você deveria ver uma janela de pré-estréia agora com seu relatório.

Tão agora você tem um relatório que trabalha em momento de designio mas você quer isto para também trabalhar a runtime. Siga estes passos ativam o relatório a runtime:

- Add um botão para sua forma principal e fixou a Legenda para ' Preview'

· Double trinco no botão para somar um evento de OnClick e somar a linha seguinte de código:

```
QuickRep1.Preview;
```

Corra sua aplicação e clique o botão de Pré-estréia para ver o relatório. Imprimir o relatório diretamente você chama o método de Impressão em vez de Pré-estréia.

Você criou um relatório de tipo de lista simples. Nos capítulos seguintes nós passaremos por muitos passos criando todos os tipos de relatórios.

#### A faixa diferente digita

É importante saber os tipos de faixa diferentes e quando os usar. Ampliando a propriedade de Faixas dos componentes de TQuickRep você somar e remove os seis a maioria tipos de faixa comuns. As faixas aparecerão no componente de TQuickRep na ordem que eles serão imprimidos finalmente.

Faixas também podem ser somadas derrubando um TQRBand da paleta de componente sobre o componente de TQuickRep. Você tem que levar então se preocupa fixar a propriedade de BandType para o propicie valor. Também é uma idéia boa para dar para o componente de faixa um nome descritivo, como PageHeaderBand, DetailBand e assim por diante. Isto é automaticamente todo acabado se você soma as faixas usando a propriedade de TQuickRep.Bands.

Além destas faixas há faixas por se agrupar criar e domina relatórios de detalhe.

#### Dados conectando

QuickReport pode recobrar dados de muitos tipos de dados. Normalmente você conectará seu relatório para um Delphi componente de DataSet, um TTable ou um TQuery. O dataset podem estar na mesma forma como o relatório, em outra forma em seu projeto ou em um Módulo de Dados. Se o dataset simplesmente é armazenado fora da forma de relatório some aquela forma ou módulo de dados para os Usos liste da forma de relatório. Então fixe a propriedade de DataSet do componente de TQuickRep para unir ao dataset.

O dataset devem ser ativos para um relatório a ser imprimido. Se não é, nenhum relatório será gerado.

#### Filtrando e Ordenando os Dados

O próprio QuickReport não tem nenhuma característica assim dados de tipo. Dados sempre está impresso na ordem apresentada a QuickReport por Delphi e o BDE. Mudar a ordem de tipo de um relatório conectada a um componente de TTable ativam um índice semelhante para a ordem de tipo que você quer. Se nenhum tal índice está disponível que você poderia considerar somando um ou poderia imprimir ao invés de uma questão ordenada.

Quando imprimindo de um TQuery você ajustam a ordem de impressão mudando a ordem de tipo na questão. Se o propicie documentação de banco de dados para informação sobre escrever declarações de SQL.

Filtrando dados podem ser feitos de vários modos. Se você imprime de um TTable e você usa Delphi 2.0 ou depois você pode usar a propriedade de TTable.Filter para somar uma condição filtrando. Se lembre de também fixar a propriedade de TTable.Filtered para Retificar.

Outra aproximação é imprimir de um componente de TQuery e usar o Onde cláusula seleccione quais registros para recobrar. Delphi lhe deixa criar parameterized examina que faz isto fácil de mudar a seleção de dados. Por favor observe isto na documentação de Delphi para mais informação.

Uma terceira opção é usar o evento de BeforePrint de sua faixa de detalhe para virar imprimindo de tempo em tempo desta faixa. Berro é um exemplo de tal um manipulador de evento:

```
procedimento TrepList.ReportDetailPrintEvent(Sender: TComponent;  
                                             PrintBand: boolean);
```

comece

```
PrintBand := CustomerTableCompany.Value> ' H';  
fim;
```

#### Texto somando e Dados o Fields

Imprimir texto em um relatório você soma componentes imprimíveis para seu relatório. Os dois componentes de texto imprimíveis básicos são TQRLabel e TQRDBText.

O componente de TQRLabel é usado para imprimir texto estático ou texto e números que são calculados por sua aplicação durante geração de relatório. Imprimir um texto estático só gota o componente em uma faixa, posicione e entre no texto que você quer na propriedade de Legenda. O texto aparecerá agora nisto é faixa de pai e sido imprimido naquela posição de parente cada tempo a própria faixa está estando impressa.

Você pode mudar o manancial e cor de manancial modificando a propriedade de Mananciais. QuickReport apóia qualquer TrueType ou mananciais de Pós-escrito.

Você pode mudar a propriedade de Legenda em todo caso os manipuladores durante geração de relatório. Deste modo você pode fazer cálculos ou operação de fio diretamente em Delphi codifique e imprima o resultado em um relatório. Podem ser achados mais informação e exemplos nisto depois neste documento.

Campos de banco de dados podem ser imprimidos usando o componente de TQRDBText. Derrube o componente em uma faixa e una a propriedade de DataSet para a mesa ou examine você quer imprimir dados de. Então selecione um campo na propriedade de DataField.

Você também pode usar o componente de TQRExpr para imprimir texto estático e dinâmico e campos de banco de dados.

#### RELATÓRIOS CRIANDO

O primeiro passo quando criando um QuickReport é criar uma forma para armazenar seu componente de TQuickRep. Nós nos referimos a esta forma como um 'Relatório Form' desde que é atos justos como um recipiente para seus relatórios e nunca é significado ser mostrado ao usuário de fim de sua aplicação. A forma de relatório é listada na Delphi Projeto Gerente e é therefor uma idéia boa para dar isto um nome que identificará isto como uma forma de relatório.

Começando todos os nomes de forma de relatório com 'rep' lhes farão se salientar. Você poderia querer fazer o mesmo com os nomes de unidade de forma.

O segundo passo é pôr um componente de TQuickRep em sua forma de relatório. Considerando que você já nomeou a forma para descrever o relatório que é uma idéia boa para dar para todos os componentes de relatório o mesmo nome. Nomeando o componente de TQuickRep simplesmente 'Report' você vai referência eles como repCustomerListing.Report, repSalesListing.Report e assim por diante.

#### Grid, Unidades e Zoom

Quando derrubando o componente de TQuickRep em uma forma você vão se um grid para agir como um guia por posicionar componentes. O grid é mostrado nas unidades de QuickReport atuais. Você pode fazer plano de relatório muito preciso posicionando e classificando segundo o tamanho componentes de QuickReport em 1/100° de uma polegada ou 1/100° de um MM. Você selecione o atualmente unidade ativa mudando a propriedade de TQuickRep.Units no inspetor de propriedade. O grid serão atualizados quando você muda esta propriedade.

Em a maioria dos monitores poderia ser difícil de ver o componente de TQuickRep inteiro desde que leva o tamanho do tamanho de papel atual selecionado. Adquirir um quadro mais claro você pode mudar a propriedade de Zoom. Mudando o zoom serão refletidos imediatamente no desenhista de forma. Esta característica também pode ser usada para zunir em em detalhes importantes por posicionar preciso e classificar segundo o tamanho.

## Tamanho de papel e margens

Você pode fazer ligação precisa de sua página ampliando a propriedade de Página do componente de TQuickRep. Trinco dobro no '+' assine à esquerda da propriedade para ampliar as propriedades de substituto. Você verá todas as colocações que controlam o plano de página agora. Os valores dados estão na unidade atualmente selecionada. O slected de margens podem ser vistos como linhas pontilhadas azuis no componente de TQuickRep. Suas faixas serão classificadas segundo o tamanho para ajustar dentro das margens.

Você pode escolher entre retrato e plano de página de paisagem e você podem virar o grid em e de com a propriedade de Regra.

PaperSize pode ser fixado a quaisquer dos valores seguintes: Carta, LetterSmall, Tablóide, Ledger, Legal, Declaração, Executivo, A3, A4, A4Small, A5, B4, B5, Fólio, Quarto, que qr10X14, qr11X17, Notam, Env9, Env10, Env11, Env12, Env14, CSheet, DSheet, e ESheet,.,.

Estes são todos os tamanhos de papel de falta definidos em Windows. Selecionando um tamanho de papel serão refletidos automaticamente no PaperLength e propriedades de PaperWidth. Você também pode fixar PaperSize para Costume e pode selecionar qualquer PaperLength e PaperWidth que você quer. Não note isso todas as impressoras apóiam todos os tamanhos de papel. Também, muitas impressoras não apóiam tamanho de papel de costume ou podem ter só tamanho de costume dentro de certos valores. Se você seleciona um tamanho de papel não apoiado por uma impressora que QuickReport trocará automaticamente ao tamanho de papel de falta quando preparando o relatório. O tamanho de papel de falta é selecionado na ligação de motorista de impressora.

Alguns motoristas de impressora definem tamanhos de papel adicionais. Estes tamanhos não podem ser fixados automaticamente nesta versão de QuickReport mas você normalmente pode criar um papel de costume com o mesmo tamanho.

Também há motoristas de impressora que não apóiam fixando os valores de tamanho de costume de aplicações. Você terá que selecionar tamanho de papel de costume no motorista de impressora e definir o papel então. Fixe ser o tamanho de papel de falta para a impressora e finalmente fixar a propriedade de TQuickRep.Page.PaperSize para Deixar de comparecer para isto. Seu tamanho de costume será escolhido agora em runtime.

### Selecionando um Manancial

Você fixou o manancial de falta para seu relatório na propriedade de TQuickRep.Font. Por dobre fazendo tique-taque na propriedade você adquirirá o Delphi manancial diálogo onde você pode selecionar nome de manancial e pode classificar segundo o tamanho, atributos de manancial fixos como tipo negrito, itálico e seleciona a cor de manancial. Os mananciais listados são todo o Windows mananciais de sistemas, tudo instalaram Verdadeiros mananciais de Tipo e qualquer manancial de Pós-escritos (se Adobe TypeManager é instalado). Você pode usar qualquer combinação de mananciais em seus relatórios mas nós conselho o uso de TrueType ou mananciais de Pós-escrito se você permitirá para o usuário prever o relatório. Os mananciais de sistemas estão só disponíveis em tamanhos limitados e não escalam muito bem em pré-estréia.

Algumas impressoras de matriz de ponto imprimem muito mais rápido se você já seleciona um manancial construa no hardware de impressora, tipicamente chamou um 'font' de impressora. Estes mananciais não são listados por QuickReport mas pode ser fixado a runtime fixando a propriedade de manancial através de código:

```
repCustomerListing.Report.Font.Name := 'TEMPOS de CG';
```

Através de falta todos os componentes no relatório herdaram o manancial que fixa do relatório. Qualquer componente que tem seu manancial explicitamente jogo ou tem sua propriedade de ParentFont fixado para Falso não herdará a colocação de manancial de relatório. Nota que você também pode anular o manancial ao nível de faixa.

O readability de seu relatório depende muito de sua seleção de manancial. Você deveria considerar isto cuidadosamente quando selecionando mananciais. Usar muitos mananciais diferentes, cores e estilos em um relatório podem fazer isto facilmente olhar atravancado e difícil ler.

#### Título de relatório e Descrição

Um relatório pode ter um título e uma descrição. Estas propriedades são principalmente por identificar e descrever o relatório em sua aplicação. Tipicamente você pode ter um sistema de cardápio listando todos seus relatórios e mostrando a descrição quando o usuário seleciona um relatório. Um exemplo disto pode ser visto no QuickReport exemplo projeto.

O título de relatório também pode ser imprimido em um relatório com o componente de TQRSysData.

#### Faixas somando

QuickReport é um gerador de relatório atado, conseqüentemente faixas são uma parte muito importante de seus relatórios. Se você é pouco conhecido com geradores de relatório atados que você pode pensar nisto como folhas pequenas de modelos de papel que você copia sobre uma página e preenche com dados. São copiados modelos diferentes em partes diferentes da página (relatório).

O modo mais fácil para somar faixas é ampliar a propriedade de TQuickRep.Bands no inspetor de propriedade. Clique no '+' assine à esquerda da propriedade para ampliar a lista de faixas comuns.

Os tipos de faixa são listados agora e o inspetor de objeto mostra se aquele tipo de faixa existe no relatório ou não. Você pode somar ou pode apagar uma faixa simplesmente mudando uma propriedade de Verdadeiro para Falso ou vice-versa. Quando uma faixa é criada deste modo adquirirá um nome de faixa descritivo, como DetailBand1, PageHeaderBand1 e assim por diante. O tipo de faixa também será fixado automaticamente.

Você também pode somar faixas selecionando o componente de TQRBand na paleta de componente e derrubando isto no relatório. Você tem que levar se preocupa fixar o tipo de faixa desejado e também deveria dar isto um nome descritivo. A propriedade de Faixas no inspetor de objeto também refletirá faixas somadas deste modo ao relatório.

A faixa digita você pode somar ao relatório que estes dois modos são:

**PageHeader** que A primeira faixa imprimiu em todas as páginas. No primeiro página imprimir é dependido da propriedade de TQuickRep.Options.FirstPageHeader. A falta é não imprimir o primeiro header de página. Intitule UMA faixa de título é a primeira faixa imprimida a um relatório (depois do muito primeiro header de página). é útil para imprimir o título de relatório, criteria de seleção de dados, data de criação e tempo e assim por diante.

**ColumnHeader** A faixa de header de coluna está impressa em cima de cada coluna de relatório. Em um único relatório de coluna regular a faixa está uma vez impressa por página e segue o header de página (e o título ata para a primeira página). Para um relatório de coluna de multi está uma vez impresso para cada coluna. É muito útil para imprimir nomes de campo.

**Detalhe** Uma faixa de detalhe está impressa para cada registro (fila) de dados em seu dataset. Esta é talvez a faixa mais importante em seu relatório e normalmente é o que leva a maioria do espaço na produção final. Você poria dados tipicamente controles imprimíveis atentos (TQRDBText e outros) nesta faixa.

**Resumo** afinal de contas faixas de detalhe estiveram impressas você pode imprimir uma faixa sumária. Isto é freqüentemente usado para imprimir totais de campos numéricos.

**PageFooter** que A última faixa imprimiu em todas as páginas. No último página imprimir é dependido da propriedade de TQuickRep.Options.LastPageFooter. A falta é não imprimir o último footer de página.

Quando você soma faixas para um relatório que eles empilham automaticamente para cima na ordem de impressão atual. Você verá que a faixa de PageHeader está em cima, seguiu pela faixa de Título, header de coluna atam e assim por diante.

As faixas terão o tipo de faixa imprimido com muito letras minúsculas nisto são mais baixo canto esquerdo. Isto é feito para ajudar identificando as faixas enquanto projetando o relatório. Este texto não será imprimido no relatório final.

Enquanto é possível somar uma faixa manualmente e fixar seu tipo para SubDetail ou GroupHeader isto não é significado ser feito. Estes atam tipos são reservados para o TQRSubDetail e componentes de TQRGroup. Fixando estes atam tipos poderiam causar o relatório para imprimir de um modo errado. Classificando segundo o tamanho as Faixas  
Faixas adquirem o tamanho horizontal deles/delas automaticamente. Para um único relatório de coluna esta é a largura de página menos as margens. O único modo para mudar uma largura de faixa é ajustar as margens.

A largura de algumas faixas também depende da propriedade de Page.Columns do componente de TQuickRep. Se você cria um relatório de coluna de multi (como um telefônico dicionário) a largura do header de coluna, detalhe, sub detalhe, header de grupo e faixas de footer de grupo ajustarão para refletir a largura disponível para uma única coluna.

Você pode ajustar a duração vertical das faixas. Isto é acabado selecionando uma faixa e resizing simplesmente isto com o rato. Se você quer controle preciso da duração de faixa que você pode ampliar a propriedade de TQRBand.Size e pode fixar um valor exato que usa a unidade de QuickReport atualmente selecionada.

