



Publicado por eduardorochoa : Quinta, Outubro 23, 2003 - 12:03 AM GMT-3 (230 leituras internas) Continuação... (16621 bytes) comentar  

Todos nós já se deparamos com uma situação parecida como essa:



Imagine que você tenha um sistema, onde exista um cadastro de produtos e de serviços e você armazene os dados na mesma tabela, apenas identificando-os

com um campo chamado por exemplo **gênero**, onde **"P"** seria **produto** e **"S"**, seria **serviço**.

Então seu cliente pede a você 2 relatórios simples de **Produtos** e **Serviços**, com os seguintes campos :

Relatório de Produtos:

Descrição, Grupo, Marca, Valor

Relatórios de Serviços:

Descrição, Grupo, Valor

Perceba que a diferença entre os dois relatórios está apenas no campo **marca**, este campo só será exibido quando for Relatório de Produtos.

Pense Bem...

Os dois relatórios são 99% iguais, pois utilizam a mesma tabela, terão o mesmo cabeçalho, o mesmo rodapé, os mesmos filtros, em fim...a única diferença é que, no caso de **serviços**, a **marca não** será exibida e obviamente que cada relatório terá seu filtro no campo **gênero** para que lista apenas produtos ou apenas serviços.

Nossa...

Ter que criar 2 relatórios para este caso...ou ter que ficar ajustando left/right/visible dos componentes...que saco...:-)

Calma, relaxa...temos uma grande solução para isso... :-)

É agora que venho falar sobre o componente **DataMirrorSection** :

Este componente tem um objetivo muito simples:

Exibir "**determinada coisa**" se um "**determinado campo**" for um "**determinado valor**", onde tudo isso é configurável.

No nosso caso, usaríamos este componente configurando-o da seguinte forma:

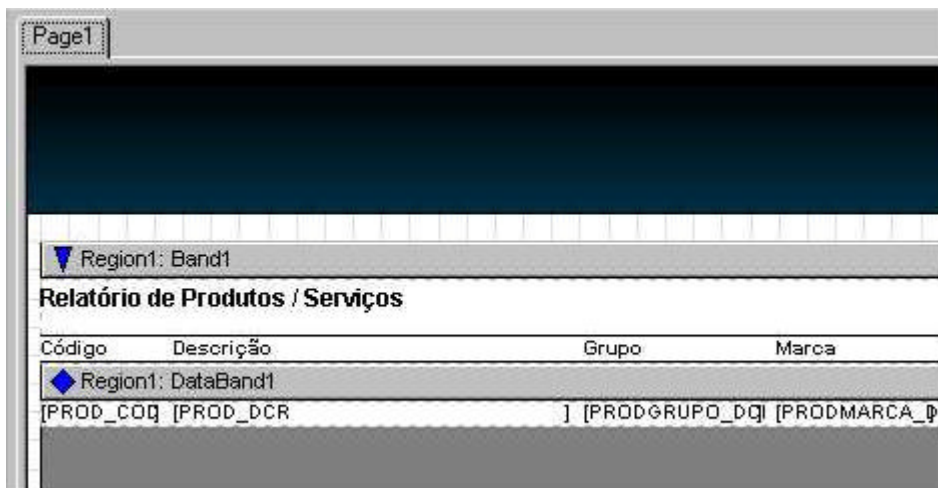
Caso o **parâmetro gênero do relatório** seja **"P"**(produto), exiba o **layout de produto**, caso seja **"S"**(serviço), exiba o **layout de serviço**.

Ops....ficou confuso ???

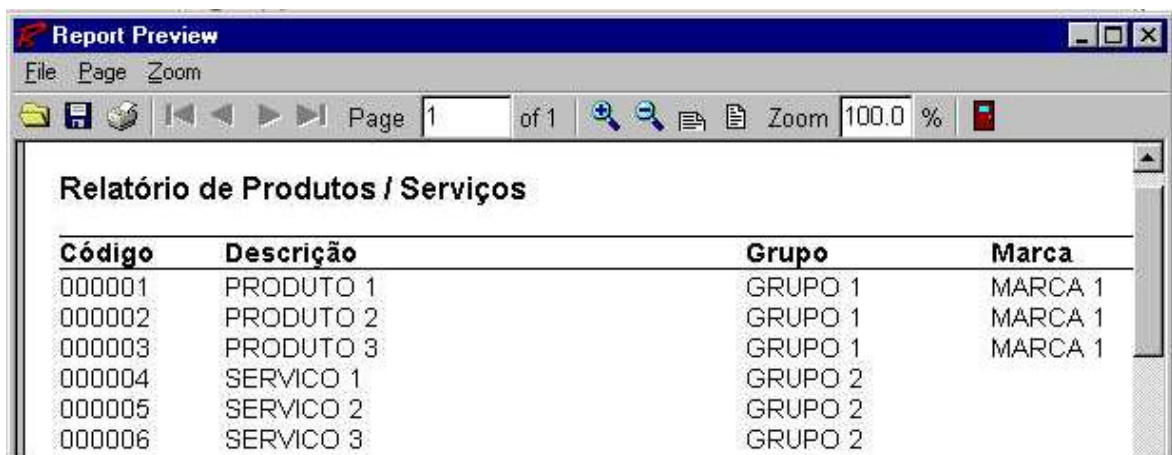
Calma, agora vamos por a mão na massa para ver como funciona esse componente:

Para que o artigo não ficasse muito extenso, resolvi não entrar em detalhes sobre a criação do relatório no que diz respeito a conexão, dataview, cabeçalho, etc., pois o objetivo deste artigo é apenas exemplificar passo a passo o uso do componente **DataMirrorSection**.

Abaixo você pode ver que criei um relatório simples de Produtos e Serviços, sem ainda me preocupar com o layout quando for Produto e quando for Serviço:



Veja o Preview:



Bom, até aqui não temos segredos, pois é um relatório simples, está sendo exibido todos os campos, independentemente(nossa, que palavra.. :-)... do gênero.

Agora iremos utilizar o **DataMirrorSection**...

Antes, deixa eu fazer um breve comentário sobre o componente **Section**:

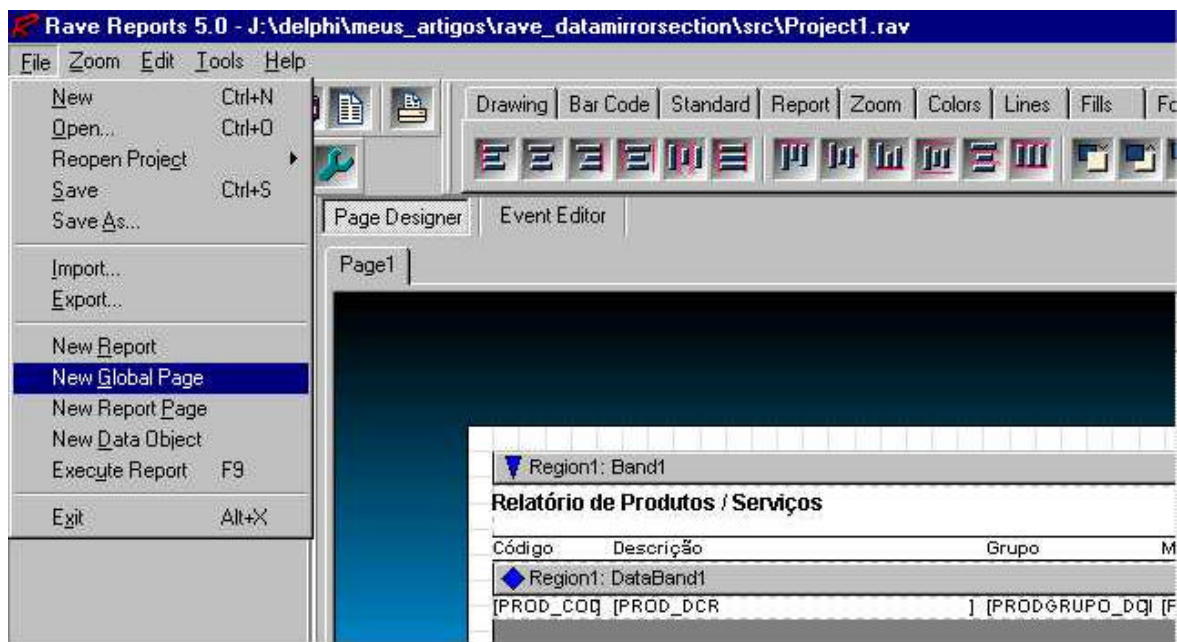
Este componente nada mais é do que um container, como um TPanel do Delphi, então ao utilizarmos ele, não se assuste, não tem segredos, o **DataMirrorSection** exige que os dados a serem exibidos, estejam no componente **Section**.

Então chega de papo e vamos lá...

