



**PrefSuite**

*the window maker's preference*

# Listados y Etiquetas personalizados de PrefGest

## WhitePaper

Julio 2008



## Contenidos

1. Propósito	3
2. Introducción	4
3. Listados personalizados	5
3.1. Ejemplos prácticos	7
3.2. Consideraciones adicionales	9
4. Etiquetas personalizadas	11

## 1. Propósito

Además de las plantillas List & Label que PrefSuite incluye para la presentación de documentos, el usuario puede tener toda la cantidad de informes que necesite, obteniendo la información de la base de datos y teniéndola disponible para la confección de la plantilla utilizando dicha información.

Para ello, se utiliza código de Visual Basic en el proyecto VB asociado a la base de datos en cuestión, mediante el cual se construye una cadena XML para pasar los datos desde VB a List & Label. Más adelante se explicará el formato necesario de dicha cadena XML.

El presente documento va dirigido a las personas encargadas de la configuración y confección de informes.

Es necesario contar con conocimientos previos de PrefGest, List & Label, Visual Basic y XML.

## 2. Introducción

En primer lugar se programa el código Visual Basic necesario, utilizando para el XML el formato que se explicará a continuación y, posteriormente, se diseña la plantilla en List & Label haciendo uso de los campos y variables obtenidos a partir del código anterior.

Estos informes y etiquetas personalizados, se programan en los eventos correspondientes según la aplicación desde la que se desea presentar dicho informe o etiqueta. En el proyecto VB, estos eventos están identificados por la cadena 'PrintCustomizedReport', precedida del nombre correspondiente según el evento.

Al 'Preparar documento' en PrefGest, el apartado en el que se asociarían estos informes o etiquetas, es el que aparece bajo la carpeta de nombre 'Personalizado'.

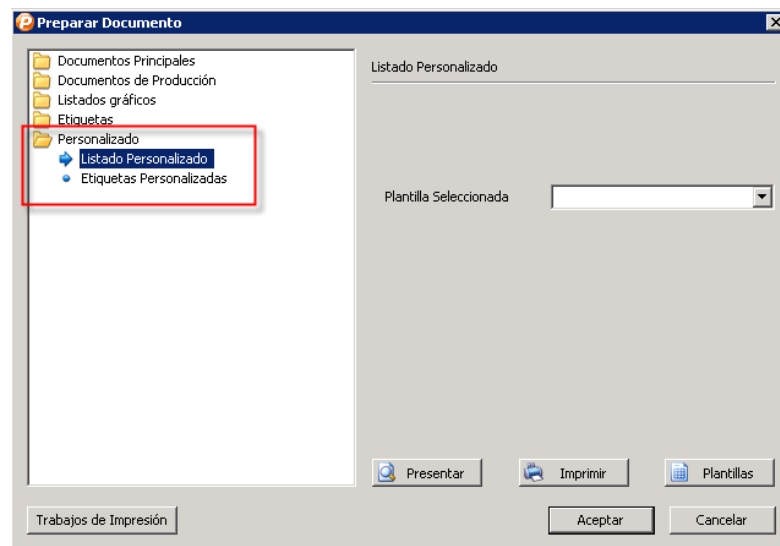


Figura 1. Preparar Documento > Personalizado.

### 3. Listados personalizados

Los listados personalizados en PrefGest utilizan una cadena XML para pasar los datos desde VBA a List & Label. El formato de dicha cadena XML es el siguiente:

- Elemento Raíz: debe llamarse **'CustomizedReport'** y además debe contener un atributo llamado **'Version'**, el cual especificará la versión que se ha utilizado para crear la estructura del documento XML (por el momento la única posible es la versión 1.0 que es la que se explica aquí).
- Elementos para definir variables de List & Label: deben llamarse **'Variable'** y pueden constar de hasta 18 atributos:
  - **Atributo 'Name'**: Es obligatorio que en cada elemento **'Variable'** haya un atributo **'Name'**, que será el que otorgará el nombre que simbolizará la variable a la hora de editar la plantilla de List & Label.
  - **Atributo 'Type'**: Simboliza el tipo de datos de la variable, habiendo 7 posibles valores (en caso de no especificarse se toma por defecto el tipo **'string'**):
    - **'bool'**: tipo de datos booleano. Toma como valores **'0'** o **'1'**.
    - **'long'**: tipo de datos entero.
    - **'double'**: tipo de datos con coma flotante.
    - **'string'**: tipo de datos cadena de caracteres.
    - **'datetime'**: tipo de datos fecha, admitiendo dos posibles modos de especificarla (según el estándar ISO8601): solamente año-mes-día (yyyy-mm-dd) o bien añadiendo además horas, minutos y segundos (yyyy-mm-ddThh:mm:ss). En este último caso la **'T'** se utiliza para separar la fecha de la hora.
    - **'sqlgraphic'**: tipo de datos metafile. Se trata de una cadena de texto que contiene una consulta SQL. Dicha consulta debe devolver una columna de una tabla de la base de datos donde la columna debe contener el metafile.
    - **'filegraphic'**: tipo de datos metafile. Se trata de una cadena de texto que contiene el nombre y la ruta completa de un archivo de tipo metafile.
    - **'rendergraphic'**: dibujo personalizado de una ventana. El valor del atributo **'Value'** de esta variable no importa. Si la variable es de este tipo, necesitará otros 3 atributos al menos (**'ColumnForNumber'** o **'VariableForNumber'**, **'ColumnForVersion'** o **'VariableForVersion'**, **'ColumnForPosition'** o **'VariableForPosition'**). Opcionalmente puede tener otros 4 atributos: **'ColumnForXMLVisualProperties'** o **'VariableForXMLVisualProperties'**, **'ColumnForPositionType'** o **'VariableForPositionType'**, y **'ColumnForPositionCode'** o **'VariableForPositionCode'**.
  - **Atributo 'Value'**: Debe contener el valor de la variable. En caso de no existir este atributo, se asigna a la variable un valor **'vacío'** por defecto (una cadena vacía para los tipos **'string'**, **'sqlgraphic'** y **'filegraphic'**, un **'0'** para **'bool'** y **'long'**, un **'0.0'** para **'double'** y una fecha nula, que será interpretada como una cadena vacía a la hora de presentarse o imprimirse por List & Label, para el tipo **'datetime'**).

- **Atributo 'Size'**: En caso de tratarse de una variable de tipo 'string', 'sqlgraphic' o 'filegraphic', mediante este atributo se indica el tamaño máximo que puede tener la cadena que se va a guardar en esa variable. Si no se especifica se asigna un valor por defecto de 100 para 'string' y 'filegraphic' y de 500 para 'sqlgraphic'.
- **Atributo 'ColumnForNumber'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Column' que especificará el número de PAF. En una variable 'renderergraphic' es necesario que se especifique un atributo de tipo 'ColumnForNumber' o de tipo 'VariableForNumber'.
- **Atributo 'VariableForNumber'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Variable' que especificará el número de PAF.
- **Atributo 'ColumnForVersion'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Column' que especificará la versión del PAF. En una variable 'renderergraphic' es necesario que se especifique un atributo de tipo 'ColumnForVersion' o de tipo 'VariableForVersion'.
- **Atributo 'VariableForVersion'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Variable' que especificará la versión del PAF.
- **Atributo 'ColumnForPosition'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Column' que hará referencia a la posición (en base 0) de la ventana en el PAF. En una variable 'renderergraphic' es necesario que se especifique un atributo de tipo 'ColumnForPosition' o de tipo 'VariableForPosition'.
- **Atributo 'VariableForPosition'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Variable' que hará referencia a la posición (en base 0) de la ventana en el PAF.
- **Atributo 'ColumnForXMLVisualProperties'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Column' que contendrá el XML de propiedades visuales del dibujo que se desea mostrar.
- **Atributo 'VariableForXMLVisualProperties'**: Equivalente al anterior pero en lugar de para un nodo 'Column', para un nodo 'Variable'.
- **Atributo 'ColumnForPositionType'**: En una variable de tipo 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Column' que contendrá información acerca de si la ventana es un submodelo ('Value'=1) o si es un escandallo ('Value'=2).
- **Atributo 'VariableForPositionType'**: Equivalente al anterior pero en lugar de para un nodo 'Column', para un nodo 'Variable'.
- **Atributo 'ColumnForPositionCode'**: En una variable 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Column' que hará referencia al código del escandallo en el caso de que el nodo 'Row' contenga un escandallo.
- **Atributo 'VariableForPositionCode'**: Equivalente al anterior pero en lugar de para un nodo 'Column', para un nodo 'Variable'.