Embora você pode fixar um valor pela largura de faixa no inspetor de objeto que este valor não é refletido em momento de corrida e é significado como só leu só informação.

#### Faixa que Imprime Ordem

Faixas aparecem no componente de TQuickRep na ordem de impressão atual. Porém é bom entender por que as faixas se alinham o modo que eles fazem. Geralmente faixa imprimirá na ordem seguinte:

Header de página repetiram para todas as páginas

|

Título (primeiro página)

|

Header de coluna repetiram para todas as colunas

|

Detalhe repetiu para todos os registros

|

Resumo (última página)

|

Footer de página repetiram para todas as páginas

Ficará um pouco mais complicado quando inclusive detalhes de substituto e faixas de grupo mas se você se lembra deste plano então que o resto será mais fácil.

#### Faixas virando de tempo em tempo

Você às vezes poderia querer incapacitar imprimindo de uma faixa que depende de alguma condição, ou em momento de desígnio ou em momento de corrida. Em momento de desígnio isto pode ser feito fixando a propriedade de TQRBand.Enabled para Falso. Esta propriedade também pode ser fixada a runtime, antes de o relatório seja gerado ou em todo caso o manipulador durante geração de relatório.

Durante geração de relatório você pode incapacitar imprimindo temporariamente também de uma faixa do evento de BeforePrint daquela faixa. Este evento adquire um parâmetro, PrintBand que pode ser fixado para Falso. Isto causará a faixa a ser incapacitada para aquela única instância. O da próxima vez o evento de BeforePrint é chamado para a mesma faixa você tem que fazer o cheque e tem que fixar o parâmetro novamente para incapacitar isto. Isto pode ser usado como uma característica filtrando simples:

```

procedimento TrepCusList.RepDetailBeforePrint(Sender: TQRCustomBand;
    var PrintBand: Boolean);
comece
se CustTableTotalSales > 3000000 então
    PrintBand := verdadeiro
outro
    PrintBand := falso;
fim;

```

É freqüentemente fácil esquecer que o cheque sobre pode ser feito mais rapidamente e menor gosta disto:

```

procedimento TrepCusList.RepDetailBeforePrint(Sender: TQRCustomBand;
    var PrintBand: Boolean);
comece
    PrintBand := CustTableTotalSales > 3000000;
fim;

```

Se você vira que seu footer de página atam isto terá o efeito de deixar um espaço em branco ao fundo da página em vez de encher isto de faixas de detalhe. Isto é porque QuickReport não confere a duração do footers de página todo o tempo - reduziria a velocidade muito desempenho. Se você muda a propriedade habilitada de seu footer de página que você deveria chamar o método de QuickRep.ResetPageFooterSize assim QuickReport pode atualizar sua informação de footer de página.

#### Componentes Imprimíveis somando para o Relatório

Imprimir um pouco de informação de fato em seus relatórios você soma componentes imprimíveis para você faixas. Há muitos componentes imprimíveis incluídos com QuickReport e você pode criar seu próprio ou adquire adiciona componentes se você precisa de características especiais não achadas em quaisquer dos componentes acionários. A mesa seguinte mostra os componentes imprimíveis incluídos com QuickReport:

TQRLabel Prints texto estático muito como um TLabel para formas de Delphi. A propriedade de Legenda pode ser mudada durante geração de relatório.

TQRDBText Prints campos de banco de dados, muito como um TDBText para formas de Delphi. Também imprime campos de memorando de mesas (campos de gota).

TQRExpr Evaluates e imprime qualquer expressão de QuickReport. Veja o capítulo em Expressões para explicação deste componente.

TQRSysData imprimia informação como título de relatório, número de página, data e tempo. Note que alguma desta informação também podem ser imprimidos com o componente de TQRExpr.

TQRMemo Much como um TQRLabel, a não ser que aceita linhas múltiplas de introduza texto. Normalmente usado para áreas de texto estáticas longas.

TQRRichText Prints um RichText formatou memorando. Pode ser unido a um componente de RichEdit em uma forma e pode ser imprimido os conteúdos. Só disponível na 32 versão de pedaço (Delphi 2.0 e 3.0).

Dados de TQRDBRichText versão atenta de TQRRichText. (Só disponível na 32 versão de pedaço (Delphi 2.0 e 3.0).

TQRShape Prints retângulos, círculos e linhas horizontais e verticais.

TQRImage imprimia imagens estáticas. Possa imprimir um bitmap (BMP), metafile (WMF) ou ícone.

Dados de TQRDBImage versão atenta de TQRImage. Recobra a imagem de um campo de gota, muito como um TDBImage para formas de Delphi.

#### Sobre os Componentes de Texto

Todos os QuickReport texto componentes compartilham um pouco de funcionalidade básica que eles herdam da classe de pai comum deles/delas. Se você quer mais informação sobre a classe imprimível básica e classe de texto leu o capítulo em ampliar QuickReport.

AutoSize

Componentes ou podem os classificar segundo o tamanho horizontalmente de ego para ajustar qualquer texto seja posto neles ou eles podem ter uma largura prefixada. Se a largura é prefixada que texto será cortado para ajustar o determinado retângulo.

#### WordWrap

Se AutoSize é que Verdadeiro texto pode medir linhas múltiplas é WordWrap é fixado para Retificar.

#### AutoStretch

Se WordWrap é Verdade, componentes podem ampliar verticalmente se eles estão a favor para pequeno do texto ajustar. Se um componente se expande que também ampliará sua faixa de pai, contanto que a propriedade de CanExpand de faixa seja fixada para Retificar. Uma faixa pode se expandir em cima de páginas de múltiplo para imprimir textos longos.

Note que se um componente se expande que não rebaixará outros componentes na mesma faixa. Se você tem que componentes cuja posição é dependida da duração de um texto estirando você deveriam colocar esses em uma faixa de criança. Veja o capítulo em criança para mais informação sobre isto.

#### Tamanho

Todos os componentes imprimíveis compartilham a propriedade de Tamanho. Se AutoSize é Falso que você pode usar esta propriedade para fixar o tamanho exato do componente. Também ache aqui é a posição do componente relativo para sua faixa de pai.

As unidades são determinadas nas unidades de QuickReport atualmente selecionadas e posicionar assim podem ser muito exatos.

Se AutoSize é fixado para falso que você também enlata resize e componentes de posição arrastando as manivelas de rato.

#### Estilo de manancial e Cor de Texto

Componentes de texto imprimíveis podem ter um manancial individual que fixa ou podem usar o manancial disto é band/report de pai. Usa a propriedade de manancial para anular o manancial que fixa da faixa de pai e usar a propriedade de ParentFont para fixar se o componente na verdade obterá o manancial de seu pai.

#### Alinhamento

Componentes podem permanecer ou direito alinhado ou centrou. Para componentes de AutoSized isto resultará no componente que se expande à esquerda (para direito alinhado texto), direito (para esquerda alinhado) texto ou ambos (para texto centrado).

Para componentes com AutoSize fixados para falso isto resultará no texto que é alinhado corretamente dentro de seu retângulo prefixado

#### AlignToBand

Através de componentes de falta print/align vão à posição começada o desenhista. Às vezes poderia ser mais prático alinhar componentes relativo para a faixa na qual é colocado. Você poderia querer centrar um título na página ou direito alinha um número de página para a página. Fixando componentes de texto para AlignToBand True seu alinhamento atual usará relativo a sua faixa de pai em vez de seu próprio retângulo de texto.

#### Cor de fundo e Transparência

Texto pode ser transparente (opaco) ou não. Se o texto não é transparente que a propriedade de Cor é usada para encher o fundo do componente de texto.

#### Armação

Todos os componentes de texto podem ter uma armação. Veja o capítulo em Armações para mais informação.

#### Habilitado e Incapacitou

Só gosta você pode habilitar e pode incapacitar faixas você pode habilitar e pode incapacitar componentes de texto fixando a propriedade Habilitada para Retificar ou Falso. Isto pode ser feito em momento de designio, a runtime antes de gerar o relatório ou de qualquer manipulador de evento durante geração de relatório.

```
procedimento repList.ReportDetailPrint(Sender: TObject;
    var PrintBand: boolean);
comece
{Imprima valores negativos em vermelho, positivo em verde, remova valor nulo}
se CustTableCredit.Value <0 então
    CreditLabel.Font.Color := clRed
outro
    CreditLabel.Font.Color := clGreen;
se CustTableCredit.Value = 0 então
    CreditLabel.Enabled := falso
outro
    CreditLabel.Enabled := verdadeiro;
fim;
```

#### Texto Estático imprimindo

Texto estático, como manchetes, títulos e nomes de coluna estão normalmente impressos com o componente de TQRLabel. Simplesmente derrube em uma faixa e fixe o texto na propriedade de Legenda. A etiqueta de texto se classificará segundo o tamanho, palmo linhas múltiplas e palavra envoltura que outorgam as colocações destas propriedades.

Se você quer imprimir um bloco longo de texto estático que você poderia usar o componente de TQRMemo. Este é um dados de non componente atento com uma propriedade de Linhas onde você pode inserir seu texto. Texto também pode ser carregado de um fluxo ou pode ser arquivado a runtime que usa o Lines.LoadFromStream ou métodos de Lines.LoadFromFile.

#### Banco de dados imprimindo o Fields

O modo mais fácil para imprimir um campo de uma mesa ou questão é usar o componente de TQRDBText. Age muito como um TQRLabel mas em vez de uma propriedade de legenda tem um DataSet e uma propriedade de FieldName. Estes podem ser usados para unir o componente para qualquer campo. Todo o tipo de campo apoiado pelo próprio Delphi é apoiado por este componente, com a exceção de dados binários armazenada em campos de gota.

Quando seleccionando um campo o componente exibirão CAMPO simplesmente Fieldname no desenhista. Se nenhum campo é conectado (nenhum) será exibido.

Uma característica sem igual deste componente é que pode também trabalhar com controles de dados incapacitou. TTable e TQuery tem método de DisableControls que causará dados saltado controla para deixar de atualizar. Isto é usado para aumentar velocidade de execução de programa. Considerando que QuickReport pode recobrar que os valores de campo igualam com controles incapacitados você pode ser aumentado desempenho incapacitando controles antes de gerar seus relatórios. Esteja entretanto atento - qualquer mestre detalhe relação jogo para cima com a propriedade de MasterSource do TTable/TQuery também será incapacitado e será feito este inadequado para relatórios que usam componentes de TQRSubDetail.

#### Formatando a Produção

TQRDBText usará qualquer opção formatando definida para o campo. Se nenhum é definido que imprimirá o campo simplesmente no formato de falta dado por Delphi e as colocações de Windows atuais. Date, tempo, moeda corrente e campos de ponto flutuantes serão afetados por isto.

Especificamente fixar um formatando de um uso de campo a propriedade de Máscara. A máscara trabalha diferentemente para tipos de campo diferentes.

Campos numéricos, inclusive inteiro e campo de flutuador, moeda corrente e campos de BCD é formatado usando a função de FormatFloat. São formatados data e campos de tempo usando o Delphi função de

FormatDateTime. Pode ser achada informação sobre ambas estas funções na documentação de Delphi e em ajuda de linha.

#### Campos de Memorando imprimindo

Campos de memorando estão impressos só como qualquer outro campo com o componente de TQRDBText. Você pode dar isto um espaço prefixado fixando AutoSize e AutoStretch para Falso e manualmente pode classificar segundo o tamanho o componente. Só o texto que ajusta dentro do retângulo será imprimido. Estirar um campo de memorando AutoStretch True fixo. QuickReport usará agora primeiro qualquer espaço que você prefixou, então começo que amplia o componente e o pai ata de brucos. Se não há bastante espaço na página ou coluna que o texto embrulhará ao próximo page/column. Em Delphi 1.0 campos de memorando podem estar até 64Kb grande, em Delphi 2.0 e 3.0 não há virtualmente nenhum limite à duração de texto.

#### Formas imprimindo

Formas são úteis para criar efeitos visuais e relatório bem olhando. Aparte da armação que pode ser puxada ao redor de todos os componentes e que é coberto em um capítulo separado você pode usar o componente de TQRShape para puxar algumas formas simples. Simplesmente derrube o componente em uma faixa, fixe a propriedade de Forma para Retângulo, Circule, HorzLine, VertLine, LeftAndRight ou TopAndBottom. Use a propriedade de Caneta para fixar largura de caneta e cor e a propriedade de Escova para fixar estilo de escova e cor. A versão atual de QuickReport não tem nenhuma característica para se expandir automaticamente ou alinhar uma forma para ajustar uma faixa mas isto pode ser feito manualmente em qualquer evento durante geração de relatório.

#### Imagens imprimindo

QuickReport pode imprimir ambos estático e dados imagens encadernadas que usam o TQRImage e componentes de TQRDBImage. QuickReport embrulha o Delphi TPicture classificam e podem imprimir os formatos apoiados por isto. Atualmente isso inclui bitmap arquiva (BMP), Windows Metafiles (WMF) e Ícone arquiva (ICO). imprimir outros formatos de arquivo você tem que adquirir um QuickReport que gráficos compatíveis adicionam componente.

#### Imagens estáticas

Coloque um componente de TQRImage em uma faixa e trinco dobro na propriedade de Quadro ou no próprio componente expor o Diálogo de Imagem Aberto. Localize a imagem você quer e clica o OK. Você pode usar a propriedade de Extensão para fazer a imagem se classificar segundo o tamanho para ajustar o tamanho do componente.

#### Imagens carregando a Runtime

Carregar uma imagem a runtime antes ou durante uso de geração de relatório o método de TQRImage.Picture.LoadFromFile. Observe o componente de TPicture na documentação de Delphi para mais informação sobre carregar imagens.

#### Imagens Armazenaram Em Mesas

Use o TQRDBImage para imprimir imagens armazenadas em campos de gota. Derrube o componente em uma forma e fixe o DataSet e propriedades de DataField para um campo de gota que contém quadros em um formato apoiado por TPicture. QuickReport carregará e imprimirá as imagens durante geração de relatório.

#### Nota:

Em QuickReport 1.0 você poderia usar o Delphi TImage e componentes de TDBImage para imprimir imagens em um relatório. Com QuickReport 2.0 não é apoiado já isto. A aproximação usou em QuickReport 2.0 resultados em tamanho significativamente menor de aplicações que não usam imagens em relatórios e nenhum crescimento em tamanho para aplicações que usam imagens em relatórios. A aproximação nova também é mais flexível quando criando adicionam componentes imprimíveis.

#### RichText Formatted imprimindo Memorandos

A 32 versão de pedaço de QuickReport para Delphi 2.0 e 3.0 podem imprimir RichText formatou memorandos com o TQRRichText e componentes de TQRDBRichText. TQRRichText pode ser unido a um componente de TRichEdit em uma forma em sua aplicação usando a propriedade de ParentRichEdit. Unindo

o componente resultarão no relatório que imprime qualquer texto está no pai RichEdit componente. Você também pode editar diretamente texto na propriedade de Linhas ou carga RichText de um arquivo ou pode fluir com o Lines.LoadFromStream e métodos de Lines.LoadFromFile.

Campos de RichText se expandirão e pode medir colunas múltiplas e páginas se a propriedade de AutoStretch é Verdade.

Imprimir RichText armazenaram em um uso de campo de gota o componente de TQRDBRichText e unem o propicie campo com o DataSet e propriedades de fieldName.

Note que se você exporta seus relatórios para outros formatos que usam o QuickReport exporte texto de funcionalidade dentro de componentes de RichText não será incluído na exportação.