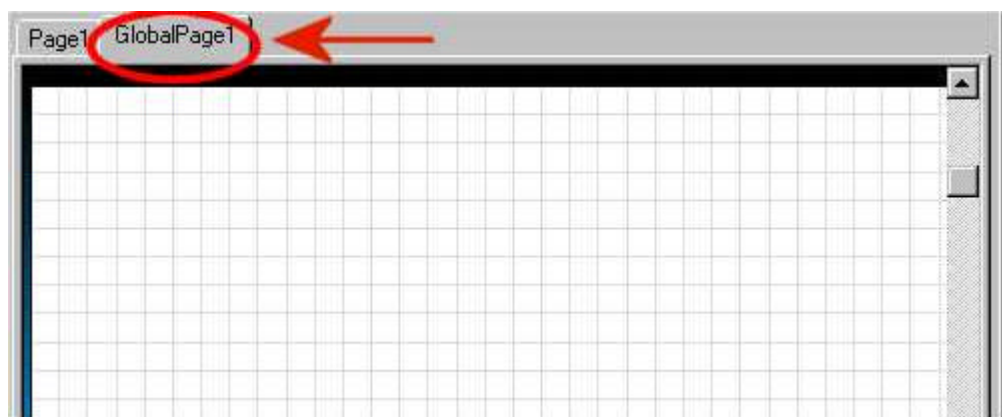
Vá no menu File e clique em **New Global Page**.

Estamos criando uma **página global**, o porque disto?

Tudo que é global em nosso relatório, colocamos neste página, pois ela é que como se fosse um **"Object Repository"** do Delphi. Então é nesta **"coisa global"** que colocaremos nossos layout's que serão exibidos no relatório.

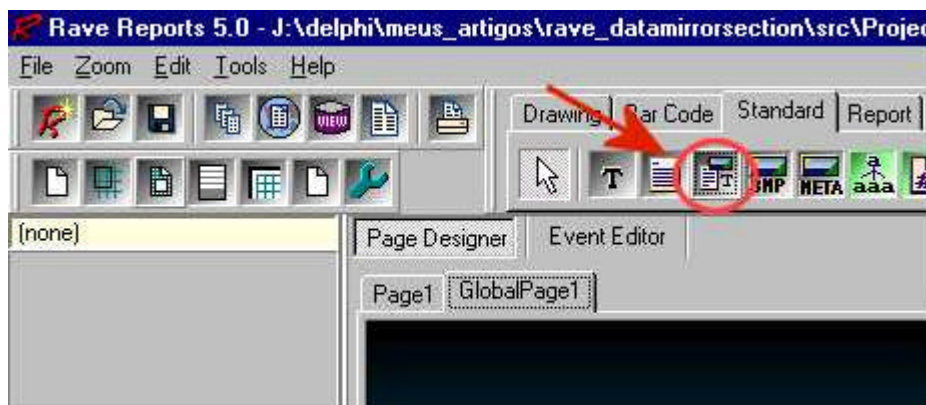


Feito isso, perceba que será criado uma palheta chamada **Global Page1**

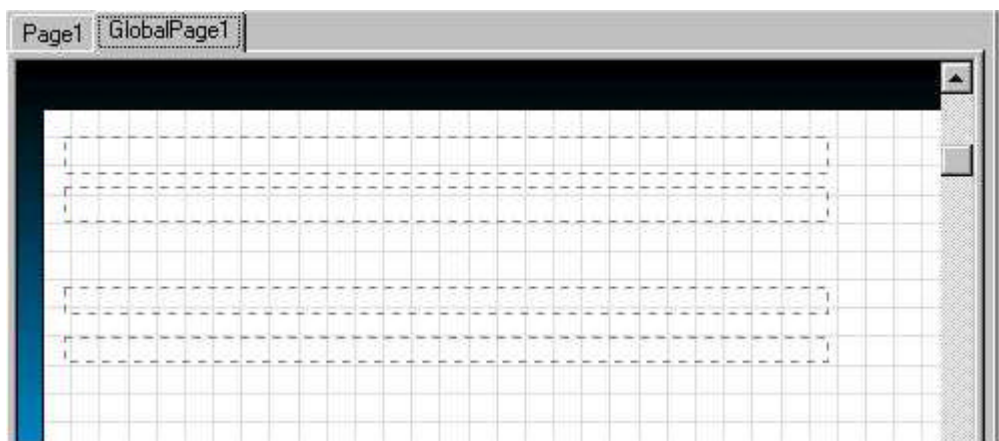


É ai dentro, nesta página que colocaremos nossos objetos globais.

