

Processador XSL-FO – Manual do Utilizador

Este documento serve para exemplificar a utilização do processador XSL-FO usado na geração de PDFs.

Expressões de substituição

As expressões permitem efectuar desde simples substituições de texto a elaborados cálculos matemáticos. As expressões são avaliadas recorrendo a uma linguagem de script baseada na linguagem Java, designada BeanShell. As expressões são construídas da seguinte forma:

\$expressão : tipo de dados \$

Os tipos de dados disponíveis são:

- # c – código: Avalia a expressão mas não apresenta o resultado.
- # t – texto: Avalia a expressão e substitui pelo valor (sem formatação)
- # e – euro: Avalia a expressão e devolve o valor formatado como valor monetário em euros.
- # p – percentagem: O valor numérico resultante é formatado como percentagem.
- # r – real: O valor numérico como número real.
- # d – data: A data resultante é apresentada no formato dd/mm/aaaa.
- # i – inteiro: O valor numérico é formatado como número inteiro, sem casas decimais.
- # s – substituição: Funcionalidade avançada. (*TODO documentar*)

Exemplo:

\$ carro:t \$ = volvo

\$ preco:e \$ = 35.000,00 €

IVA: \$ preco*0.21:e \$ = IVA: 7.350,00 €

Controlo do Fluxo de Execução

Cada controlo deverá ficar num único parágrafo, consistindo apenas no texto de controlo. Isto porque, a partir do momento em que é identificado um parágrafo de controlo, ele será removido. Isto poderá trazer algum transtorno, uma vez que o texto não vai ficar conforme desenhado no processador de texto.

Controlo IF

O bloco de controlo IF serve para incluir texto de acordo com uma ou mais condições. As condições serão avaliadas usando BeanShell e deverão resultar num valor booleano.

A sintaxe do controlo IF é a seguinte:

#if : condição 1#

Texto a incluir se a condição 1 for verdadeira

#elseif : condição 2#

Texto a incluir se a condição 2 for verdadeira

.....

#elseif : condição n#

Texto a incluir se a condição n for verdadeira

#else#

Texto a incluir se nenhuma das condições anteriores for verdadeira.

#end#

O #endif# ainda é admitido para manter a compatibilidade com versões anteriores da biblioteca.

Exemplo	Resultado
<pre>#if:(carro.equals("volvo"))# O Carro é um Volvo. #elseif:(carro.equals("mercedes") carro.equals("bmw"))# O Carro é uma bomba!! #else# O Carro é um chasso velho. #end#</pre>	O Carro é um Volvo.

Controlo ATABLE

O controlo ATABLE permite representar listas de forma tabular, reconstruindo uma tabela com repetição de linhas e/ou remoção de colunas de uma tabela template.

A sintaxe é a seguinte:

`#atable:variável : valor inicial : valor final : incremento#`

seguida de uma tabela template.

Não existe tag para marcar o fim do processamento de tabela.

A tabela de template é definida da seguinte forma:

1. Linha de controlo de colunas, em que cada célula da tabela contém a tag `#if:condição #`. A coluna correspondente a cada célula só é apresentada se a condição for verdadeira. Esta não será apresentada no resultado final.
2. Cabeçalho da tabela que corresponde à primeira linha sem informação de controlo.
3. Linhas de controlo de linha (opcional). Na primeira célula de cada linha da tabela poderá ser introduzida uma tag que indica processamento especial para a linha em que se encontra. Isto permite substituir ou remover linhas da tabela template. As tags disponíveis são as seguintes:
 1. `#showif: condição #` - Caso a condição seja verdadeira, apresenta esta linha e termina o ciclo de replicação de linhas.
 2. `#removeif: condição #` - Não apresenta esta linha se a condição for verdadeira.
 3. `#header#` - Indica que esta linha deverá ser acrescentada ao cabeçalho da tabela.
 4. `#footer#` - Indica que esta linha deverá ser acrescentada ao rodapé da tabela.
4. Linhas “normais” que serão repetidas sem restrições.

Usando a informação na tag “atable”, é gerado um ciclo, criando a variável *variavel* com o valor *valor inicial* no interpretador beanshell que será incrementada em valores de *incremento* até alcançar o *valor valor final*. Em cada iteração do ciclo as linhas da tabela template serão repetidas pela ordem em que são encontradas, interrompendo a repetição de linhas caso a linha de controlo assim o obrigue (por exemplo, `#showif: condição #`).