## EXPRESSÕES USANDO

QuickReport inclui um avaliador de expressão avançado. É projetado para dar desempenho excepcional em avaliação repetida da mesma expressão durante geração de um relatório especialmente. De fato, usando o avaliador de expressão em um relatório não é muito mais lento que expressões compiladas codificadas duras. O avaliador também é flexível e pode ser ampliado facilmente com funções novas. Como uma gratificação há um construtor de expressão dialoga que pode aliviar a criação de expressões. Esta versão de usos de QuickReport o avaliador de expressão no TQRExpr e componentes de TQRGroup. Versões a chegar de QuickReport também usarão isto para dados que filtram e somarão apoio por variáveis.

Expressões compo

QuickReport expressões sintaxe está muito como Objeto expressões de Pascal. Berro é uma lista de operadores apoiados:

Operador Description

- + Some
- Subtraia
- \* Multiplique
- / Divida
- () Parênteses
- E Lógico E
- Ou Lógico OU
- Não Lógico NÃO
- = Igual
- <Menos que
- > Maior que
- <= Menos que ou igual
- >= Maior que ou igual
- <> Não igual

As funções standards incluídas com QuickReport são:

Descrição de função

- Retorno de data data atual como um fio
- Retorno de tempo tempo atual como um fio
- Str(num) Converte o argumento numérico para um fio
- Copy(str,s,l) Lucros um substring de str
- Int(num) Lucros a parte de inteiro de um número
- Frac(num) Lucros a parte fracionária de um número
- If(expr, r1, r2) Devolve para r1 ou r2 dependendo do expr de boolean
- TypeOf(expr) Lucros que os dados digitam de expr
- Sqrt(num) Lucros a raiz quadrada de um número
- Verdadeiro valor Lógico Verdadeiro
- Falso valor Lógico Falso

Sum(expr) Lucros a soma de expr  
Lucros de conta o número de entires  
Min(expr) Lucros o mais baixo valor de expr  
Max(expr) Lucros o valor mais alto de expr  
Average(expr) Médias o expr

O avaliador de expressão está trabalhando com quatro tipos de dados: Fios, Inteiro, Flutuador e Boolean. Não são apoiados dados binários e campos de memorando em expressões. Berro é uma lista de como são convertidos campos de banco de dados a QuickReport dados tipos:

Dados digitam tipo de Campo  
Campos de Fio de fio, data e campos de tempo  
Inteiro campos de SmallInt, campos de byte, campos de inteiro,  
Flutue campos de Flutuador, campos de moeda corrente,  
Boolean campos de Boolean

Campos de banco de dados usando em expressões  
Qualquer campo em qualquer referenced de mesa em seu relatório ou em um módulo de dados em sua aplicação pode ser tido acesso em umas expressões. Nomes de campo só podem ser referenced qualquer um pelo próprio nome de campo (por exemplo Nome) ou pelo nome de mesa seguido por um ponto e o campo nomeia (por exemplo Customers.Name). Se você não especifica um nome de mesa que QuickReport procurará o campo em todos os dados disponíveis fixa e usa a primeira instância achada. Dados fixa na forma de relatório será procurado primeiro, seguiu por qualquer moduls de dados.

A versão atual do avaliador de expressão não apóia nomes de campo com caráter especiais embutidos como espaço em branco, ' / ', pontilhe, sinal de dólar e assim por diante.

Fios usando em Expressões

Deveriam ser postos fios em expressão em únicas citações. O seguinte é uma expressão válida:

```
' Delphi é grande! '
```

Duração de fio de máximo é 255 caráter.

Exemplos de expressão

Berro é alguns exemplos de expressões que podem ser usadas com QuickReport:

Descrição de expressão

1 constante de inteiro, devolve 1

1.5 ponto flutuante constante, devolve 1.5

' Delphi' String constante, lucros ' Delphi' como um fio

Verdadeira constante Lógica, lucros Verdadeiro

1 + 2 cálculo numérico, devolve 3

2 \* (3 + 2.5) cálculo numérico, devolve 11

' Delphi' + ' é great' Amarram calcualtion, lucros ' Delphi é great'

Lucros de nome o valor do Nome de campo se existe

Customer.Name Returns o valor do Nome de campo na mesa de Cliente

Nome + ' ' + Contato Soma o campo de Nome, um espaço em branco e o campo de Contato

AmountPaid \* TaxRate / 100 cálculo de campo Numérico

' Impresso ' + cálculo de Fio de Data

' Quantia total pagada é ' + str(AmountPaid) cálculo de Fio

if(AmountPaid > 5000, ' order' Grande, ' order' Pequeno) Lucros ' order' Grande se AmountPaid é maior que 5000, outro ' order' Pequeno

if(CheckField, ' X', ' ') Imprime um X se CheckField é Verdade

TQRExpr usando

Imprimir expressões em um relatório você usa o componente de TQRExpr. Entre na expressão na propriedade de Expressão ou trinco dobro na propriedade expor o construtor de expressão dialogue.

O construtor de expressão lhe deixa projetar sua expressão visualmente e pode ser útil aprender criar expressões.

Use a propriedade de Máscara para formatar a produção de uma expressão. Formatar busca acabado as mesmas regras como para o componente de TQRDBText.

Se sua expressão inclui que qualquer agregação funciona, como SOMA ou CONTA você tem que unir a propriedade de Mestre para o dataset que será usado para atualizar a expressão. Para um relatório simples estes será normalmente o mesmo dataset como uniu a seu componente de TQuickRep, mas em um relatório complicado com muitos datasets que você tem que levar se preocupam unir ao correto. Uma expressão com uma função de agregação será recalculated cada tempo o ponteiro de registro do mestre unido está avançado.

A última propriedade importante para conhecer por TQRExpr é ResetAfterPrint. Se você usa uma função de agregação em uma expressão que você pode ou poderia não querer clarear a expressão depois que esteja impresso. Se não é reajustado que há pouco continuará atualizando, mas você poderia querer frequentemente pôr expressões em footers de grupo fazer cálculos em dados naquele grupo. Fixar ResetAfterPrint True farão a expressão então calcular o resultado ao invés para cada grupo se o relatório inteiro.

#### Empilhe Tamanho

A versão atual do evaluator de expressão usa o pilha de sua aplicação para passar parâmetros enquanto preparando uma expressão. Quando preparando uma expressão complexa com muitos níveis de parênteses o uso de pilha podem ser vários kilobytes. Em Delphi 1.0 o tamanho de pilha de falta é fixado a 16Kb e isto pode ser excedido facilmente e pode ser dado para um Runtime erro 202 se empilha conferindo é virado em. Aumentar o tamanho de pilha em Delphi 1.0 Opções seletas | Projeto no cardápio, vá para a página de Linker e aumente o tamanho de pilha para pelo menos 32768 bytes. Uma mais recente versão de QuickReport não usará empilhe espaço quando preparando expressões.

#### GRUPOS CRIANDO

Com QuickReport você pode criar relatórios muito simples e altamente complicados com grupos. Podem ser aninhados grupos a níveis ilimitados e se você une vários datasets em seu relatório que você pode fazer se agrupando em qualquer dataset você quer.

O componente de TQRGroup é achado na QuickReport componente paleta página e pode ser derrubado em um relatório. Quando criou aparece como uma faixa. Esta faixa será o header de grupo. Toda vez o grupo quebra (ou um grupo nivelado mais alto quebra) o esta faixa de header será imprimida. Além você também pode somar uma faixa de footer de grupo. Fazer isto, selecione o componente de TQRBand na paleta e derrube no relatório. Una a propriedade de TQRGroup.FooterBand para a faixa recentemente criada.

TQRGroup tem uma propriedade de Mestre que será fixada automaticamente ao componente de QuickReport principal. Deveria aparecer sobre qualquer faixa de detalhe no relatório. Uma faixa de footer de grupo corretamente unida deveria aparecer berro a faixa de detalhe.

A propriedade mais importante de TQRGroup exclui o Mestre é Expressão. Entre em qualquer expressão de QuickReport válida no inspetor de objeto ou exponha o evaluator de expressão para ajudar projetando a expressão. O grupo quebrará sempre que o resultado das mudanças de expressão. Considere uma mesa de cliente com um índice ativo em Estado + Companhia. Imprimir um header de grupo cada tempo o relatório chega a um estado novo você entraria em Estado ou Customer.State simplesmente como a expressão de grupo. O QuickReport expressão evaluator localizarão o campo Estatal e imprimirão o header de grupo antes de qualquer estado novo. Um footer de grupo fixo serão imprimidos quando são listados todos os clientes em um estado, útil por contar o número de clientes no estado, acumulando vendas e assim por diante.

O berro de plano de relatório é usado para criar um cliente listando:

O componente de TQuery usa a questão seguinte:

selecione \* de ordem de cliente através de estado, companhia

A expressão de grupo diz Estado simplesmente. Note os dois componentes de TQRExpr usados no relatório. O primeiro TQRExpr é colocado no header de grupo e usos a expressão seguinte:

```
if(State <> '', Declare, ' state' Desconhecido)
```

Desde então não todos os clientes têm um estado registrado esta expressão devolverá ' state' Desconhecido para esses e volta para o estado atual para os outros.

O componente de TQRExpr no footer de grupo tem a expressão seguinte:

```
' Os clientes em ' + Estado + ': ' + Str(Count)
```

Esta expressão imprimirá uma conta de cliente para cada estado. A propriedade de Mestre é unida ao componente de TQuickRep e ResetAfterPrint é fixado para Retificar assim a Conta reiniciará para cada estado. Berro é como o relatório olha em um pré-estréia:

### 3 Técnicas avançadas

#### RELATÓRIOS CRIANDO ATRAVÉS DE CÓDIGO

Em algumas aplicações você poderia precisar criar relatórios de dados não conhecido em momento de desígnio. Claro que isto pode ser feito por um usuário de fim com o editor de relatório mas poderia ser tempos quando é mais prático criar o relatório interiormente em sua aplicação. Isto poderia ser tão simples quanto deixando para um usuário selecionar qualquer mesa e há pouco poderia listar os campos na mesa ou pode ser avançado relatórios mais.

Considerando que relatórios projetaram com QuickReport é basicamente Delphi forma com componentes neles pode ser criado dynamically em sua aplicação. Você pode criar uma forma em branco, crie um componente de relatório, some faixas e componentes, conecte dados e você terá um relatório. Considerando que tudo isso pode ser um trabalho tedioso que QuickReport provê alguns mecanismos para firmar o processo e muito mais fácil.

#### O Relatório Construtor Classes

QuickReport 2 introduz a classe de TQRBuilder. Esta é uma bonita classe de abstrato que pode ser usada para criar uma armação de relatório. Isso é um componente de TQuickRep com uma faixa de título e header de página e footer.

O berro de procedimento de código usará esta classe para criar uma armação de relatório vazia. Se lembre de USAR a unidade de QREXTRA na unidade onde você tenta isto:

```
procedimento MyForm.CreateReport;  
comece  
  com TQRBuilder.Create(Self) faça  
  prova  
    Título := ' Report' Em branco;  
    Ativo := Verdadeiro;  
  prova  
    FetchReport.Preview;  
  finalmente  
    Livre;
```

```

    finalmente
      Livre;
    fim;
  fim;

```

TQRLisBuilder é uma subdivisão de classe de TQRBuilder. Soma a funcionalidade para criar um dados relatório atento:

```

procedimento MyForm.CreateDataReport;
var
  aReport: TQuickRep;
comece
  com TQRListBuilder.Create(Self) faça
    prova
      DataSet := CustomerTable; {assume CustomerTable existem}
      Titulo := ' Cliente Listing';
      Ativo := Verdadeiro;
    prova
      aReport := FetchReport;
      aReport.Preview;
    finalmente
      aReport.Free;
    fim;
  finalmente
    Livre;
  fim;
fim;

```

A função de QRCreateList faz isto até mais fácil lhe devolvendo um relatório que usa uma única linha de código:

```

procedimento MyForm.CreateDataReport;
var
  aReport: TQuickRep;
comece
  QRCreateList(aReport, Ego, CustomerTable, ' Cliente Listing', nada);
  aReport.Preview;
  aReport.Free;
fim;

```

O parâmetro de nada é de fato uma lista de fio que lista os campos que nós queremos incluir no relatório. Enviando nada causarão todos os campos que ajustaram na página a ser incluída. O berro de exemplo cria um relatório com só três campos da mesa de Cliente:

```

procedimento MyForm.CreateDataReport;
var
  aReport: TQuickRep;
  SomeFields: Tlist;
comece
  SomeFields := TList.Create;
  SomeField.Add (' Company');
  SomeFields.Add (' Contact');
  SomeFields.Add (' Phone');
  QRCreateList(aReport, Ego, CustomerTable,
    ' Cliente Listing', SomeFields);
  aReport.Preview;

```

```
aReport.Free;  
fim;
```

#### Modelos usando com classes de Construtor de Relatório

Avançar até mesmo se expandem nisto você pode passar um já componente de relatório existente para o construtor de lista classifica ou a função de QRCreateList. O relatório novo será criado então em cima dos relatórios existentes e será deixado qualquer faixa e componentes que já estão lá intatos. Deste modo você pode ter um modelo de relatório genérico com um pouco de informação básica e talvez um logotipo de companhia e QuickReport deixado constroem um relatório em cima deste modelo.

#### Indo para um mais baixo nível

Atualmente o TQRListBuilder é a única classe de construtor de relatório incluída com QuickReport. Depois versões incluirão classes para auto crie muitos outros tipos de relatórios. Porém você pode ampliar um relatório facilmente você ou até mesmo pode criar um relatório que usa os métodos do nada provido por QuickReport.

O primeiro passo é criar um componente de TQuickRep. Você pode somar as faixas então fixando as Faixas substitua propriedades:

```
procedimento MyForm.CreateReport;  
var  
  aReport: TQuickRep;  
comece  
  aReport := TQuickRep.Create(Self);  
  aReport.Parent := Ego;  
  com aReport faça  
  comece  
    DataSet := CustomerTable;  
    Bands.HasTitle := Verdadeiro;  
    Bands.HadDetail := Verdadeiro;
```

Você pode somar componentes imprimíveis agora para as faixas que usam o método de TQRBand.AddPrintable:

```
  com Bands.DetailBand1.AddPrintable(TQRDBText) faça  
  comece  
    Size.Top := 5;  
    Size.Left := 20;  
    DataSet := CustomerTable;  
    FieldName := ' Company';  
  fim;  
fim;  
aReport.Preview;  
aReport.Free  
fim;
```

AddPrintable criará um componente imprimível da classe passado a isto como um parâmetro e porá isto na faixa.

#### QRPRINTER USANDO

Se você acha que você não pode criar o tipo de printout você quer usar QuickReport diretamente mas ainda gosta da funcionalidade de pré-estréia e a habilidade para economizar e carregar um ' printout' você pode usar o objeto de QRPrinter. Cada QuickReport tem seu próprio QRPrinter privado e há um QRPrinter global definiu a startup de aplicação. Isto pode ser usado para quase qualquer tipo de impressão em vez do objeto de Impressora de ação de Delphi. Imprimindo à Tela de QRPrinter você a funcionalidade de pré-estréia de QuickReport pode usar com suas próprias rotinas de impressão.