Agora vá na palheta **Standard** do Rave e selecione o componente **Section**.



Coloque 4 componentes **Section** na Global Page:



Ajuste os nomes das **Sections** para que fique mais fácil depois identificá-las:

Section 1 : **sctTitProd**

Layout dos **Títulos** para **Produtos**

Section 2 : **sctTitServ**

Layout dos **Títulos** para **Serviços**

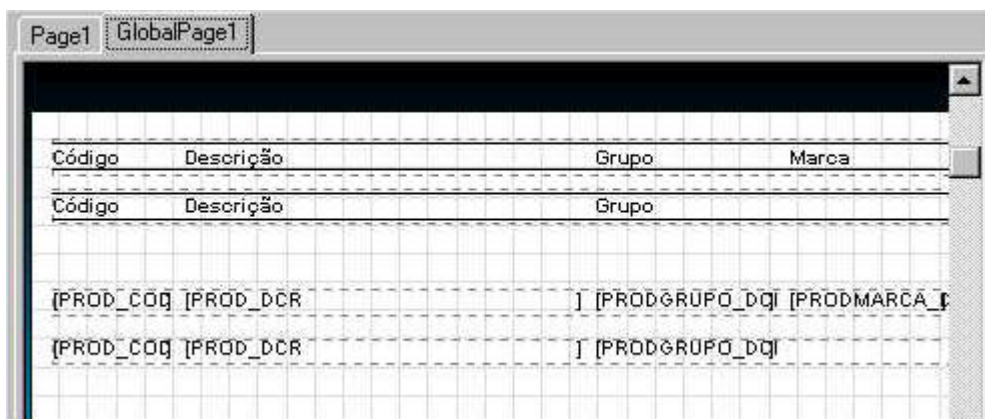
Section 3 : **sctRegProd**

Layout dos **Registros** para **Produtos**

Section 4 : **sctRegServ**

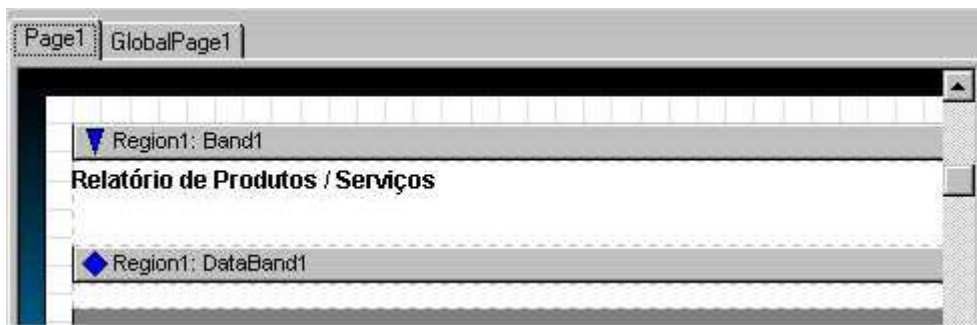
Layout dos **Registros** para **Serviços**

Vamos agora montar os layout's, colocando os componentes cada um dentro da sua respectiva **Section**, veja como ficará:



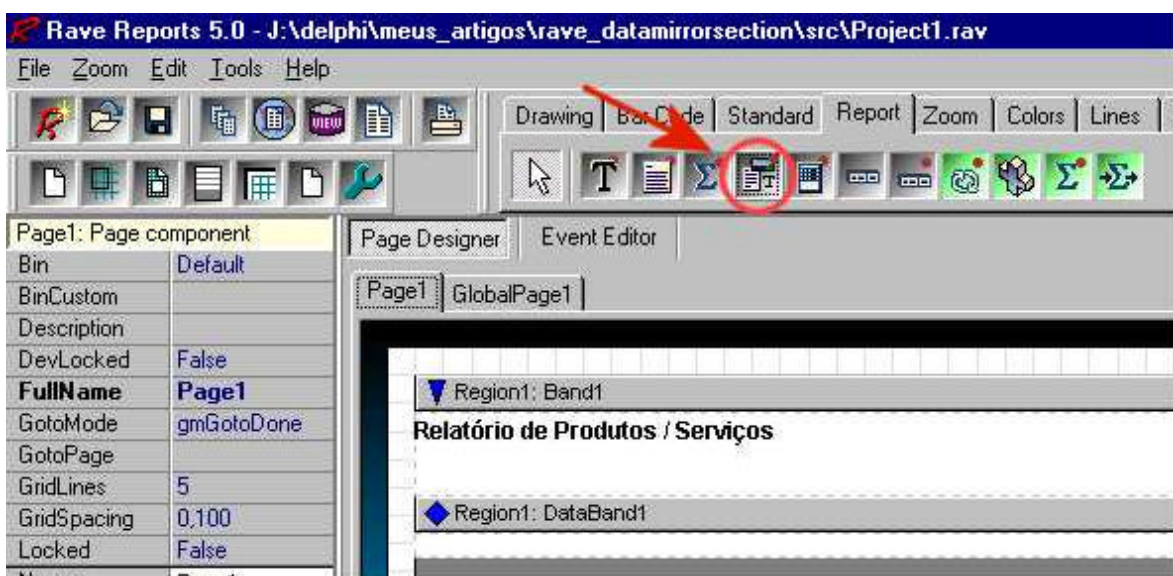
Perceba que a **Section 1 e 3** estão no layout que será usado para **Produtos**, pois contém o campo **marca**, enquanto que a **Section 2 e 4** estão no layout que será usado para **Serviços**.