Exemplo, o controlo ATABLE seguinte:

#atable:ii:0:10:1#

#if:true#	#if:(carro.equals("volvo"))#	#if:false#
Coluna 1	Coluna 2	Coluna 3
#showif:ii == 4# ii é 4	Aparece se ii é 4	Também aparece se ii é 4
#removeif:ii==7# ii não é 7	Removido se ii é 7	Também é removido se ii é 7
Linha normal: ii=\$ii:i\$	A mesma linha normal	Ainda a linha normal

irá gerar a tabela seguinte:

Coluna 1	Coluna 2
ii não é 7	Removido se ii é 7
Linha normal: ii=0	Coluna do Volvo
ii não é 7	Removido se ii é 7
Linha normal: ii=1	Coluna do Volvo
ii não é 7	Removido se ii é 7
Linha normal: ii=2	Coluna do Volvo
ii não é 7	Removido se ii é 7
Linha normal: ii=3	Coluna do Volvo
ii é 4	Aparece se ii é 4
ii não é 7	Removido se ii é 7
Linha normal: ii=5	Coluna do Volvo
ii não é 7	Removido se ii é 7
Linha normal: ii=6	Coluna do Volvo
Linha normal: ii=7	Coluna do Volvo
ii não é 7	Removido se ii é 7
Linha normal: ii=8	Coluna do Volvo
ii não é 7	Removido se ii é 7
Linha normal: ii=9	Coluna do Volvo

NOTA: A tabela gerada contém informação para manter as linhas repetidas juntas na mesma página. O resultado final está dependente das heurísticas de *typesetting* do FOP. Outro efeito indesejado é a altura das linhas da tabela quando são introduzidas linhas de controlo. Se o gerador/conversor de FO incluir informação com a altura das linhas, esta será preservada, o que pode levar à geração de uma tabela com linhas de tamanhos diferentes. A inclusão outros controlos ATABLE dentro de um controlo ATABLE não é suportada.

Controlo FOR

O controlo FOR segue a lógica do ATABLE, mas para um pedaço de texto em vez de uma linha da tabela template.

A sintaxe é a seguinte:

#for: variável : valor inicial : valor final : incremento #

Para ilustrar o funcionamento do controlo FOR, temos o seguinte exemplo:

Exemplo	Resultado
<i>#for:i:0:10:1#</i> Linha \$i:i\$ <i>#end#</i>	Linha 0 Linha 1 Linha 2 Linha 3 Linha 4 Linha 5 Linha 6 Linha 7 Linha 8 Linha 9

Controlo FOREACH

O controlo FOREACH é uma alternativa controlo FOR. Em vez de usar um ciclo controlado por incrementos numéricos, itera-se nos elementos de uma lista ou array.

A sintaxe é a seguinte:

#foreach: variável : lista ou array #

Exemplo	Resultado
<i>#foreach:item:stock#</i> Item: \$item:t\$ <i>#end#</i>	Item: parafusos Item: porcas Item: anilhas

Controlo WHILE

O controlo WHILE efectua um ciclo enquanto uma expressão for verdadeira.

A sintaxe é a seguinte:

#while: expressão #

Exemplo	Resultado
<i>\$k=0:c\$</i> <i>#while:k<3#</i> Item: \$++k:i\$ <i>#end#</i>	Item: 1 Item: 2 Item: 3

Outros processamentos

Para além de controlos específicos para o fluxo de execução do processador de XSL-FO, foram definidos alguns controlos adicionais de apresentação.

Imagens

É possível gerar PDFs com imagens incluídas no documento original. No entanto, por vezes as imagens não são incluídas correctamente durante a fase de conversão para XSL-FO. Para facilitar a inclusão de imagens, foi criado o controlo IMAGE.

A sintaxe é a seguinte:

#image: largura : altura : URI #.

O bloco de texto onde o controlo IMAGE foi encontrado será substituído por uma inclusão de imagem. A imagem encontra-se no URI indicado (pode ser relativo ou pode ser um URL absoluto). Opcionalmente, pode-se definir uma largura e altura para a imagem. Os caracteres separadores “:” deverão prevalecer caso a altura e/ou largura não sejam definidos.

A imagem será apresentada centrada no local onde se encontra: página, célula de tabela, etc.

Exemplo	Resultado
# image:200:50:"infosistema.png" #	

Fontes e tipos de letra

O controlo FONT foi definido para contornar o problema de quando uma determinada fonte não está disponível no computador onde foi escrito o documento usado para gerar o ficheiro XSL-FO. Para tal, deverá ser indicado um descriptor da fonte a usar.