Um Trabalho de Impressão de Amostra

QRPrinter soma alguns métodos, propriedades e eventos em cima do TPrinter standard. Produzir um printout de pré-estréia você passará pelo seguinte pisa:

- Check se QRPrinter.Status é mpReady
- Call QRPrinter.Preview para expor a janela de pré-estréia
- Call QRPrinter.BeginDoc para começar um trabalho
- Use o Delphi métodos de TCanvas para puxar e imprimir texto para o QRPrinter.Canvas
- Call QRPrinter.NewPage entre páginas
- Call QRPrinter.EndDoc quando todas as páginas são acabado

4 QuickReport se expandindo

#### SOBRE QUICKREPORT SOMAR-ONS

QuickReport é projetado para ser expansível de vários modos. Somar-ons estará disponível ambos de QSD e outro Delphi somar-em fomentador de ferramenta. Além você poderia querer escrever seu próprio somar-ons se você não pode achar um ajustando sua necessidade particular. Este capítulo cobrirá como escrever quatro tipos diferentes de QuickReport somar-ons.

Componentes imprimíveis

Embora QuickReport inclui componentes imprimíveis para muitos propósitos poderia haver situações onde você precisa ter tipos de componente adicionais. Alguns tipos de componentes incluídos com QuickReport são componentes por imprimir barra-códigos, formatos de arquivo gráficos avançados, componentes de texto especializados, etiquetas caprichosas e muito mais.

Para um componente ser impresso por QuickReport precisa descer do componente imprimível básico, TQRPrintable. Você pode criar os descendentes diretamente ou pode escrever envolturas ao redor de componentes existentes. Criando um componente imprimível inclui alguns ou tudo o seguinte pizam

- Create a funcionalidade de impressão básica do componente
- Make o componente aparece no fim usuário relatório desenhista
- Include funcionalidade de exportação

Filtros de exportação

Além do ASCII texto exportação filtro incluído com QuickReport você poderia precisar de funcionalidade para exportar relatórios para outros formatos de arquivo como Microsoft Excel, Formule, RTF ou outros formatos. QuickReport inclui os ganchos e classes básicas precisaram criar filtros de exportação para qualquer formato de arquivo.

Bibliotecas de função

O evaluator de expressão de QuickReport podem ser ampliados com bibliotecas de função que implementam quase qualquer função digita você pode imaginar. Funções matemáticas, baixo acesso de arquivo de nível, formatos de banco de dados especiais não apoiados pelo dataset de Delphi fio dirigindo modelo, avançado, cálculo de data é exemplos de bibliotecas que podem ser somadas. QuickReport inclui as classes que você precisa escrever sua própria biblioteca de função facilmente. Uma vez implementado você pode usar isto em qualquer expressão dentro de QuickReport ou outras partes de sua aplicação utilizando o QuickReport expressão evaluator.

Informe os Editores

O fim usuário relatório editor incluído com QuickReport pode ser substituído facilmente por outros editores e a construção em desenhista que funcionalidade atenta de todas as classes de QuickReport faz para criando seu próprio editor de relatório muito fácil. Basicamente QuickReport usa uma três aproximação de nível para terminar usuário relatório editando. Usando o non incluído TQRDesigner visual classifique você pode somar seu próprio editor visual sem o desgosto habitual com mensagens de Windows, API chama e baixa programação nivelada. De fato o editor em QuickReport, TQREditor, é implementado em menos de 200 linhas de código!

#### ESCREVENDO UM COMPONENTE IMPRIMÍVEL

Provavelmente os mais úteis adicionam característica de QuickReport é a habilidade para criar um componente imprimível. Usando um objeto orientado desígnio QuickReport vira o relatório que imprime

conceito um pouco de cabeça para baixo. QuickReport não sabe imprimir qualquer coisa especial. Sabe sobre imprimir algo - o TQRPrintable fundam classe. A coisa atual que está impresso terá que ser uma subdivisão de classe de TQRPrintable e tem que saber se imprimir!

São projetados todos os componentes imprimíveis acionários em QuickReport deste modo. Um componente imprimível herdará algumas propriedades da classe básica e pode somar suas próprias propriedades.

Uma Amostra Componente de CheckBox

Um exemplo pode ser um componente de checkbox que somará uma propriedade Conferida. Anulará o método de Pintura então (para aparecimento de tempo de desígnio) e o método de Impressão (para pré-estréia e imprimindo). Na Pintura e Imprime métodos puxaria um checkbox e um sinal de conferido que dependem da propriedade Conferida. A fonte para tal um componente é listada berro:

qrcb de unidade;

interface

usos classificam, quickrp;

tipo

```
TQRCheckBox = class(TQRPrintable)
```

```
privado
```

```
  FChecked: boolean;
```

```
  procedimento SetChecked(Value: boolean);
```

```
público
```

```
  constructor Create(AOwner: TComponent); anule;
```

```
  Pintura de procedimento; anule;
```

```
  procedimento Print(OfsX, OfsY.: inteiro); anule;
```

```
publicado
```

```
  propriedade Conferiu: boolean leram FChecked escrevem para SetChecked;
```

```
  fim;
```

Registro de procedimento;

implementação

```
constructor TQRCheckBox.Create(AOwner: TComponent);
```

```
comece
```

```
  Create(AOwner herdado);
```

```
  FChecked := Falso;
```

```
  fim;
```

```
procedimento TQRCheckBox.SetChecked(Value: boolean);
```

```
comece
```

```
  se Valor <> FChecked então
```

```
  comece
```

```
    FChecked := Valor;
```

```
    Invalide;
```

```
  fim;
```

```
  fim;
```

```
procedimento TQRCheckBox.Paint;
```

```
comece
```

```
  com Tela faça
```

```
  comece
```

```
    Pen.Color := clBlack;
```

```

Pen.Width := 1;
Rectangle(0, 0, largura - 1, Altura - 1);
se Conferiu então
comece
  MoveTo(0, 0);
  LineTo(Width - 1, Altura - 1);
  MoveTo(0, altura - 1);
  LineTo(Width - 1, 0);
fim;
fim;
fim;

```

```

procedimento TQRCheckBox.Print(OfsX, OfsY, : inteiro);
var
  CalcLeft,
  CalcTop,
  CalcRight,
  CalcBottom: Longint;
comece
  com ParentReport.QRPrinter faça
  comece
    Canvas.Pen.Color := clBlack;
    Canvas.Pen.Width := 1;
    CalcLeft := XPos(OfsX + Size.Left);
    CalcTop := YPos(OfsY + Size.Top);
    CalcRight := XPos(OfsX + Size.Left + Size.Width)
    CalcBottom := YPos(OfsY + Size.Top + Size.Height)
    Canvas.Rectangle(Left, Tampe, Corrija, Fundo);
    se Conferiu então
      comece
        MoveTo(Left, Topo);
        LineTo(Right, Fundo);
        MoveTo(Left, Fundo);
        LineTo(Right, Topo);
      fim;
    fim;
  fim;

```

```

Registro de procedimento;
comece
  RegisterComponents (' QReport', [TQRCheckBox]);
fim;

```

#### Dados Componentes Atentos

Dados escrevendo componentes atentos estão muito como uma mistura entre um componente imprimível standard e uns dados de Delphi standards componente atento. Porém obedecer o padrão de QuickReport de referenciando um DataSet em vez de um DataSource você quererão registrar um editor de propriedade especial para sua propriedade de fieldName provavelmente e imitarão a funcionalidade do Delphi o fieldName propriedade editor para componentes com uma propriedade de DataSource - listando todos os campos naquele DataSet. Este editor de propriedade é implementado na unidade de QREPORT desde que é um tempo de designio só característica.

#### AMPLIANDO A EXPRESSÃO EVALUATOR

A construção em evaluator de expressão pode ser ampliada por suas próprias funções. De fato, são implementadas todas as funções no evaluator de expressões como adiciona funções, não há nenhuma função codificada dura no próprio parser. Criando um adicione função requer dois passos:

- Writing uma classe de função, herdando de TQREvElementFunction,
- Registering a função

Há dois tipos básicos de adicione funções - funções simples e funções agregadas. Ambos estes funcionam tipos podem ser com ou sem parâmetros. Funções também podem aceitar um número variável de parâmetros. Exemplos nos tipos de função diferentes são:

Simple sem Tempo de parâmetros

Simple com parâmetros

EVENTO de AFTERPREVIEW PARA TQUICKREP

Declaração

propriedade AfterPreview: TQRAfterPreviewEvent

Aplica

TQuickRep

Propósito

O evento de AfterPreview é chamado sempre que um relatório foi previsto e a forma de pré-estréia está fechada. Uso típico para este evento pode ser livrar um objeto de relatório que foi previsto em um pré-estréia de modeless.

Também veja

AfterPrint

EVENTO de AFTERPRINT PARA TQUICKREP

Declaração

propriedade AfterPrint: TQRAfterPrintEvent

Aplica

TQuickRep

Propósito

O evento de AfterPrint é chamado sempre que o um relatório esteve impresso. Uso típico para este evento é livrar um objeto de relatório que esteve impresso em uma linha de fundo.

Também veja

AfterPreview

EVENTO de AFTERPRINT PARA FAIXAS

Declaração

propriedade AfterPrint(Sender: TQRCustomBand; BandPrinted: Boolean)

Aplica

TQRBand, TQRChildBand,

Propósito

AfterPrint é chamado sempre que uma faixa esteve impressa. Você pode usar o evento para fazer qualquer tipo de cálculo ou lookup requerido depois de uma certa faixa está impresso. O parâmetro de BandPrinted contará o resultado do manipulador de evento de qualquer evento de BeforePrint para a faixa.

Também veja

BeforePrint

PROPRIEDADE de ALINHAMENTO

Declaração

Alinhamento de propriedade: TAlignment (taLeftJustify)

Aplica

TQRDBText

TQRExpr

TQRLabel

TQRMemo

TQRSysData

Propósito

Alinhamento controla como texto é alinhado dentro de um controle ou uma faixa. Se AlignToBand é que Falso texto será alinhado dentro do espaço horizontal do próprio componente enquanto será alinhado dentro da página inteira ou coluna se AlignToBand é Verdade.

Significado de valor

taLeftJustify Alinham texto para lado de esquerda de controle de page/column (falta)

taRightJustify Alinham texto para corrigir lado de controle ou page/column

taCenter Centram texto dentro de controle ou page/column

Também veja

AlignToBand

PROPRIEDADE de ALIGNTOBAND

Declaração

propriedade AlignToBand: Boolean (falso)

Aplica

TQRDBText

TQRExpr

TQRLabel

TQRMemo

TQRSysData

Propósito

A propriedade de AlignToBand decide se texto será alinhado dentro de um controle isto ou dentro de sua faixa de pai. Alinhamento estará de acordo com a colocação da propriedade de Alinhamento.

Significado de valor

Falso Alinhamento será executado dentro do próprio controle (falta)

Verdadeiro Alinhamento será executado dentro da faixa de pai de controles

Também veja

Alinhamento

PROPRIEDADE de ALIGNTOBOTTOM

Declaração

propriedade AlignToBottom: boolean

Aplica

TQRBand, TQRChildBand,

Propósito

AlignToBottom True fixo se você quer imprimir qualquer faixa só sobre o footer de página em vez da colocação regular corrija a faixa prévia. A propriedade é útil quando você quer imprimir uma faixa sumária ao fundo da última página ou imprimir um resumo em uma fórmula de preprinted, i.e. um cheque.

PROPRIEDADE de ALLDATASETS

Declaração

propriedade AllDataSets: TList;

Aplica

TQuickRep

Propósito

AllDataSets é uma lista de todo o referenced de datasets em um relatório. Componentes de QRExpr podem ter acesso valores de campo de datasets incluído nesta lista. Se você quer ter acesso campos de outro datasets que você pode os somar para isto lista.

Exemplo

```
CustReport.AllDataSets.Add(Employee)
```

MÉTODO de APPLYSETTINGS

Declaração

procedimento ApplySettings

Aplica

TPrinterSettings

Propósito

ApplySettings fixará as colocações de impressora atualmente selecionadas fisicamente na impressora selecionada. O método é chamado automaticamente por QuickReport quando um relatório está sendo começado.

Também veja

TPrinterSettings

PROPRIEDADE de AUTOSIZE

#### Declaração

propriedade AutoSize: boolean (verdadeiro)

Aplica

TQRDBText

TQRExpr

TQRImage

TQRLabel

TQRMemo

TQRSysData

Propósito

Esta propriedade decide se o controle se ampliará horizontally para ajustar o texto. Para multiline (memorando) componentes ajustará a duração da linha mais longa enquanto para únicos componentes de linha (i.e. TQRLabel) isto significará a duração do texto inteiro. Note aqueles únicos componentes de linha não estirão vertically se esta propriedade é Verdade. Para AutoSize fixo classificando segundo o tamanho vertical para Falso e AutoStretch True.

Tarefas

· para fazer um columnar informar onde texto não sobreporá AutoSize fixo para Falso e manualmente fixou a largura do componente para o máximo permitiu valor.

#### Significado de valor

Verdadeiro Componente ampliará horizontally para ajustar o texto (falta)

Falso Componente não mudará largura

Também veja

AutoStretch, Altura, Classificam segundo o tamanho, Largura, WordWrap,

PROPRIEDADE de AUTOSTRETCH

#### Declaração

propriedade AutoStretch: Boolean (falso)

Aplica

TQRDBText

TQRExpr

TQRLabel

TQRMemo

Propósito

AutoStretch é usado para selecionar como componentes se comportarão se não há bastante espaço para imprimir todo o texto dentro de sua determinada altura. Se AutoStretch é Verdade que o componente se expandirá vertically um enfileiram de cada vez até todo o texto está impresso. A faixa de pai também será ampliada. Se não há bastante espaço na coluna ou na página será começado um column/page novo. Nota que um pouco de cuidado deveria ser tomado quando relatórios artificiosos com estirão controles. Para agora você só um componente estirão deveria colocar em cada faixa.

Tarefas

· para rebaixar outros componentes automaticamente quando umas extensões de componente somam um ChildBand para a faixa atual e movem estes controles para aquela faixa.

#### Significado de valor

Verdadeiro Componente ampliará vertically se não há bastante quarto para o texto

Falso Componente não ampliará vertically. Texto será cortado (falta)

Também veja

AutoSize, WordWrap,

PROPRIEDADE DISPONÍVEL

#### Declaração

propriedade Disponível: boolean

Aplica

TQuickRep

Propósito

Use a propriedade Disponível para determinar se um componente de relatório está em tal um estado que lhe permitem para chamar o Pré-estréia ou para Imprimir métodos. Disponível é Falso sempre que um

relatório está no meio de uma preparação ou imprimindo processo. Impressão chamando ou Pré-estréia enquanto Disponível é Falso não elevará uma exceção, o método há pouco devolverá sem qualquer advertência adicional.

PROPRIEDADE de BANDLIST

Declaração

Propriedade BandList: TList

Aplica

TQuickRep

Propósito

BandList é uma lista de todas as faixas incluída em um relatório. Geralmente não é nenhuma razão para você ter acesso esta lista exclui se você está escrevendo algum especial adicione componentes. As faixas são listadas na mesma ordem como eles aparecem no componente de TQuickRep.

PROPRIEDADE de FAIXAS PARA TQRSubDETAIL

Declaração

Faixas de propriedade: TQRSubDetailGroupBands

Aplica

TQRSubDetail

Propósito

A propriedade de TQRSubDetail.Bands tem duas propriedades de substituto, HasHeader e HasFooter. Eles podem ser usados criar e destruir header e footer ata para um componente de TQRSubDetail. Criando faixas com esta propriedade em vez de derrubar uma faixa da paleta de componente e unir isto manualmente você assegura que o tipo de faixa e unir é corretamente acabado. Também é um muito aproximação mais rápida.

O header e faixas de footer aparecerão no deles/delas propicie imprimindo posição no componente de relatório.