Agora volte para a página onde está o relatório e arranque os componente que estavam lá, deixando-o da seguinte forma :

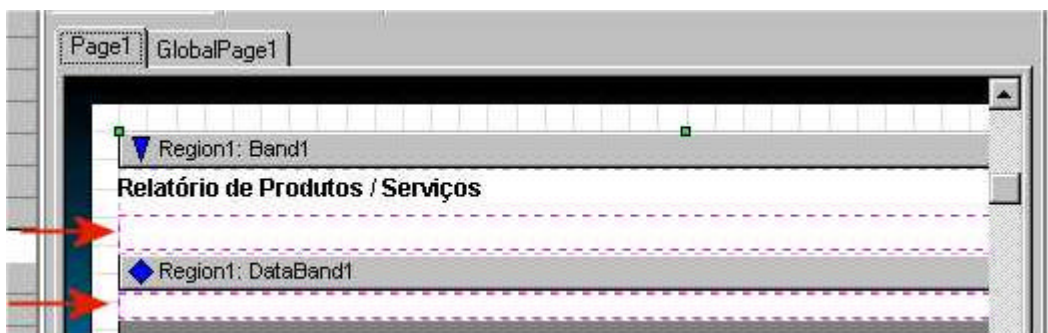


Perceba que não temos mais o layout definido nesta parte, o que usaremos agora é o famoso **DataMirrorSection**.

Vá na palheta **Report** do Rave e selecione o componente **DataMirrorSection**:



Depois disso, adicione um **DataMirrorSection** em **Band1**, que será usado para exibir os **títulos** das colunas e adicione **mais um DataMirrorSection** em **DataBand1**, que será usado para exibir os **registros**.



Vamos agora criar um **parâmetro** no nosso relatório, onde este parâmetro determinará se é um relatório de **Produto** ou **Serviço**.

Este parâmetro irá se chamar **Genero**, onde poderá receber um dos dois valores:

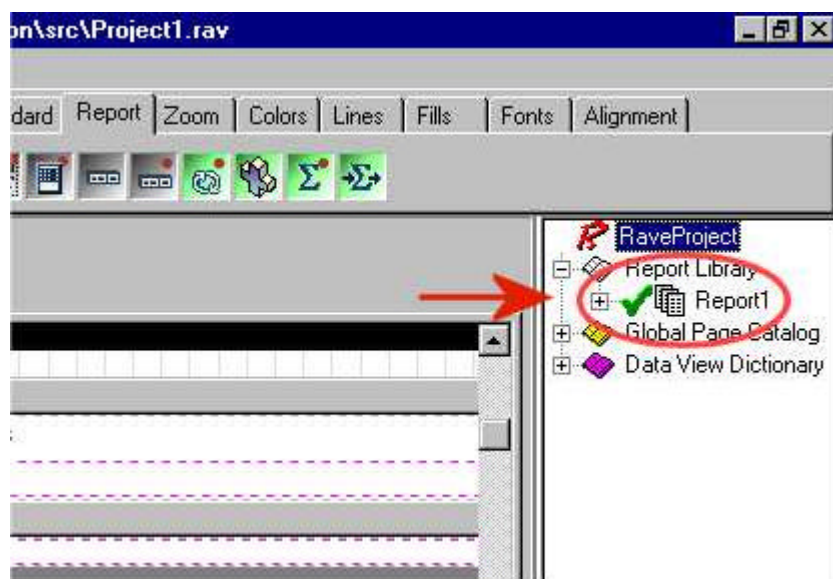
P – (no caso de Produtos)