- **Atributo 'ColumnForProductionLot'**: En una variable 'renderergraphic' especifica el valor del atributo 'Name' en el nodo 'Column' que hará referencia al lote al que pertenece el submodelo si no pertenece a ningún lote (por ejemplo porque el listado personalizado se presenta desde ventas).
- **Atributo 'VariableForProductionLot'**: Equivalente al anterior pero en lugar de para un nodo 'Column', para un nodo 'Variable'.
- Elementos para definir los campos de List & Label: se deben llamar '**Column**' y pueden constar de hasta 4 atributos:
  - **Atributo 'Name'**: Es obligatorio que en cada elemento 'Column' haya un atributo 'Name', que será el que otorgará el nombre que simbolizará el campo a la hora de editar la plantilla de List & Label.
  - **Atributo 'Type'**: Simboliza el tipo de datos del campo, habiendo 7 posibles valores (los mismos que para las variables).
  - **Atributo 'DefaultValue'**: Mediante este atributo se puede asignar un valor por defecto al campo por si en alguna fila de la tabla no se desea poner un valor para ese campo. En caso de no existir este atributo, se asigna como valor por defecto para el campo un valor 'vacío' por defecto (una cadena vacía para los tipos 'string', 'sqlgraphic' y 'filegraphic' un '0' para 'bool' y 'long', un '0.0' para 'double' y una fecha nula, que será interpretada como cadena vacía a la hora de mostrarse o imprimirse, para el tipo 'datetime').
  - **Atributo 'Size'**: En caso de tratarse de un campo de tipo 'string', 'sqlgraphic' o 'filegraphic', mediante este atributo se indica el tamaño máximo de la cadena que se va a guardar en ese campo. Si no se especifica se asigna un valor por defecto de 100 para 'string' y 'filegraphic' y de 500 para 'sqlgraphic'.
- Elementos para los datos a imprimir en la tabla: cada registro que se le pase a la tabla contenida en el listado estará asociado a un elemento '**Row**' dentro de la cadena XML. El número de atributos que puede tener un elemento '**Row**' puede variar entre 0 y el número máximo de elementos '**Column**' que se hayan incluido en la cadena XML. El nombre de esos atributos debe coincidir con alguno de los valores del atributo 'Name' de los distintos elementos '**Column**' que se hayan definido. Asociados a esos atributos estará el valor que contendrá ese campo para ese registro del listado. Si determinado campo correspondiente a un elemento '**Column**' no se especifica como atributo de un elemento '**Row**' entonces se tomará como valor para ese campo el valor por defecto asignado al procesar el anteriormente citado elemento '**Column**'.

### 3.1. Ejemplos prácticos

#### Definición de variables:

```
<Variable Name="Nombre"/>
```

No se define ni el valor ni el tipo ni la longitud máxima: se tomará por defecto el tipo 'string', el valor "" y la longitud de cadena 100.

```
<Variable Name="Apellido" Value="Pérez" Size="20"/>
```

Al no definirse un tipo de datos, la variable se tomará como de tipo 'string'. Contendrá el valor "Pérez" y el valor máximo de la cadena podría ser de 20 caracteres.

```
<Variable Name="WorkerCode" Value="28" Size="10" Type="long"/>
```