Tarefas

- para somar uma faixa de header para um jogo de QRSubDetail o Bands.HasHeader True
- para remover uma faixa de header Bands.HasHeader fixo para Falso
- para somar uma faixa de footer Bands.HasFooter True fixo
- para remover uma faixa de footer Bands.HasFooter fixo para Falso

Também veja

TQRSubDetail

PROPRIEDADE de FAIXAS

Declaração

Faixas de propriedade: TQuickRepBands (TQuickRep)

Faixas de propriedade: TQRSubDetailGroupBands (TQRSubDetail)

Aplica

TQuickRep

TQRSubDetail

Propósito

A propriedade de Faixas é usada para selecionar que tipos de faixa quer você para incluir em um relatório. Ampliando a propriedade de Faixas e fixando um das propriedades de substituto para Retificar criarão uma faixa do tipo especificado. Faixas também podem ser somadas por seleciona o componente de TQRBand na Delphi componente paleta, derruba isto em um relatório e fixa o tipo de Faixa e Nomeia mas somando isto com a propriedade de Faixas requerem muito menos clicando. Fixando esta propriedade em código você também pode somar e pode remover faixas a runtime

Exemplo

MyReport.Bands.HasDetail := Verdadeiro;

PROPRIEDADE de BANDTYPE

Declaração

propriedade BandType: TQRBandType

Aplica

TQRBand, TQRChildBand,

Propósito

BandType instrói QuickReport que papel tem uma faixa em um relatório. O tipo de faixa é automaticamente fixo quando faixas são somadas fixando uma propriedade de substituto da propriedade de TQuickRep.Bands para Retificar. Se você soma uma faixa escolhendo isto da paleta de componente você tem que levar se preocupa fixar o tipo de faixa certo.

Significado de valor

rbTitle Intitulam faixa, imprimida uma vez na primeira página do relatório, logo após o header de página se isso é habilitado na primeira página.

rbPageHeader Imprimiram em cima de todas as páginas. Imprimindo na primeira página depende da colocação da propriedade de TQuickRep.Options.FirstPageHeader.

rbDetail Imprimiram uma vez para cada registro (fila) no dataset de mestre em seu relatório.

rbPageFooter Imprimiram no fundo de cada página. Imprimindo na última página depende da colocação da propriedade de TQuickRep.Options.LastPageFooter.

rbSummary Imprimiram afinal de contas na última página faixas de detalhe.

rbGroupHeader TQRGroup usa este tipo de faixa. Também usado como uma faixa de header para um TQRSubDetail

rbGroupFooter que Footer atam para um TQRGroup ou um TQRSubDetail

rbSubDetail tipo de faixa Especial usado pelo componente de TQRSubDetail. Não fixe este tipo de faixa manualmente.

rbColumnHeader Imprimiram em cima de cada coluna em um relatório. Útil por imprimir nomes de campo em um relatório de coluna de multi.

rbOverlay não usaram em QuickReport 2. Incluiu para para trás compatibilidade

rbChild tipo de faixa Especial usado por TQRChildBand. Não fixe este tipo de faixa manualmente.

EVENTO de BEFOREPRINT DE TQUICKREP

Declaração

propriedade BeforePrint: TQRBeforePrintEvent

Aplica

TQuickRep

Propósito

O evento de BeforePrint é chamado sempre que um relatório é aproximadamente para ser gerado. Note que este testamento aconteceu ambos antes de um pré-estréia e imprime - um nome melhor para este evento provavelmente seria BeforeGenerate mas o nome atual é mantido para para trás compatibilidade. Uso típico deste evento é inicializar qualquer variável você precisa em outros manipuladores de evento durante geração de relatório.

EVENTO de BEFOREPRINT PARA FAIXAS

Declaração

propriedade BeforePrint(sender: TObject; var PrintBand: boolean)

Aplica

TQRBand

TQRChildBand

TQRSubDetail

TQRGroup

Propósito

O evento de BeforePrint é incendiado toda vez uma faixa é aproximadamente ser imprimido. Você pode executar cálculos ou legendas de jogo de componentes durante este evento. Você pode incapacitar imprimindo da faixa para a instância atual fixando o parâmetro de PrintBand para Falso.

Também veja

AfterPrint

ESCOVE PROPRIEDADE

Declaração

Escova de propriedade: TBrush

Aplica

TQRShape

Propósito

Use escova para fixar o estilo de abastecimento para componentes de TQRShape fixe para imprimir um retângulo ou círculo. Para mais informação sobre TBrush por favor veja o Delphi ajuda on-line.

Tarefas

- para imprimir uma forma transparente Brush.Style fixo para bsClear
- Set cor de abastecimento com Brush.Color

Também veja

Caneta

PROPRIEDADE de LEGENDA

Declaração

Legenda de propriedade: fio

Aplica

TQRLabel

Propósito

Legenda é o texto que será impresso pelo componente. A propriedade de Legenda pode ser mudada em eventos durante geração de relatório. Se classificará segundo o tamanho de acordo com tal muda e as colocações de AutoSize e AutoStretch.

CENTRE PROPRIEDADE

Declaração

Centro de propriedade: Boolean (Verdadeiro)

Aplica

TQRDBImage

TQRImage

Propósito

Centro é usado para decidir como uma imagem se alinhará dentro do componente.

Significado de valor

Retifique A imagem será centrada dentro da cliente se dirija

Falso A imagem será alinhada com canto esquerdo superior da área de cliente

Também veja

Extensão

PROPRIEDADE de CHILDBAND

Declaração

propriedade ChildBand: TQRCustomBand (runtime, só leia)

Aplica

TQRBand,

TQRChildBand,

TQRSubDetail,

TQRGroup

Propósito

ChildBand devolve qualquer faixa de criança que tem a faixa atual como é o mestre. Você não fixa esta propriedade diretamente mas jogo isto pela faixa de criança.

COLORA PROPRIEDADE

Declaração

Cor de propriedade: TColor

Aplica

TQRDBText

TQRExpr

TQRLabel

TQRMemo

TQRSysData

Propósito

A propriedade de Cor fixa a cor de fundo do componente.

Também veja

Transparente

Manancial

#### PROPRIEDADE de COLUMNTOPPOSITION

Declaração

propriedade ColumnTopPosition: inteiro

Aplica

TQuickRep

Propósito

ColumnTopPosition devolve a posição vertical na página onde a ColumnHeader band/Detail faixa começará impressão. O valor está em 1/10° MM e só é definido quando TQuickRep.Available é Falso. Você provavelmente terá nunca a referência este valor.

#### PROPRIEDADE de CÓPIAS

Declaração

Cópias de propriedade: inteiro

Aplica

TPrinterSettings

Propósito

Propósito

Tarefas

Tarefas

#### PROPRIEDADE de CURRENTCOLUMN

Declaração

propriedade CurrentColumn: inteiro

Aplica

TQuickRep

Propósito

CurrentColumn lhe falará que coluna do relatório está estando atualmente preparada. Em um único relatório de coluna este é sempre 1, como também é o caso durante imprimir de headers de Página e Footers e faixas de Title/summary.

Também veja

Colunas

#### PROPRIEDADE de CURRENTX

Declaração

propriedade CurrentX: inteiro

Aplica

TQuickRep

Propósito

CurrentX é o atualmente posição horizontal ativa durante geração de relatório. Esta é a posição esquerda da faixa atual. O valor está nas unidades de QuickReport nativas (1/10° MM) e só é definido enquanto

Disponível é Falso.

Também veja

CurrentY

#### PROPRIEDADE de CURRENTY

Declaração

propriedade CurrentY: inteiro;

Aplica

TQuickRep

Propósito

CurrentY é o atualmente posição vertical ativa no relatório durante geração de relatório. Esta é a posição de topo da faixa atual. O valor está no nateive unidades de QuickReport (1/10° MM) e só é definido enquanto Disponível é Falso.

Também veja

CurrentX

#### PROPRIEDADE de DADOS

Declaração

Dados de propriedade: TQRSysDataType

#### Propósito

A propriedade de dados é usada para selecionar que artigo de dados de sistema imprimirá o componente de TQRSysData.

#### Tarefas

Tarefas

#### Significado de valor

qrsColumnNo número de coluna Atual (sempre 1 para um único relatório de coluna)

qrsDate data Atual

qrsDateTime data Atual e tempo

qrsDetailCount Somam número de registros em seu dataset. Se você tem várias mesas em seu relatório que esta é a conta de registro para a mesa de mestre. Este dados sempre não está disponível quando você está usando um servidor de SQL

qrsDetailNo O número de registro atual em seu dataset. Se você tem várias mesas em seu relatório que este é o número de registro para a mesa de mestre

qrsPageNumber número de página Atual

qrsPageCount Exibe o número total de páginas no relatório. Nota que isto requer que o relatório está preparado como um dois relatório de passagem (TQuickRep.Options.TwoPass). Isto poderia levar tempo significativamente mais longo.

qrsReportTitle O título de relatório (TQuickRep.ReportTitle)

qrsTime tempo Atual

#### PROPRIEDADE de DATAFIELD

##### Declaração

propriedade DataField: fio

Aplica

TQRDBImage

TQRDBText

##### Propósito

A propriedade de DataField seleciona o campo será impresso pelo componente de TQRDBText. Entre no nome do campo ou selecione da gota abaixo lista no inspetor de propriedade. TQRDBText pode ser conectado a qualquer texto ou campos numéricos, inclusive campos de memorando. TQRDBImage deveria ser conectado a um campo de gráficos.

Também veja

DataSource, TQRDBText,

#### PROPRIEDADE de DATASET

##### Declaração

propriedade DataSet: TDataSet

Aplica

TQuickRep

TQRSubDetail

##### Propósito

A propriedade de DataSet une o relatório ou sub detalhe para um TTable ou componente de TQuery. O dataset devem ser ativos para um relatório a ser impresso.

O dataset podem estar na forma de relatório ou em um módulo de dados (Delphi 2 e 3).

Serão impressos registros na ordem de tipo provida pelo dataset. Mudar a ordem de impressão dos registros ativam um índice que emparelha a ordem desejada ou cria uma questão ordenada.

#### PROPRIEDADE de DATASET PARA TQRDBTEXT

##### Declaração

propriedade DataSet: TDataSet

Aplica

TQRDBText

TQRDBImage

##### Propósito

DataSet une o TQRDBText ou componente de TQRDBImage para uma mesa ou componente de questão. O campo a ser exibido pode ser selecionado então com a propriedade de DataField. O dataset podem estar em uma forma separada ou em um módulo de dados (Delphi 2.0 e depois).

#### PROPRIEDADE de DESCRICÃO

Declaração

Descrição de propriedade: TString

Aplica

TQuickRep

Propósito

Descrição pode ser usada para armazenar informação sobre um relatório. Um uso típico disto é exibir a descrição quando um usuário seleciona um relatório para imprimir ou prever.

Também veja

ReportTitle

#### DESENHISTA PROPERTY

Declaração

Desenhista de propriedade: TQRDesigner

Aplica

TQuickRep

Propósito

Desenhista é unido automaticamente para o atualmente objeto de desenhista ativo sempre que você expõe um relatório em um componente de QREditor. O desenhista é responsável para dirigir toda a interação entre o QuickReport e o QREditor atual. Você regularmente não terá acesso este valor.

#### PROPRIEDADE de DISPOSITIVO

Declaração

Dispositivo de propriedade: fio

Aplica

TPrinterSettings

Propósito

A propriedade de Dispositivo devolve o nome do dispositivo de impressora atualmente selecionado em um objeto de PrinterSettings.

Também veja

Motorista, Porto,

#### MOTORISTA PROPERTY

Declaração

Motorista de propriedade: fio

Aplica

TPrinterSettings

Propósito

A propriedade de Motorista devolve o nome do motorista de impressora para a impressora atualmente selecionada em um objeto de PrinterSettings.

Também veja

Dispositivo, Porto,

#### PROPRIEDADE HABILITADA

Declaração

propriedade Habilitou: boolean

Aplica

TQRBand

TQRChildBand

TQRSubDetail

TQRGroup

TQRLabel

TQRDBText

TQRExpr

Propósito

Contanto que esta propriedade seja Verdade que o componente é ativo e será imprimido no relatório. Se é Falso que o componente não será imprimido. Para faixas este resultado em todos os componentes imprimíveis na faixa que não está impresso. Habilitado pode ser mudado de eventos durante geração de relatório.

Se você muda a propriedade Habilitada de uma faixa de PageFooter ou uma faixa de criança de um PageFooter que você deveria chamar o método de TQuickRep.ResetPageFooterSize.

O evento de BeforePrint para faixas e evento de OnPrint para componentes imprimíveis pode ser usado para virar temporariamente de tempo em tempo componentes.

Também veja

BeforePrint

OnPrint

PROPRIEDADE de EXPORTFILTER

Declaração

propriedade ExportFilter: TQRExportFilter

Aplica

TQuickRep

Propósito

Quando uma classe de filtro de exportação é referenced na propriedade de ExportFilter que o filtro será ativado durante qualquer preparação de relatório. Normalmente você usará o método de ExportToFilter para exportar um relatório em vez de ter acesso este método diretamente. Qualquer nomeou filtro de exportação é destruído automaticamente quando o componente de TQuickRep é destruído.

Exemplo

Exportar um relatório para um filtro você pode usar o código seguinte:

```
MyReport.ExportFilter := TQRAsciiExportFilter.Create (' c:\report.txt ');  
MyReport.Prepare;  
MyReport.ExportFilter.Free;
```

O modo mais fácil, usar ExportToFilter se pareceriam:

```
MyReport.ExportToFilter(TQRAsciiExportFilter.Create (' c:\report.txt '));
```

Também veja

ExportToFilter

PROPRIEDADE EXPORTANDO

Declaração

propriedade Exportando: boolean (só leia)

Aplica

TQuickRep

Propósito

Se um filtro de exportação é nomeado atualmente ao ExportFilter propriedade Exportando devolverá Verdadeiro, caso contrário devolverá Falso. Exportando só é definido enquanto Disponível é Falso

Também veja

ExportFilter

MÉTODO de EXPORTTOFILTER

Declaração

procedimento ExportToFilter(AFilter: TQRExportFilter)

Aplica

TQuickRep

Propósito

Chame ExportToFilter quando você quer economizar um relatório para um arquivo externo que usa um QuickReport exportação filtro. Crie uma instância de uma classe de filtro de exportação e passe como um parâmetro para o método:

```
MyReport.ExportToFilter(TQRASCIIExportFilter.Create (' REPORT.TXT'));
```

Também veja

## ExportFilter

### PROPRIEDADE de EXPRESSÃO PARA TQREXPR

Declaração

Expressão de propriedade: fio

Aplica

TQRExpr

Propósito

Entre em qualquer expressão de QuickReport válida na propriedade de expressão. Para mais informação sobre criar expressões veja o capítulo separado neste tópico. Se lembre de fixar a propriedade de Mestre se você cria uma expressão agregada.

Também veja

Mestre

### PROPRIEDADE de EXPRESSÃO PARA TQRGROUP

Declaração

Expressão de propriedade: fio

Aplica

TQRGroup

Propósito

Entre em qualquer expressão de QuickReport válida na propriedade de expressão. O grupo quebrará sempre que o resultado das mudanças de expressão. Para mais informação veja o capítulo em criar expressões e em criar relatórios se agrupados.

### PROPRIEDADE de MANANCIAL

Declaração

Manancial de propriedade: TFont

Aplica

TQRDBText

TQRExpr

TQRLabel

TQRMemo

TQRSysData

Propósito

A propriedade de Manancial fixa as características de manancial do componente, inclusive typeface, classificam segundo o tamanho, atributos e cor. Para uma descrição cheia da propriedade de Manancial o Delphi Online vêem ajuda ou documentação impressa.