S – (no caso de Serviços)

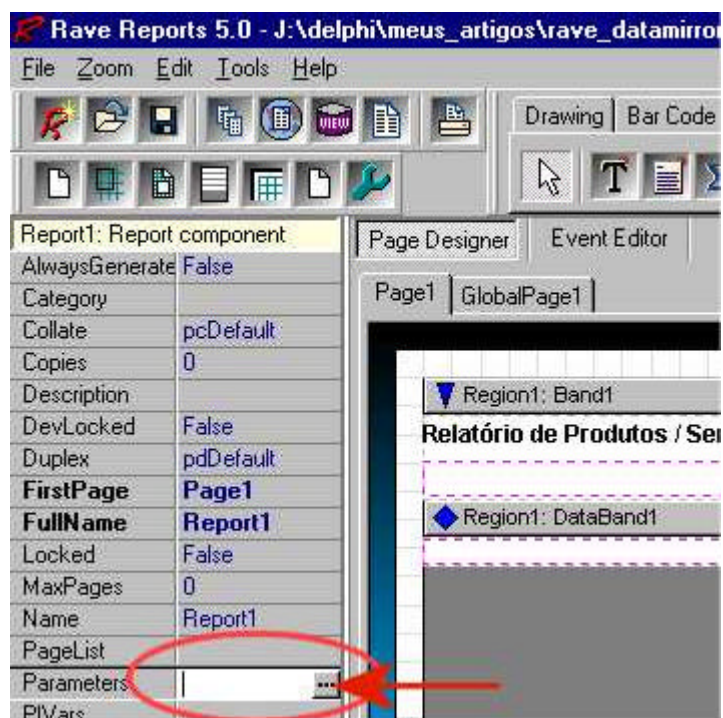
Ou seja, caso passe o valor do parâmetro como **"P"**, exibiremos o layout de **Produtos** e caso passe **"S"**, exibiremos o layout de **Serviços**.

Vamos então criar o parâmetro:

Na lista de objetos (tree view), que fica do lado direito, selecione o **Report1**:



Agora, no **Object Inspector**, clique na propriedade **Parameters** (nos três pontinhos) para exibir a lista de parâmetros do **Report1**.



Inclua um item chamado **Genero**:



Com isso, já temos nosso parâmetro criado, onde ele será alimentando em **tempo de execução** através da aplicação.

Agora vamos para parte legal da história, fazer exibir determinado layout de acordo com este parâmetro:

Selecione o **primeiro DataMirrorSection**, que está em **Band1**, ele que é responsável em exibir os **títulos das colunas**.

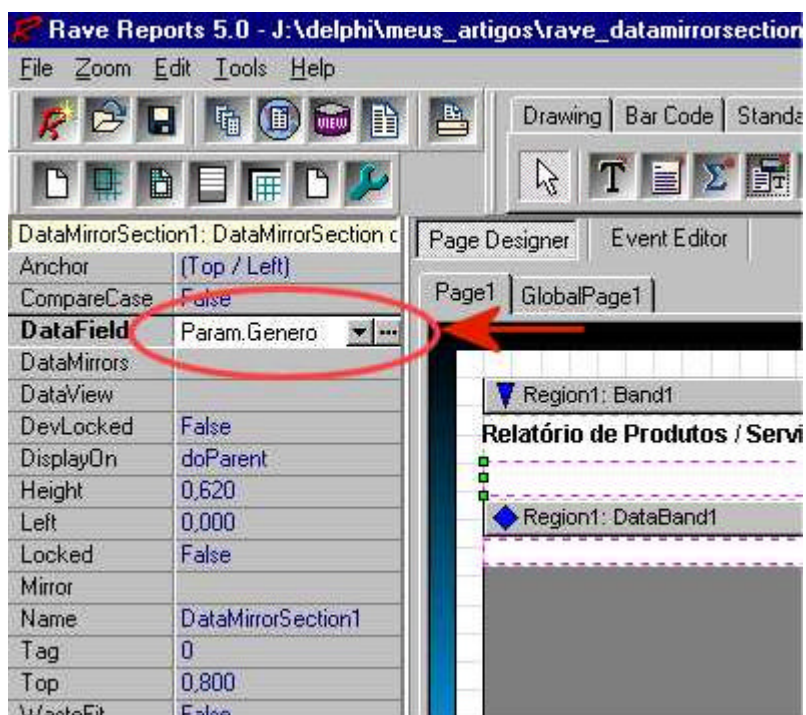
Vamos modificar apenas 2 propriedades dele:

DataField

É nesta propriedade que determinamos **com que campo** o **DataMirrorSection** irá comparar os valores para exibir determinado layout.