La variable será de tipo 'long' y el valor 28. Al no tratarse de una cadena el valor 'Size' será ignorado.

```
<Variable Name="Fecha" Value="2002-10-24" Type="datetime"/>
```

Se define una variable que contiene una fecha (24 de Octubre de 2002).

```
<Variable Name="Altura" Value="2.7" Type="double"/>
```

Se define una variable de tipo double, con el valor 2.7.

```
<Variable Name="EsCierto" Value="1" Type="bool"/>
```

Se define una variable de tipo booleano, de valor '1' (Verdadero).

```
<Variable Name="Dibujo" Type="sqlgraphic" Value="SELECT Metafile FROM
ContenidoPAFBlob WHERE numero=53 and version=1 and orden=0"/>
```

Se define una variable de tipo sqlgraphic, cuyo valor es una cadena que contiene una consulta SQL que a su vez devolverá un metafile para ser mostrado en la plantilla.

### Definición de campos (columnas de la tabla):

```
<Column Name="Trabajador" DefaultValue="Name" Size="30"/>
```

Campo donde no se especifica el tipo de datos, luego se toma como 'string'. La longitud de dicho string será como mucho de 30 caracteres. Si en algún elemento '**Row**' no se especifica el valor para 'Trabajador' entonces se asignará automáticamente el valor 'Name'.

```
<Column Name="Edad" DefaultValue="0" Size="4"/>
```

A pesar de ser un valor numérico (o al menos eso se desprende del nombre del atributo), el campo será pasado a List & Label como 'string', pues no se ha especificado ningún tipo de datos. El valor por defecto será '0'.

```
<Column Name="Date" Type="datetime"/>
```

Campo de datos que representará una fecha. Al no indicarse un valor por defecto, cuando en un elemento de tipo '**Row**' no se especifique el atributo 'Date' lo que se asignará por defecto será la fecha del momento de la presentación o impresión del documento List & Label.

```
<Column Name="Contador" Type="long" DefaultValue="0"/>
```

Campo de datos entero. Si no se especifica se tomará 0 como valor por defecto.

```
<Column Name="Metafile" Type="filegraphic"/>
```

Campo con una cadena que indica el nombre de un fichero que contiene un metafile. Por defecto se asignará un valor vacío a la cadena.

### Definición de los datos:

```
<Row Trabajador="Jorge" Edad="23" Date="2002-07-04"/>
```

Registro de datos donde se ha omitido la columna "Contador", a la que se asignará por defecto el valor 0. También se ha omitido el nombre de fichero, por lo que no se imprimirá ningún metafile asociado a este registro

```
<Row Trabajador="Jorge Again" Date="2002-07-04T15:59:59"  
Metafile="C:\temp\dibujo.wmf"/>
```

En este registro se ha utilizado el formato extendido de 'datetime' poniendo fecha y hora. Asociado al registro se imprimirá el metafile guardado en 'C:\temp\dibujo.wmf'. El valor de la edad y el del contador se pondrán a 0 por defecto.

```
<Row/>
```

Aunque el registro está vacío, se utilizarán todos los valores por defecto para cada campo: 'Name' para 'Trabajador', '0' para 'Edad', '0' para 'Contador', una fecha nula para el campo 'Date' y un nombre de fichero vacío para 'Metafile'.

## 3.2. Consideraciones adicionales

### ¿Qué ocurre si...

**...en un elemento de tipo 'string' la cadena que se intenta meter sobrepasa el valor del atributo 'Size'?** Si se trata de una variable no ocurre nada, y se toma el nuevo tamaño. Sin embargo, si se trata de un campo se trunca la cadena para adaptarla al tamaño correcto.

**...se repiten dos atributos dentro de un mismo elemento?** Independientemente de que se trate de 'Row' 'Column' o 'Variable' el listado da un error y no se imprime.

**...se colocan dos elementos 'Column' con el mismo nombre en el documento XML, con lo que se estará intentando crear dos campos con el mismo nombre en la plantilla List&Label?** Se lanza un mensaje de error en el programa. El listado no se imprime.