QuickReport 2.0 apóia TrueType e mananciais de impressora.

Tarefas

· para expor um diálogo de seleção de manancial à direita trinco dobro coluna desta propriedade no inspetor de propriedade.

Também veja

Colora, ParentFont, Transparente,

### PROPRIEDADE de FOOTERBAND

Declaração

propriedade FooterBand: TQRCustomBand

Aplica

TQRSubDetail

TQRGroup

Propósito

FooterBand é usado para unir um TQRSubDetail ou componente de TQRGroup para uma faixa de footer de grupo. O modo mais fácil para criar um footer de grupo para um TQRSubDetail é fixar a propriedade de Bands.HasFooter para Retificar. Para TQRGroup você tem que somar uma faixa da paleta de componente e fixar a propriedade de FooterBand para unir à faixa recentemente criada. Este testamento fixou o tipo de faixa correto (rbGroupFooter)

Footer ata para TQRSubDetail está afinal de contas impresso registros de detalhe para um único mestre está impresso.

Para um TQRGroup a faixa de footer está impressa sempre que a expressão de grupo muda ou umas mudanças de expressão de grupo niveladas mais altas.

Também veja

HeaderBand

PROPRIEDADE de FORCENEWCOLUMN

Declaração

propriedade ForceNewColumn: boolean

Aplica

TQRBand,

TQRChildBand,

TQRGroup,

TQRSubDetail

Propósito

Se você projeta um relatório de coluna de multi e sempre quer imprimir uma certa faixa em cima de um jogo de coluna novo esta propriedade para Retificar. Isto é só válido para Detalhe, header de grupo, footer de grupo e faixas de detalhe de substituto, mais claro que criança ata de quaisquer destes. Falta é Falsa.

Também veja

ForceNewPage

PROPRIEDADE de FORCENEWPAGE

Declaração

propriedade ForceNewPage: boolean

Aplica

TQRBand,

TQRChildBand,

TQRGroup,

TQRSubDetail

Propósito

Se você quer imprimir uma faixa sempre no topo de uma página nova fixe esta propriedade para Retificar. Isto é só válido para detalhe, header de grupo, footer de grupo e faixas de detalhe de substituto, mais faixas de criança de quaisquer destes. Falta é Falsa.

PROPRIEDADE de HEADERBAND

Declaração

propriedade HeaderBand: TQRCustomBand

Aplica

TQRSubDetail

Propósito

HeaderBand é usado para unir um componente de TQRSubDetail para uma faixa de header de grupo. O modo mais fácil para criar um header de grupo para um TQRSubDetail é fixar a propriedade de Bands.HasHeader para Retificar.

Faixas de Header estão impressas antes de todos os registros de detalhe para um único mestre está impresso.

Também veja

FooterBand

PROPRIEDADE de LEFTOFFSET

Declaração

propriedade LeftOffset: inteiro

Aplica

TPrinterSettings

Propósito

LeftOffset é um leia só propriedade que devolve a área não publicável no lado esquerdo do papel para a impressora atualmente selecionada em um objeto de PrinterSettings. O valor está em 1/10° de milímetros.

Também veja

TopOffset

PROPRIEDADE de LINHAS

Declaração

Linhas de propriedade: TStringList

Aplica

TQRMemo

Propósito

Todo o texto no componente de QRMemo pode ser tido acesso pela propriedade de Linhas. Os conteúdos podem ser mudados durante geração de relatório se desejou.

Tarefas

· para nomear para os conteúdos de um uso de StringList o método de Lines.Assign.

MASCARE PROPRIEDADE

Declaração

Máscara de propriedade: fio

Aplica

TQRDBText

TQRExpr

Propósito

A propriedade de Máscara formatará a produção de um dados atento de componente calculado. Se nenhuma máscara é formatando fixo será executado de acordo com a colocação dos campos propriedade de DisplayFormat ou falta que formatam para o tipo de dados respectivo.

Para TDateField, TDateTimeField, e TTimeField, formatando é executado por DateTimeToStr. Se nenhuma Máscara ou DisplayFormat não é nomeado um fio, o valor é formatado de acordo com a falta especificações de Windows no [Internacional] seção do arquivo de WIN.INI.

Para TBCDField, TCurrencyField, e TFloatField, formatando é executado por FloatToTextFmt. Se nenhuma Máscara ou DisplayFormat não é nomeado um fio, o valor é formatado de acordo com o valor da propriedade de Moeda corrente do campo.

DOMINE PROPRIEDADE PARA TQRCHILDDBAND

Declaração

Mestre de propriedade: TQRCustomBand

Aplica

TQRChildBand

Propósito

Vínculos de mestre uma faixa de criança para uma faixa de mestre. Se você une que a criança múltipla ata a um único mestre eles deveriam ser unidos em uma cadeia - uma faixa nunca deveria ter mais que uma criança que aponta diretamente a isto.

DOMINE PROPRIEDADE PARA TQREXPR

Declaração

Mestre de propriedade: TComponent

Aplica

TQRExpr

Propósito

A propriedade de Mestre é usada quando você quer criar uma expressão que agrega um campo ao longo do relatório. Um exemplo de tal uma função é SUM(AmountPaid) onde você quer imprimir a quantia total pagada todas as ordens ao fundo de um relatório. Sempre que você usa qualquer função de agregação em uma expressão que você precisará fixar o mestre.

A propriedade de Mestre conta para QuickReport quando atualizar o valor de expressão. Você pode unir o Mestre para seu componente de QuickRep ou qualquer componente de QRSubDetail em um relatório. Sempre que QuickReport avança ao próximo registro no dataset conectado ao Mestre que o valor de expressão é atualizado.

Se também

Expressão, Operação, ResetAfterPrint,  
MÉTODO de NEWCOLUMN

Declaração

procedimento NewColumn;

Aplica

TQuickRep

Propósito

Chame NewColumn de manipuladores de evento durante geração de relatório mover à próxima coluna na página. Se você está imprimindo um único relatório de coluna que o fluxo continuará na próxima página. Este método pode ser chamado seguramente do evento de BeforePrint de qualquer faixa ou evento de AfterPrint mas não deveria ser chamado de um imprimível controla evento de OnPrint ou um OnStartPage ou evento de OnEndPage.

Também veja

NewPage

MÉTODO de NEWPAGE

Declaração

procedimento NewPage

Aplica

TQuickRep

Propósito

Chame NewPage de manipuladores de evento durante geração de relatório mover à próxima página do relatório. Este método pode ser chamado seguramente do evento de BeforePrint de qualquer faixa ou evento de AfterPrint mas não deveria ser chamado de um imprimível controla evento de OnPrint ou um OnStartPage ou evento de OnEndPage.

Também veja

NewColumn

EVENTO de ONENDPAGE

Declaração

propriedade OnEndPage: TQREndPageEvent;

Aplica

TQuickRep

Propósito

OnEndPage é chamado sempre que uma página é acabado durante geração de relatório. Você pode usar este evento para atualizar ou reajustar qualquer variável usou em outro manipulador de evento ou faz qualquer tela atualiza você precisa por alguma razão especial.

Também veja

OnStartPage

EVENTO de ONNEEDDATA

Declaração

propriedade OnNeedData: procedimento (Remetente: Tobject; MoreData: boolean)

Aplica

TQuickRep,

TQRSubDetail

Propósito

OnNeedData é usado quando criando relatórios de fontes diferente de datasources de Delphi, como listas de fio, ordens, texto arquiva e quase qualquer outra fonte imaginável. OnNeedData é chamado sempre que QuickReport emitiria um comando de DataSet.Next normalmente para mover ao próximo registro de um dataset. MoreData fixando para Falso causará o relatório (ou sub detalhe) terminar.

Exemplo

Este código imprimirá todos os fios em um TStringList para em um relatório:

```
var
```

```
    CurrentItem: Inteiro;
```

```
    SomeStringList: TStringList;
```

```

procedimento TForm1.QuickRep1BeforePrint(Sender: TQuickRep;
var
  PrintReport: Boolean);
comece
  CurrentItem := 0;
  PrintReport := SomeStringList.Count > 0;
fim;

```

```

procedimento TForm1.QuickRep1NeedData(Sender: TObject; var MoreData; Boolean);
comece
  se CurrentItem < SomeStringList.Count então
    QRLabel1.Caption := SomeStringList[CurrentItem];
    Inc(CurrentItem);
    MoreData := CurrentItem <= SomeStringList.Count;
fim;

```

Para mais informação em como criar evento dirigido relatórios olham o capítulo mais cedo neste tópico nesta documentação.

#### EVENTO de ONPREVIEW

##### Declaração

propriedade OnPreview: procedimento (Remetente: TObject)

##### Aplica

TQuickRep

##### Propósito

Use o evento de OnPreview se você quer usar um pré-estréia de customized em vez do pré-estréia standard proveu por QuickReport. Se um OnPreview evento manipulador é provido que QuickReport chamará aquele evento em vez de expor o pré-estréia de falta. O parâmetro de Remetente passado ao manipulador de evento aponta a um objeto de QRPrinter que deveria ser nomeado por conta própria à propriedade de QRPrinter de um componente de TQRPreview forma de pré-estréia:

```

procedimento RepForm.MyReportOnPreviewEvent(Sender: TObject);
comece
  MyPrevForm.QRPreview.QRPrinter := QRPrinter(Sender);
  MyPrevForm.Show;
fim;

```

#### EVENTO de ONSTARTPAGE

##### Declaração

propriedade OnStartPage: procedure(Sender: TObject);

##### Aplica

TQuickRep

##### Propósito

OnStartPage é chamado sempre que uma página nova é aproximadamente ser gerado. Você pode usar isto para inicializar qualquer variável local você precisa usar durante preparação de uma única página.

##### Também veja

OnEndPage

#### PROPRIEDADE de OPÇÕES

##### Declaração

Opções de propriedade: TQuickReportOptions

##### Aplica

TQuickRep

##### Propósito

A propriedade de opções é usada para fixar algumas características de relatório. Se TQuickReportOptions para mais informação.

##### Também veja

TQuickReportOptions

## PROPRIEDADE de ORIENTAÇÃO

Declaração

Orientação de propriedade: TPrinterOrientation

Aplica

TPrinterSettings

Propósito

Orientação controla se serão imprimidas as páginas de um relatório em Paisagem ou modo de Retrato.

Você não pode trocar orientação durante um relatório. Um relatório composto deveria ter a mesma orientação para todas as seções de relatório.

Fixar esta propriedade você deveria USAR a unidade de Impressoras desde que TPrinterOrientation é declarado naquela unidade.

Significado de valor

poPortrait O relatório será impresso em Retrato (vertical) modo (falta)

poLandscape que O relatório será impresso em modo de Paisagem

## PROPRIEDADE de OUTPUTBIN

Declaração

propriedade OutputBin: TQRBin

Aplica

TPrinterSettings

Propósito

OutputBin controla que empapela caixa um relatório será produção para. Se uma impressora não apóia esta propriedade a falta (primeiro) caixa de produção será usada. Você não pode mudar caixa de produção durante um relatório.

Os valores seguintes são apoiados:

Primeiro Superior Abaixo

Envelope Manual mediano

EnvManual Auto Trator

SmallFormat LargeFormat LargeCapacity

Cassete Último

## CHAME PROPRIEDADE

Declaração

Página de propriedade: TQRPage

Aplica

TQuickRep

Propósito

A propriedade de Página é usada para fixar muitas opções para plano de página. Veja TQRPage para mais informação sobre o tópico.

Também veja

TQRPage

## PROPRIEDADE de PAGENUMBER

Declaração

propriedade PageNumber: inteiro

Aplica

TQuickRep

Propósito

PageNumber devolve o número de página no que QuickReport está trabalhando atualmente. O valor só é definido enquanto Disponível é Falso.

## PROPRIEDADE de PAPERLENGTH

#### Declaração

propriedade PaperLength: inteiro

Aplica

TPrinterSettings, TQRPage,

Propósito

PaperLength devolve a duração do tamanho de papel atualmente selecionado em 1/10º milímetros.

Fixando esta propriedade causarão a propriedade de PaperSize para ser fixado a Costume.

Também veja

PaperSize, PaperWidth,

PROPRIEDADE de PAPERSIZE

#### Declaração

propriedade PaperSize: TQRPaperSize

Aplica

TPrinterSettings, TQRPage,

Propósito

PaperSize fixa o tamanho de papel de produção preferido para um relatório. Se o tamanho de papel selecionado não é apoiado por uma impressora que o tamanho de falta será usado.

Selecionar um non listado uso de tamanho de papel o tamanho de Costume que fixa e fixou os valores físicos que usam o PaperLength e propriedades de PaperWidth.

Os valores seguintes são apoiados:

Omita Carta LetterSmall

Tablóide Ledger Legal

Declaração Executivo A3

A4 A4Small A5

B4 B5 fôlio

Quarto qr10X14 qr11X17

Note Env9 Env10

Env11 Env12 Env14

CSheet DSheet ESheet

Costume

Também veja

PaperLength, PaperWidth,

PROPRIEDADE de PAPERSIZESUPPORTED

#### Declaração

propriedade PaperSizeSupported(PaperSize: TQRPaperSize): boolean

Aplica

TPrinterSettings

Propósito

PaperSizeSupported devolverá Verdadeiro se a impressora atualmente selecionada apóia imprimindo ao tamanho de papel enviou como um parâmetro.

Também veja

PaperSize

PROPRIEDADE de PAPERWIDTH

#### Declaração

propriedade PaperWidth: inteiro

Aplica

TPrinterSettings, TQRPage,

Propósito

PaperWidth devolve a duração do tamanho de papel atualmente selecionado em 1/10º milímetros.

Fixando esta propriedade causarão a propriedade de PaperSize para ser fixado a Costume.

Também veja  
PaperLength, PaperSize,  
PROPRIEDADE de PARENTFONT

Declaração  
propriedade ParentFont: boolean (verdadeiro)

Aplica  
TQRDBText  
TQRExpr  
TQRLabel  
TQRMemo  
TQRSysData

Propósito

A propriedade de ParentFont determina onde um controle procura sua informação de manancial. Se ParentFont é Verdade, o controle usa o manancial em seu componente de pai (faixa) propriedade de Manancial. Se ParentFont é Falso, o controle usa sua própria propriedade de Manancial.

Usando ParentFont, pode assegurar você que todos os controles em um relatório têm um aparecimento uniforme.

Quando o ParentFont é Verdade para uma faixa a faixa usa o valor de seu pai QuickReport componente.

Significado de valor

Verdadeiro Obtenha informação de manancial de controle de pai (falta)

Falso Uso a própria propriedade de Manancial do componente para características de manancial fixadas

Também veja

Manancial

PROPRIEDADE de PIXELSPERX

Declaração

propriedade PixepsPerX: inteiro (só leia)

Aplica

TPrinterSettings

Propósito

PixelsPerX devolve a resolução horizontal da impressora atualmente selecionada. O valor devolvido é o número de pixels por polegada.

Também veja

PiexlsPerY

PROPRIEDADE de PIXELSPERY

Declaração

propriedade PixepsPerY: inteiro (só leia)

Aplica

TPrinterSettings

Propósito

PixelsPerY devolve a resolução vertical da impressora atualmente selecionada. O valor devolvido é o número de pixels por polegada.

Também veja

PiexlsPerX

PREPARE MÉTODO

Declaração

procedimento Prepara

Aplica

TQuickRep

Propósito

Prepare gerará um relatório sem imprimir isto ou prever isto diretamente. Exemplo de uso é se você quer gerar um relatório e salvar isto para um arquivo de QRP. O relatório gerado é armazenado no objeto de TQuickRep.QRPrinter. Quando você é acabado com o relatório você deveria livrar o QRPrinter manualmente e deveria fixar a propriedade para NADA.