A sintaxe é a seguinte:

#font:descriptor da fonte#

O descriptor é uma string, avaliada pelo BeanShell, que corresponde ao valor da propriedade “font” definida na norma CSS2 e resume informação sobre família (tipo de letra), tamanho, estilo, etc. Infelizmente, o FOP apenas consegue reconhecer informação sobre família e tamanho. Por esse motivo o descriptor deverá tomar o seguinte valor:

<tamanho> <família>

Para mais informações, consultar:

<http://www.renderx.com/tutorial.html> - Font and Text Attributes (link)

XSL-FO - Formatting Properties (link)

FOP Compliance Page - Common Font Attributes (link)

Na tabela seguinte são apresentados alguns exemplos de fontes:

Exemplo	Resultado
#font:"10pt ocrb10"# 1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier; #end#	1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier;
#font:"14pt Times"# 1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier; #end#	1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier;
#font:"10pt Helvetica"# 1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier; #end#	1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier;
#font:"8pt Arial Black"# 1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier; #end#	1234567890 abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Aqui é times; Aqui é helvetica; Aqui é courier;

NOTA: É necessário possuir uma licença para embeber fontes num ficheiro PDF, inclusivamente as fontes padrão do PDF.

Observações

Existem algumas observações e/ou limitações relacionadas com as bibliotecas e ferramentas utilizadas.

Processador de Texto

Por vezes o processador de texto utilizado gera dois blocos de texto consecutivos, partindo uma frase em duas partes. Do ponto de vista do utilizador não existe diferença, uma vez que o texto aparece seguido, mas do ponto de vista de implementação poderá impedir a correcta detecção de expressões e/ou controlos. Uma forma de contornar este problema é seleccionar toda a expressão/texto que deveria ser detectado e aplicar o mesmo estilo ou limpar a formatação.

No entanto, esta limitação poderá ser explorada para apresentar texto que de outra forma seria detectado como expressão ou controlo, como é o caso deste documento.

Conversor RTF para XSL-FO

Por vezes os processadores de texto possibilitam a introdução de objectos externos como caixas de texto, botões, formatações complicadas e/ou funcionalidades específicas da aplicação. Como o conversor não consegue lidar com estes objectos externos, ignora-os, não sendo reflectidos no resultado final.

Esta limitação não só se aplica a este conversor, mas sim a todos os conversores de qualquer ficheiro para XSL-FO.

FOP 0.95 beta

O FOP (versão 0.95 beta) é a biblioteca que faz a transformação do documento XSL-FO para PDF, contém algumas limitações. Apesar de ser uma versão beta, encontra-se estável e corrige alguns bugs relacionados com o carregamento de fontes. Para mais informação, consultar:

Apache FOP ([link](#))

Apache FOP Compliance ([link](#))

Inclusão de imagens e outros recursos externos

Por vezes as imagens não são detectadas e/ou encontradas. Para facilitar a detecção de imagens externas, foi alterado o código para pesquisar de acordo com a seguinte ordem:

1. Classpath – Tenta abrir a imagem como *java resource*
2. Classpath – Igual ao anterior, mas acrescentar o carácter “/” ao inicio da localização do recurso
3. Sistema de ficheiros – Procura o recurso no directório actual
4. Sistema de ficheiros – Procura o recurso no directório “fonts”
5. Sistema de ficheiros – Procura o recurso num directório indicado programaticamente
6. Localizações por omissão definidas na ferramenta FOP

NOTA: Esta funcionalidade também é utilizada para localizar ficheiros de fontes para incluir no PDF.

BeanShell

O BeanShell é um motor de scripting com sintaxe semelhante ao Java. Este é usado para avaliar todas as expressões utilizadas no documento, assim como alguns dos parametros passados ao vários controlos de fluxo de execução. Para mais informação, consultar:

BeanShell - Lightweight scripting for Java ([link](#))

Tabelas e controlo ATABLE

Por vezes a tabela resultante do controlo ATABLE não tem os contornos de linhas, células ou colunas da tabela. Isto está relacionado com a forma com que a tabela foi formatada no processador de texto. É frequente ocorrer com o cabeçalho da tabela ou quando se remove a última coluna, uma vez que se encontram no interior da tabela template. Caso esta situação ocorra, deverá ser seleccionada a célula/coluna/linha em questão e forçar a criação dos contornos para cada célula ou linha.