**...se colocan dos elementos 'Variable' con el mismo nombre en el documento XML, como si se estuviera intentando crear dos variables con el mismo nombre en la plantilla List&Label?** El segundo nombre es el que finalmente se corresponderá con la variable utilizada en la plantilla.

**...la consulta sql de una variable o campo 'sqlgraphic' está mal escrita?** SQL Server producirá un mensaje de error indicando por qué la consulta no es correcta. El campo de tipo metafile quedará vacío. Además, en modo Debug se producen varios Assertion Failed.

...la consulta sql de una variable o campo está vacía? El metafile asociado al campo o variable quedará vacío.

...el nombre de fichero introducido como valor en una variable o campo de tipo 'filegraphic' no se corresponde con un fichero que exista realmente? El metafile asociado al campo o variable quedará vacío.

...el fichero con el que se corresponde el valor de una variable o campo de tipo 'filegraphic' no se corresponde con un fichero de formato metafile? El metafile asociado al campo quedará vacío.

## Distribución de los elementos en el documento XML

En principio, la distribución de los objetos dentro del documento es indiferente excepto para el orden de impresión de los registros que se explica en el apartado c). Así, el usuario puede alternar elementos 'Row', 'Column' y 'Variable' dentro de la cadena XML con la única restricción de que el elemento raíz debe llamarse 'CustomizedReport'. Incluso si se desea se puede subdividir la estructura interna del documento, por ejemplo metiendo todos los elementos 'Row' dentro de otro elemento 'RowDirectory', los 'Column' en 'ColumnDirectory', etc.

Por ejemplo, los siguientes documentos XML tendrían el mismo comportamiento:

```
<CustomizedReport Version="1.0">
  <Row Number="1"/>
  <Variable Client="Cliente" Type="long"/>
  <Column Name="Version" DefaultValue="1" Type="long"/>
  <Column Name="Number" Type="long"/>
  <Row Number="23" Version="3"/>
</CustomizedReport>
```

```
<CustomizedReport Version="1.0">
  <ColumnGroup>
    <Column Name="Version" DefaultValue="1" Type="long"/>
    <Column Name="Number" Type="long"/>
  </ColumnGroup>
  <VariableGroup>
    <Variable Client="Cliente" Type="long"/>
  </VariableGroup>
  <RowGroup>
    <Row Number="1"/>
    <Row Number="23" Version="3"/>
  </RowGroup>
</CustomizedReport>
```

## Orden de los datos

El orden en que se pasan los datos a la tabla coincide con el orden en que están colocados los elementos 'Row' en la cadena XML. Este orden puede ser alterado mediante el uso de la variable de usuario de List & Label @SORT\_ORDER.

## 4. Etiquetas personalizadas

El funcionamiento de las etiquetas es muy similar al de los listados. La estructura del documento XML es exactamente igual en ambos casos, por lo que la misma cadena XML se puede utilizar tanto para imprimir un determinado listado como para imprimir las etiquetas asociadas a dicho listado. Cada elemento 'Row' de la cadena XML se corresponderá con una etiqueta. Sin embargo, en las plantillas de etiquetas no existen campos y variables de forma diferenciada, por lo que todos los datos son introducidos mediante variables. De este modo, tanto las declaraciones de las columnas (elementos 'Column') como las de las variables (elementos 'Variable') se corresponderán con variables a la hora de editar, imprimir o presentar una plantilla. La única diferencia entre ambas será que las variables correspondientes a los elementos 'Variable' serán constantes para todas y cada una de las etiquetas, mientras que las variables correspondientes a los elementos 'Column' variarán para cada etiqueta en función del valor que se asigne al atributo identificado por ese elemento 'Column' dentro del elemento 'Row'.

Al convertirse en variables tanto los elementos 'Variable' como los elementos 'Column', el atributo 'Name' de un elemento 'Variable' no debe coincidir con el de un elemento 'Column'. Si así ocurriera el elemento 'Variable' sería ignorado y el elemento 'Column' sería finalmente el que se asociaría a dicho nombre de variable.