Exemplo  
MyReport.Prepare;  
MyReport.QRPrinter.Save (' REPORT.QRP');  
MyReport.QRPrinter.Free;  
MyReport.QRPrinter := nada;  
Também veja  
Pré-estréia  
Impressão  
PrintBackground  
ExportToFilter

#### PREVEJA MÉTODO

Declaração  
Pré-estréia de procedimento;  
Aplica  
TQuickRep  
Propósito  
Chame Pré-estréia para expor um em pré-estréia de tela de um relatório. Através de falta QuickReport usará sua própria construção em forma de pré-estréia de padrão mas você pode anular o evento de TQuickRep.OnPreview para criar sua própria forma de pré-estréia. Pré-estréia é uma chamada modal e execução de programa não continuará até o pré-estréia está fechado.  
Também veja  
Impressão

#### IMPRIMA MÉTODO

Declaração  
Impressão de procedimento  
Aplica  
TQuickRep  
Propósito  
Chame Impressão para imprimir um relatório. O relatório será gerado diretamente à impressora atualmente selecionada.  
Exemplo  
MyReport.Print  
Também veja  
Pré-estréia

#### PROPRIEDADE de IMPRESSORA

Declaração  
Impressora de propriedade: TQRPrinter  
Aplica  
TQuickRep  
Propósito  
Isto está igual à propriedade de QRPrinter. Por favor veja aquela propriedade para mais info.

#### PROPRIEDADE de PRINTERSETTINGS

Declaração  
PrinterSettings: TQRPrinterSettings  
Aplica  
TQuickRep  
Propósito  
PrinterSettings é usado fixar o número de Cópias e virar multi chame imprimindo (dúplex) de tempo em tempo. Veja TQRPrinterSettings para mais informação.

#### MÉTODO de PRINTERSETUP

Declaração  
procedimento PrinterSetup;

Aplica

TQuickRep

Propósito

PrinterSetup exporá um diálogo de ligação de impressora que permite para o usuário selecionar impressora, número de cópias e outra impressora colocações específicas antes de imprimir um relatório. Nota que as seleções feitas no diálogo de ligação é só válida como para a instância de objeto de relatório atual.

#### MÉTODO de PRINTBACKGROUND

Declaração

procedimento PrintBackground

Aplica

TQuickRep

Propósito

PrintBackground criará uma linha de fundo para imprimir o relatório e execução de programa de retorno para a próxima linha de código. O evento de AfterPrint será chamado quando a geração de relatório é completada. Este método está só disponível na 32 versão de pedaço.

Também veja

Impressão

#### PROPRIEDADE de PRINTIFEMPTY

Declaração

propriedade PrintIfEmpty: boolean

Aplica

TQuickRep

TQRSubDetail

Propósito

PrintIfEmpty controla como o relatório (ou sub seção de detalhe) se comportará se não há nenhum registro disponível para imprimir. Se PrintIfEmpty é serão imprimidos Verdadeiro headers e footers, se Falso nada será imprimido.

#### PROPRIEDADE de QRPRINTER

Declaração

propriedade QRPrinter: TQRPrinter

Aplica

TQuickRep

Propósito

Cada componente de TQuickRep cria seu próprio QRPrinter privado objetam quando um relatório está estando preparado. Este objeto de QRPrinter local pode ser referenced pela propriedade de QRPrinter. Componentes imprimíveis imprimirão este a tela de QRPrinter. Não confunda este QRPrinter com o um QRPrinter público criou a startup de aplicação e que está disponível para uso geral em sua aplicação.

#### PROPRIEDADE de RECORDCOUNT

Declaração

propriedade RecordCount: inteiro

Aplica

TQuickRep

Propósito

RecordCount devolve o número de registros em um dataset. Se o dataset é SQL que RecordCount baseado devolve 1. Este valor é usado para calcular a posição da barra de progresso.

Também veja

RecordNumber

#### PROPRIEDADE de RECORDNUMBER

Declaração

propriedade RecordNumber: inteiro

Aplica

TQuickRep

Propósito

RecordNumber é o número de registro atual que está impresso no jogo de dados de mestre. Este valor é usado para atualizar a barra de progresso.

Também veja

RecordCount

PROPRIEDADE de REPORTTITLE

Declaração

propriedade ReportTitle: fio

Aplica

TQuickRep

Propósito

ReportTitle é usado para dar uma descrição pequena do relatório. O título de relatório aparece como o nome de trabalho de impressão no Gerente de Impressão e em qualquer trança de impressora de cadeia. Este também é o valor devolvido pelo componente de TQRSysData com Dados fixados a ReportTitle. Você também pode usar isto como um artigo de cardápio se você quer que os usuários possam selecionar entre relatórios. Dar para uma descrição longa de um uso de relatório a propriedade de Descrição

Também veja

Descrição

PROPRIEDADE de RESETAFTERPRINT

Declaração

propriedade ResetAfterPrint: Boolean

Aplica

TQRExpr

Propósito

Quando usando funções agregadas em um relatório você às vezes poderiam querer um total corrente e às vezes poderiam querer reiniciar o cálculo depois do componente esteve impresso, i.e. criar totais se agrupados. Este comportamento é controlado pela propriedade de ResetAfterPrint. Esta propriedade não tem nenhum significado se a propriedade de Mestre é em branco (nenhuma agregação é executada).

Tarefas

· para reajustar um componente de TQRExpr manualmente durante chamada de preparação de relatório o TQRExpr.Reset de qualquer evento de relatório.

Significado de valor

Retifique é reajustado O valor do componente de TQRExpr depois do valor atual esteve impresso.

Falso O valor do componente de TQRExpr nunca é reajustado.

MÉTODO de RESETPAGEFOOTERSIZE

Declaração

procedimento ResetPageFooterSize

Aplica

TQuickRep

Propósito

Se você habilita ou incapacita que qualquer footer de página ata durante geração de relatório você deveria chamar ResetPageFooterSize assim QuickReport enlata recalculate quanto espaço está disponível para dados de detalhe em uma página.

Exemplo

```
PageFooterBand1.Enabled := falso;
```

```
QuickRep1.ResetPageFooterSize;
```

PROPRIEDADE de SHOWPROGRESS

Declaração

propriedade ShowProgress: boolean

Aplica

TQuickRep

Propósito

Se ShowProgress é Verdade que uma forma de progresso aparecerá quando você chama o QuickRep.Print ou métodos de QuickRep.PrintBackground. Se ShowProgress é Falso que nenhuma forma de progresso aparecerá.

A forma de progresso tem um botão de cancelamento que o usuário pode apertar para cancelar o trabalho de relatório.

#### ESTIRE PROPRIEDADE

Declaração

Extensão de propriedade: Boolean (Falso)

Aplica

TQRDBImage

TQRImage

Propósito

Extensão seleciona como QuickReport escalará um quadro relativo para o tamanho de componente atual.

Significado de valor

Retifique A imagem será estirada para ajustar dentro da área de cliente

Falso A imagem será imprimida em seu tamanho original e eventualmente será cortada ao retângulo de cliente se muito grande ajustar dentro disto

#### PROPRIEDADE de TEXTO

Declaração

Texto de propriedade: Fio

Aplica

TQRSysData

Propósito

Serão imprimidos os conteúdos da propriedade de Texto antes ao valor atual do componente de TQRSysData. Por exemplo se Texto é ' Página ' e Dados é fixado a PageNumber o componente exibirá ' Página 1 ', ' página 2 ' e assim por diante.

#### TPRINTERSETTINGS CLASS

Unidade

QRPRNTR

Declaração

TPrinterSettings = classe

Propósito

TPrinterSettings é uma classe de QuickReport interna tinha acesso a motorista de impressora. Você pode ter acesso isto pela propriedade de TQRPrinter.PrinterSettings.

Propriedades

Cópias

Dispositivo

Motorista

Dúplex

LeftOffset

MaxExtentX

MaxExtentY

MinExtentX

MinExtentY

Orientação

OutputBin

PaperLength

PaperSize

PaperSizeSupported

PaperWidth

PixelsPerX

PixelsPerY

Porto

Impressora

Título  
TopOffset

Métodos  
ApplySettings  
Também veja  
TQRPrinter

#### TQRABOUTBOX CLASS

Unidade  
QRABOUT  
Declaração  
TQRAboutBox: TForm  
Propósito

Esta classe implementa o QuickReport sobre caixa e tem geralmente pequeno uso em uma aplicação.

Tarefas  
Mostrar o sobre chamada de caixa crie uma instância da forma e chame o método de espetáculo.

#### COMPONENTE de TQRBAND

Unidade  
QUICKRPT  
Declaração  
TQRBand = class(TQRCustomBand)  
Propósito

Faixas são alguns dos blocos de edifício principais de seus relatórios. Você derruba faixas em um componente de TQuickRep e fixou a propriedade de BandType para contar como a faixa se comportará durante geração de relatório.

Você não pode fixar largura de faixa manualmente como é calculado automaticamente por QuickReport que depende do tamanho de papel atualmente selecionado, margens e número de colunas.

Tarefas

- Use a propriedade de Manancial para fixar o manancial de pai de componentes de texto todo imprimíveis colocado na faixa
- Bands pode ser habilitado e pode ser incapacitado com a propriedade Habilitada
- Set a cor de fundo de faixa com a propriedade de Cor
- Use a propriedade de Tamanho para fixar duração de faixa exata
- Use o método de AddPrintable para somar componentes imprimíveis facilmente para uma faixa a runtime

Propriedades  
AlignToBottom  
BandType  
ChildBand

Habilitado  
ForceNewColumn  
ForceNewPage  
Armação  
HasChild  
LinkBand  
Tamanho  
Unidades  
Eventos  
AfterPrint  
BeforePrint  
Métodos  
AddPrintable  
Também veja  
TQRChildBand

## COMPONENTE de TQRCHILDBAND

Unidade

QUICKRPT

Declaração

TQRChildBand = class(TQRCustomBand)

Propósito

Faixas de criança são usadas como extensões para faixas regulares. Se você tem faixas com ampliar componentes e deseja outros componentes ser movido adequadamente você pode criar uma faixa de criança e pode pôr os componentes móveis nisto. Também é útil se você tem faixas muito longas que medirão páginas múltiplas.

Se você cria mais que uma criança para qualquer faixa que você deveria os criar como uma cadeia e deveria unir uma faixa de criança para a faixa de mestre, então outra faixa de criança para a primeira criança e assim por diante. Se um mestre tem mais que uma criança nomeou diretamente que só um deles será imprimido.

Você pode habilitar e pode incapacitar a criança individualmente. Uma faixa de criança será imprimida até mesmo se é que faixa de pai era inválida.

Some uma faixa de criança fixando para qualquer faixa propriedade de HasChild para Retificar. Isto criará um componente de TQRChildBand e fixará a propriedade de mestre corretamente. Você também pode criar isto selecionando isto na paleta de componente e derrubando isto no relatório. Você tem que levar então se preocupa fixar a propriedade de Mestre corretamente.

Propriedades

AlignToBottom

BandType

ChildBand

Habilitado

ForceNewColumn

ForceNewPage

Armação

HasChild

LinkBand

Tamanho

Unidades

Eventos

AfterPrint

BeforePrint

Métodos

AddPrintable

Também veja

TQRBand

## COMPONENTE de TQRDBIMAGE

Unidade

QRCTRLS

Declaração

TQRDBImage = class(TQRPrintable)

Propósito

TQRDBImage é usado para imprimir imagens armazenadas em binário (GOTA) campos. Qualquer formato de gráficos apoiado por Delphi pode ser imprimido.

Propriedades

Centro

DataField

DataSource

Extensão  
Também veja  
TQRImage  
COMPONENTE de TQRDBTEXT  
Unidade  
QRCTRLS  
Declaração  
TQRDBText = class(TQRCustomLabel)  
Propósito

Use um componente de TQRDBText para imprimir o valor de um campo de banco de dados. É essencialmente um dados versão atenta do TQRLabel. Qualquer tipo de campo de texto pode ser impresso, inclusive campos de Fio, vários campos numéricos, campos de data e campos de memorando. Podem ser impressos campos também calculados. Texto pode medir linhas múltiplas e páginas. Você conecta o componente para o campo de dados fixando o DataSource e propriedades de DataField. Dados regulares distintos componentes atentos trabalhos de TQRDBText até mesmo com controles de dataset incapacitados. Gerando um relatório com controles incapacitaram melhorará velocidade.

Formatar produção de numérico ou campos de data usam a propriedade de Máscara. Se nenhuma Máscara é fixa o DisplayFormat do próprio campo é usado. Se nenhum DisplayFormat é falta formatando fixo será usado. Para informação em como fixar o olhar de propriedade de Máscara para cima FloatToTextFmt e DateTimeToStr na documentação de Delphi.

Para mais informação sobre alinhamento de texto, palavra embrulhando e classificando segundo o tamanho vê o componente de TQRLabel.

Propriedades  
Alinhamento  
AlignToBand  
AutoSize  
AutoStretch  
Cor  
DataSet  
DataField  
Manancial  
Máscara  
ParentFont  
Transparente  
WordWrap

Além este componente tem todas as propriedades e eventos de todos os componentes imprimíveis.

Também veja  
TQRExpr  
COMPONENTE de TQREXPR  
Unidade  
QRCTRLS  
Declaração  
TQRExpr = class(TQRCustomLabel)  
Propósito

TQRExpr é um componente de propósito geral por imprimir campos de banco de dados, cálculos e texto estático. Introduza uma expressão de QuickReport válida na propriedade de Expressão.

Se você usa uma função de agregação em sua expressão tenha certeza você fixou a propriedade de Mestre.

Ter acesso o valor do componente de TQRExpr por acesso de código a propriedade de Valor. Esta é uma TQRExprResult registro estrutura - veja QRPRNTR.PAS para a declaração desta estrutura.

Veja o capítulo em criar expressões para mais ajuda em usar este componente.

Propriedades  
Alinhamento  
AlignToBand  
AutoSize  
AutoStretch  
Expressão  
Manancial  
Máscara  
Mestre  
ParentFont  
ResetAfterPrint  
Transparente  
WordWrap

Além este componente tem todas as propriedades e eventos de todos os componentes imprimíveis.

COMPONENTE de TQRIMAGE

Unidade  
QRCTRLS  
Declaração  
TQRImage = class(TQRPrintable)  
Propósito  
Propósito  
Tarefas  
Tarefas  
Propriedades  
AutoSize  
Centro  
Quadro  
Extensão

Além este componente tem as propriedades de todos os componentes imprimíveis.

Também veja  
TQRDBImage  
COMPONENTE de TQRLABEL

Unidade  
QRCTRLS  
Declaração  
TQRLabel = class(TQRCustomLabel)  
Propósito

O componente de TQRLabel é um controle imprimível imprimia texto de non-banco de dados estático ou outro. Entre no texto a ser exibido na propriedade de Legenda. Texto pode ser dividido em linhas de múltiplo e até mesmo páginas múltiplas.

Tarefas

- Se você quer o componente a horizontally de expand/shrink ajustar o texto AutoSize True fixo.
- para imprimir texto em um prefixe, possivelmente retângulo de multiline AutoSize fixo e AutoStretch para Falso e fixou os controles classificam segundo o tamanho à área desejada.
- para centrar texto na página ou em um Alinhamento de jogo de coluna para taCenter e AlignToBand True.
- Para texto para ampliar isto é vertically de área para ajustar texto AutoSize fixo para Falso e AutoStretch True.
- para impedir para o texto de ser jogo impresso Habilitou para Falso.