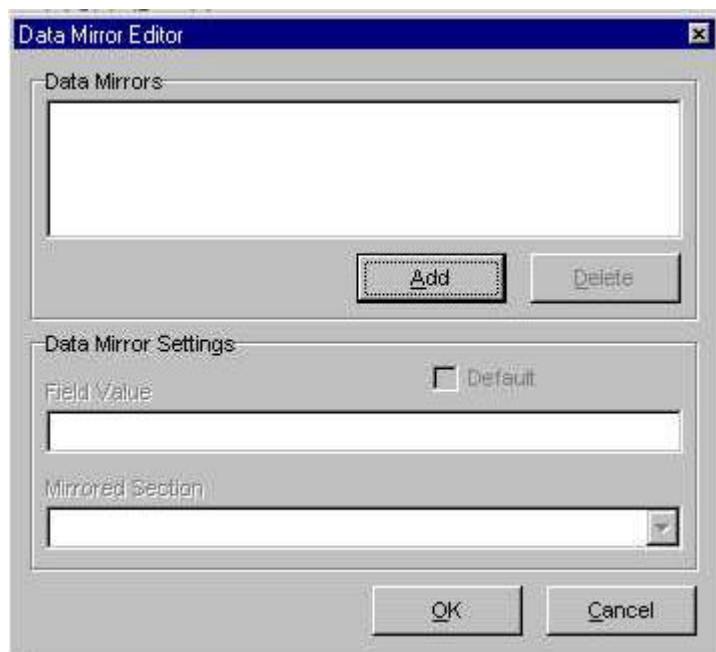
Aqui poderíamos usar parâmetros, campos de tabelas, etc. Basta clicar nos três pontinhos que será exibido a tela para seleção do campo...No nosso caso, é simples, queremos apenas comparar com o parâmetro **Genero** que acabamos de criar.

Para isso, digite o seguinte valor nesta propriedade: **Param.Genero**



Agora iremos para a propriedade **DataMirrors**, é nela que indicamos **o que exibir** de acordo com o valor do **parâmetro Genero**.

Clicando nesta propriedade(nos três pontinhos), aparecerá a seguinte tela:



Calma...não se assuste, parece ser complexa, mas não é... :-)

Aqui, basicamente temos que nos preocupar apenas com duas propriedades: **Field Value** e **Mirrored Section**.

O **Field Value** determina o **valor** do campo a ser comparado para exibir uma determinada **Section**, onde está **Section** será informada na propriedade **Mirrored Section**.

Já definimos "**com o que**" o **DataMirrorSection** será comparado, definimos que será comparado com o **parâmetro gênero** (lembre-se, definimos isso na propriedade DataField).

Agora precisamos determinar "**o que**" exibir se o **parâmetro gênero** for "**P**" e "**o que**" exibir se for "**S**".

Então temos que adicionar 2 itens nesta tela, pois temos duas situações.

Vamos lá...

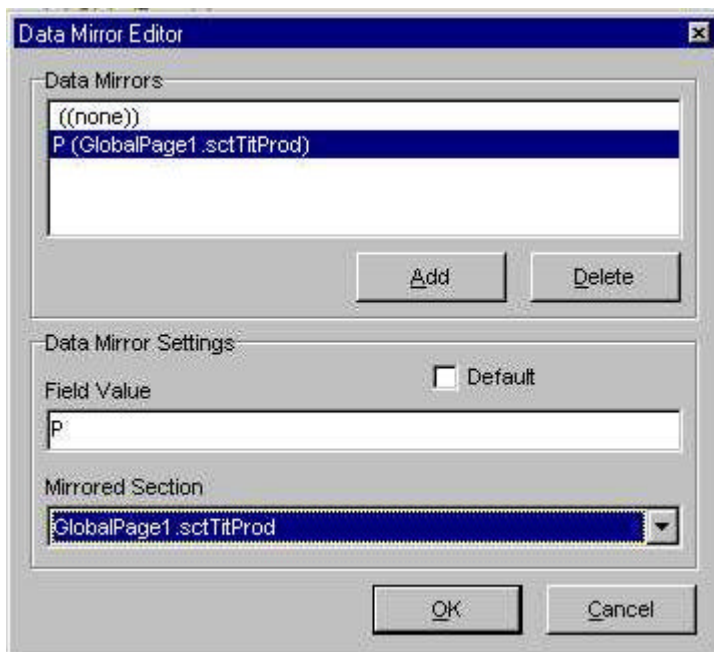
Clique 2 vezes no botão "Add".