Propriedades  
Alinhamento  
AlignToBand  
AutoSize  
AutoStretch  
Legenda

Cor  
Manancial  
ParentFont  
Transparente  
WordWrap

Além este componente tem todas as propriedades e eventos de todos os componentes imprimíveis.

Também veja

TQRMemo

COMPONENTE de TQRMEMO

Unidade

QRCTRLS

Declaração

TQRMemo = class(TQRCustomLabel)

Propósito

TQRMemo é usado para imprimir uma quantia grande de texto que não vem de um campo de banco de dados. Pode ser texto estático ou você pode mudar isto durante geração de relatório. O campo pode ser fixado para ampliar vertically como precisou e pode medir páginas múltiplas então se necessário.

Propriedades

Alinhamento

AlignToBand

AutoSize

AutoStretch

Cor

Manancial

Linhas

ParentFont

Transparente

WordWrap

Além este componente tem todas as propriedades e eventos de todos os componentes imprimíveis.

Também veja

TQRLabel

COMPONENTE de TQRSHAPE

Unidade

QRCTRLS

Declaração

TQRShape = class(TQRPrintable)

Propósito

TQRShape é usado para puxar formas simples como retângulos, círculos e linhas em um relatório.

Fixe o tipo de forma que você quer na propriedade de Forma.

Fixe a cor de caneta e nomeie você quer na propriedade de Caneta.

Fixe o estilo de abastecimento e cora você quer na propriedade de Escova.

Propriedades

Escova

Caneta

Forma

Também veja

SeAlso

TQRSHAPETYPE TYPE

Unidade

QRCTRLS

Declaração

TQRShapeType = (qrsRectangle, qrsCircle, qrsVertLine, qrsHorLine, qrsTopAndBottom, qrsRightAndLeft);

Propósito

TQRShapeType é a definição de tipo da propriedade de TQRShape.Shape.

Também veja

Forma

COMPONENTE de TQRSUBDETAIL

Unidade

QUICKRPT

Declaração

TQRSubDetail = class(TQRControllerBand)

Propósito

TQRSubDetail é usado para unir datasets adicional em um relatório. Tipicamente você montará uma relação de detalhe de mestre entre mesa ou componentes de questão e criará uma relação semelhante com componentes de TQRSubDetail.

O componente é derrubado em uma forma e atos como uma faixa. Você pode criar nomeie um dataset à propriedade de DataSet ou crie um evento motorista substituto detalhe com o evento de OnNeedData.

Una a faixa para o componente de mestre com a propriedade de Mestre. Imprimindo do SubDetail serão antes ou depois da faixa de detalhe de mestre que depende da colocação da propriedade de PrintBefore. A faixa deveria aparecer na desenhista de acordo com imprimir posição no relatório final.

Se o capítulo em relatórios de detalhe de mestre para mais informação sobre criar este tipo de relatórios.

Se a descrição de TQRBand para informação geral sobre a faixa propriedades relacionadas.

Tarefas

Tarefas

Propriedades

AlignToBottom

Faixas

ChildBand

DataSet

Habilitado

FooterBand

ForceNewColumn

ForceNewPage

Armação

HasChild

HeaderBand

LinkBand

Mestre

PrintBefore

Tamanho

Eventos

AfterPrint

BeforePrint

OnNeedData

COMPONENTE de TQRSYSDATA

Unidade

QRCTRLS

Declaração

TQRSysData = class(TQRCustomLabel)

Propósito

TQRSysData é usado para imprimir informação de sistema como título de relatório, número de página atual e assim por diante. Seleccione os dados para imprimir na propriedade de Dados. Fixe qualquer texto precedendo na propriedade de Texto.

Propriedades  
Alinhamento  
AlignToBand  
AutoSize  
Dados  
Manancial  
ParentFont  
Texto  
Transparente  
COMPONENTE de TQUICKREP  
Unidade  
QUICKRPT  
Declaração  
TQuickRep = class(TQRBasePanel)

Propósito

TQuickRep é a jarda de construção na qual você constrói todos seus relatórios. É um componente visual que leva a forma do tamanho de papel atualmente selecionado. Relatórios são criados derrubando faixas e componentes imprimíveis no componente de TQuickRep e conectando isto para um dataset.

Somar faixas facilmente ampliam a propriedade de Faixas e selecionam as faixas que você quer em seu relatório. As faixas selecionadas serão criadas e serão dadas um nome descritivo. Se você soma muitos (longo) faixas para seu relatório eles poderiam desaparecer berro a extremidade do componente de TQuickRep. Trinco certo no componente e seleciona gire faixas para ver faixas escondidas.

Trinco certo no componente e Pré-estréia seletor para adquirir um pré-estréia de seu relatório em momento de desígnio (enquanto no Delphi forma desenhista).

Gota componentes imprimíveis no TQuickRep se aparecem para os imprimir a uma posição estática como um revista. Trinco certo e Faixas de Couro seletas para remover as faixas temporariamente, lhe permitindo trabalhar com um espaço em branco revestem superfície. Trinco certo novamente e seleciona Reajustado Visão de Faixa para voltar as faixas.

Chame o método de Impressão para imprimir um relatório em sua aplicação. Chame Pré-estréia para expor um pré-estréia do relatório de seu código.

Na 32 versão de pedaço (Delphi 2.0 e depois) você pode chamar PrintBackground para imprimir o relatório em uma linha de fundo.

Propriedades  
AllDataSets Available  
BandList Bands  
ColumnTopPosition CurrentColumn  
CurrentX CurrentY  
Descrição de DataSet  
Desenhista ExportFilter  
FinalPass exportando  
Emoldure HideBands  
Opções Chamam  
Impressora de PageNumber  
PrinterSettings PrintIfEmpty  
QRPrinter RecordCount  
RecordNumber ReportTitle  
RotateBands ShowProgress  
SnapToGrid State  
Unidades Zunem  
Eventos  
AfterPreview AfterPrint  
BeforePrint OnEndPage

OnNeedData OnPreview

OnStartPage

Métodos

ExportToFilter NewColumn

NewPage Prepare

Impressão de pré-estrela

PrinterSetup PrintBackground

ResetPageFooterSize

PROPRIEDADE TRANSPARENTE

Declaração

propriedade Transparente: Boolean (Verdadeiro)

Aplica

TQRDBText

TQRExpr

TQRLabel

TQRMemo

TQRSysData

Propósito

A propriedade Transparente determina se o controle pintará fundo para isto ou não. Se o fundo não é pintado que você pode ver outros componentes subjacentes pelo texto.

Significado de valor

Retifique O fundo não será pintado (falta)

Falso O fundo será pintado com a cor começada a propriedade de Cor

Também veja

Cor

AMOLDE PROPRIEDADE

Declaração

Forma de propriedade: TQRShapeType

Aplica

TQRShape

Propósito

Forma é usada para selecionar o tipo de forma para o component de TQRShape.

Significado de valor

qrsCircle

qrsHorLine

qrsRectangle

qrsRightAndLeft

qrsTopAndBottom

qrsVertLine

CLASSIFIQUE SEGUNDO O TAMANHO PROPRIEDADE PARA FAIXAS

Declaração

Tamanho de propriedade: TQRBandSize

Aplica

TQRBand,

TQRChildBand,

TQRSubDetail,

TQRGroup

Propósito

Tamanho é usado para fixar duração exata de uma faixa. Também informará a largura atual da faixa mas a largura não pode ser fixada. Os valores são informados e são fixados usando as unidades de QuickReport atualmente selecionadas. Ate duração também pode ser fixada selecionando a faixa no desenhista e arrastando o topo ou manivelas de agarramento de fundo ou fixando a propriedade de Altura para o componente.

PROPRIEDADE de UNIDADES

Declaração

Unidades de propriedade: TQRUnits

Aplica

TQuickRep

Propósito

A propriedade de Unidades decide que unidades de medida trabalha QuickReport com. Os valores de todas as propriedades de Tamanho, como também a propriedade de TQuickRep.Page reflete a unidade atualmente selecionada. Quando esta mudança de propriedade todos os componentes atualizam para refletir a unidade recentemente selecionada.

Significado de valor

Valores de polegadas são determinados em polegadas.

Valores de MM são determinados em Milímetros

Valores de Pixels são determinados em unidades de Pixel para a resolução de tela atual

Valores nativos são determinados no formato de QuickReport interno que é 1/10° MM

Unidades de caráter são determinadas em unidades de Caráter e refletem o QuickRep.Font.Size atualmente selecionado (Nota: esta colocação não está trabalhando corretamente no lançamento atual)

PROPRIEDADE de WORDWRAP

Declaração

propriedade WordWrap: Boolean (Verdadeiro)

Aplica

TQRDBText

TQRExpr

TQRLabel

TQRMemo

Propósito

WordWrap decide se um componente embrulhará que texto enfileira para ajustar dentro de uma duração de linha de predefined ou não. Um componente imprimirá todo o texto

A propriedade de WordWrap determina se texto em uma etiqueta ou memorando embrulha à margem certa de forma que isto ajusta no componente. Você pode fazer o componente ampliar vertically para dar texto de impressão que não ajusta na altura prefixada fixando AutoStretch True. WordWrap não tem nenhum efeito se AutoSize é Verdade.

Significado de valor

Retifique O componente embrulhará texto à margem certa (falta)

Falso Texto será cortado a margem certa

Também veja

AutoSize, AutoStretch, Tamanho,

ZUNA PROPRIEDADE

Declaração

Zoom de propriedade: inteiro

Aplica

TQuickRep

Propósito

Fazer isto mais fácil de ver o relatório inteiro em tela enquanto o projetando podem zunir fora com a propriedade de zoom. Uma colocação de 100 lhe dará uma 1 a 1 representação do relatório, 50 lhe darão uma 50% representação de tamanho e 150 zunirão em 50% no relatório. O valor pode ser fixado a qualquer coisa entre 1 e 300. Fixando o Zoom não tem nenhum efeito no tamanho de um impresso ou previu relatório.

2 apêndices

Um - QUICKREPORT CLASSE HIERARQUIA

Berro é uma lista alfabética que todas as classes acharam nas seções de interface da fonte de QuickReport.

Algumas destas classes são significados como classes internas e não são documentados no guia de referência.

EQRError

TControlHandle

- TPrinterSettings
- TQRAboutBox
- TQRBasePanel
  - TQRCustomBand
    - TQRBand
    - TQRChildBand
    - TQRControllerBand
      - TQRSubDetail
      - TQRDetailLink
  - TQRGroup
- TQuickRep
  - TQuickReport
- TQRBuilder
  - TQRListBuilder
- TQRCompEd
- TQRCompositReport
- TQRCompress
- TQRController
- TQRCreateReportThread
- TQRDataSetup
- TQRDesigner
- TQREditor
- TQREvElement
  - TQREvElementArgumentEnd
  - TQREvElementDataField
  - TQREvElementError
  - TQREvElementFunction
- TQREvResultClass
- TQREvaluator
- TQRExpert
- TQRExportFilter
  - TQRAsciiExportFilter
  - TQRCSVExportFilter
  - TQRHtmlExportFilter
- TQRExportFilterLibrary
- TQRExportFilterLibraryEntry
- TQRExprBuilder
- TQRFiFo
- TQRFrame
- TQRGauge
- TQRGauge
- TQRLibrary
  - TQRFunctionLibrary
- TQRLibraryEntry
- TQRPageList
- TQRPreview
- TQRPreviewImage
- TQRPrintable
  - TQRCustomLabel
  - TQRDBText
  - TQRExpr
    - TQRDBCalc
  - TQRLabel
  - TQRMemo
  - TQRSysData
  - TQRCustomRichText

- TQRDBRichText
- TQRRichText
- TQRDBImage
- TQRImage
- TQRShape
- TQRPrinter
- TQRPrinterSettings
  - TQuickRepPrinterSettings
- TQRProgressForm
- TQRRichEdit
- TQRStandardPreview
- TQRStream
- TQRSubDetailGroupBands
- TQRTableSelect
- TQRUnitBase
  - TQRBandSize
  - TQRPage
  - TQRPrintableSize
- TQuickRepBands
- TTextCont
- TTextRec
- TToolbar

#### B - DESCRIÇÕES de ARQUIVO

Berro é um listando e descrição de todos os arquivos no QuickReport FONTE diretório:

QRPRNTR.PAS Baixa unidade de impressora nivelada que implementa o TQRPrinter classifica, classes de pré-estréia, evaluator de expressão e classe de ligação de impressora

QUICKRPT.PAS unidade de QuickReport Principal que implementa TQuickRep, TQRBand, TQRSubDetail e TQRPrintable. Toda a lógica informando está nesta unidade

QRCTRLS.PAS Todos os componentes imprimíveis standards

QREXTRA.PAS Report construtores, filtros de exportação, QREditor,

QREPORT.PAS Inscrição unidade com todos os editores de propriedade

QRPREV.PAS Padrão pré-estréia forma

QRPREV.DFM Form arquivo para pré-estréia standard

QRALIAS.PAS Dialog para selecionar um pseudônimo

QRALIAS.DFM Form arquivo para diálogo de seleção de pseudônimo

QRCOMPED.PAS o QuickReport componente editor dialoga

QRCOMPED.DFM Form arquivo para QuickReport componente editor

QREXPBLD.PAS Expressão construtor dialoga

QREXPBLD.DFM Form arquivo para construtor de expressão

QREXPBLD.DPR Report arquivo de projeto especialista

QREXPDLG.PAS Report o perito dialoga e unidade de implementação

QREXPDLG.DFM Form arquivo para perito de relatório

QRDATASU.PAS Dados ligação diálogo para o componente de TQREditor

QRDATASU.DFM Form arquivo para diálogo de ligação de dados

QRHTML.PAS HTML e CSW exportam unidade de filtro

QRNEW.PAS Delphi 3.0 unidade que implementa o 'container' de QuickReport

QR2CONST.PAS Identifier declaração unidade para QR2CONST.RC

QR2CONST.RC fonte de recursos de fio

QUICKRPT.RC Bitmap recursos fonte

QREPORT.RC Componente paleta bitmap fonte

#### C - CONVERTENDO DE QUICKREPORT 1

Há muitos de diferenças entre QuickReport 1 e 2. A mudança mais notável é o fato que o próprio componente de QuickReport é agora uma superfície onde você desenha para seu relatório. Em QuickReport 1 você projetou o relatório diretamente na superfície de forma.

A maioria dos componentes convertem automaticamente, porém alguma interação manual é precisada converter relatórios prosperamente.

1. faça um auxilio de seu projeto antes de tentar qualquer conversão
2. aberto qualquer relatório DFM arquiva e procura a propriedade de MESTRE (de TQRDetailLink). Remova essas linhas
3. aberto o relatório (ou troca para formar visão em Delphi 2)
4. mova qualquer componente de faixas de rbSubDetail velhas para o componente de TQRSubDetail automaticamente criado (que é uma faixa). Apague a faixa velha.
5. mova qualquer componente de um header de grupo velho ate ao componente de TQRGroup automaticamente criado (que é uma faixa). Apague a faixa velha.
6. exceto o relatório. Resposta sim fixar qualquer definição de componente incorreta e sim remover alguma conexão para manipuladores de evento que mudaram. Escreva abaixo qual os manipuladores de evento você precisa atualizar (se qualquer)
7. atualize os manipuladores de evento. O modo mais fácil para fazer isto é recriar há pouco o evento (por dobre fazendo tique-taque no inspetor de objeto) e faz um cut/paste do bloco de código velho. Deixe o toco de evento velho vazio e Delphi removerá isto para você.
8. exceto o relatório novamente.
9. trinco certo para fazer um pré-estréia de tempo de desígnio.
10. Repita passos 2 - 9 para todos os relatórios.