Aparecerá 2 itens na tela.

Selecione **um item** e ajuste as propriedades:

FieldValue: **P**

Mirrored Section: **GlobalPage1.sctTitProd**

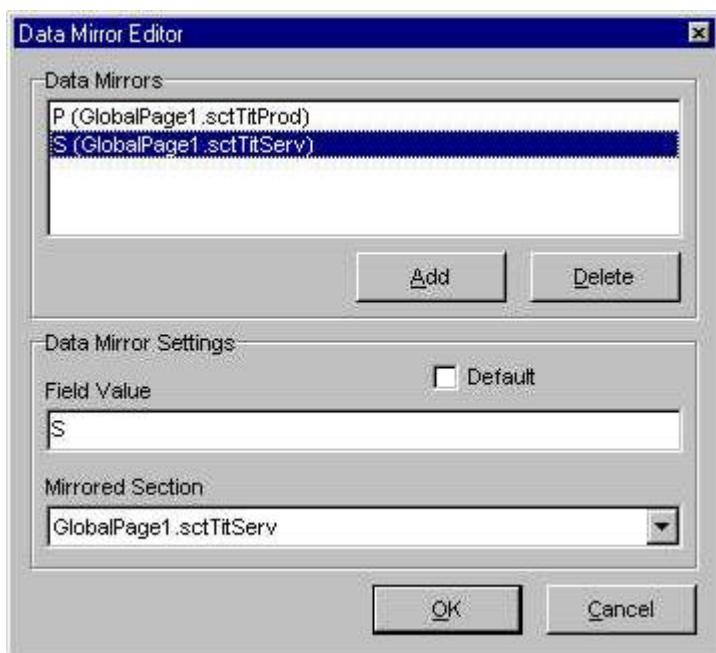


Isto determina que, se o **parâmetro gênero** for "**P**", então deverá exibir a **Section sctTitProd** que está em **GlobalPage1** (lembre-se, esta é a section que montamos para o layout dos **títulos** das colunas para **produtos**).

Agora selecione o **outro item** e ajuste as propriedades :

FieldValue : **S**

Mirrored Section: **GlobalPage1.sctTitServ**



Isto determina que se o **parâmetro gênero** for "**S**", então deverá exibir a **Section sctTitServ** que está na **GlobalPage1** (lembre-se, esta é a section que montamos para o layout dos **títulos** das colunas para **serviços**).

O campo **default**, determina que o **item** será exibido caso o valor do campo **não** seja nenhum dos valores comparados. Imagine que o valor do **parâmetro gênero** **não** seja informado ou seja **diferente** de "**P**" e "**S**", então neste caso, **nada** seria exibido no **DataMirrorSection**. Nesta situação, você poderá determinar qual é a **Section**, qual item padrão a ser exibido, para isso basta **ligar** o checkbox de **default**.

Feito isso, pronto, basta clicar no botão ok para confirmar.

O que fizemos neste momento, foi em relação ao **layout dos títulos das colunas**. Agora o mesmo terá que ser feito para o **layout dos registros**, ajustando o **outro DataMirrorSection** que está em **DataBand1**.

Os procedimentos serão os mesmos, basta ficar atento que, o que será exibido agora é a **Section sctRegProd** ou **sctRegServ** (section's relativas ao layout de **registros**).

Pronto, depois de ter ajustado o **DataMirrorSection dos registros**, execute o relatório passando o **parâmetro gênero** como "**P**" e depois como "**S**" para ver o resultado final.

Acho que de agora em diante, você pensará duas vezes antes de fazer 2 relatórios para este caso simples. O **DataMirrorSection** não é usado somente para este caso, tudo dependerá da sua criatividade. Imagine você querendo exibir imagens diferentes de acordo com um valor do campo da tabela...Em fim, o importante é saber como o componente funciona e para que serve, agora o que você irá fazer com ele, só depende de você....:-)

Eduardo Rocha
eduardo@edudelphipage.com.br
<http://www.edudelphipage.com.br>