



PrefSuite

the window maker's preference

VB y List & Label

WhitePaper

Junio de 2007



Contenidos

1. Propósito	3
2. Introducción	4
3. Añadir variables mediante VB al List & Label	5
3.1. Declaración de variables	5
3.2. Asignación de valores a las variables	5
4. Añadir campos mediante VB al List & Label	6
4.1. Declaración de campos	6
4.2. Asignación de valores a los campos	6
5. Añadir campos mediante VB a la Hoja de Trabajo	8
5.1. Declaración de campos	8
5.2. Asignación de valores a los campos	8

1. Propósito

Debido a que los listados en PrefGest se presentan utilizando las variables y/o campos que el programa otorga para tal fin, es necesario a veces obtener más información de la base de datos para dichos listados.

Este documento contiene explicaciones acerca de cómo personalizar ciertos listados de PrefGest, obteniendo de la base de datos el valor que interese y dejándolo disponible para el listado que corresponda.

Va dirigido a los usuarios encargados de la configuración y personalización de informes, por lo que es necesario tener conocimientos de List & Label.

Debido a que esta información se obtiene mediante el proyecto VB, es necesario también tener conocimientos de Visual Basic.

2. Introducción

Para la impresión de documentos en PrefGest, el programa pasa a List & Label una serie de variables y campos acordes a la aplicación desde la que se presenta el informe, y a la plantilla que se está utilizando.

Pero es posible, para la impresión de Presupuestos, Albaranes y Facturas, Hojas de Trabajo y Pedidos a Proveedor, obtener más información de la base de datos y pasarla como variable o como campo a List & Label, para contar también con ellos en dichos listados.

Estas personalizaciones se programan en el proyecto VB, en los eventos y módulos que se detallan a continuación.

3. Añadir variables mediante VB al List & Label

3.1. Declaración de variables

En el siguiente ejemplo, mediante la instrucción **“DefineVariable”** se creará la variable “Nombre” dentro del nivel “VarPAF”.

El código suministrado a continuación ha de situarse en el evento Sales_DefineVariablesAndFields del módulo SalesModule del proyecto VB.

```
Public Sub Sales_DefineVariablesAndFields(ByVal Sales As PrefGest.ISales, ByVal hJob As Long, ByVal SalesDoc As PrefSales.ISalesDoc, ByVal DocType As SalesPrintDocumentType, ByVal Layout As String)
    Sales.DefineVariable hJob, "VarPAF.Nombre", "", PLL_TEXT
End Sub
```

3.2. Asignación de valores a las variables

Este evento se ejecutará una vez y permite asignar el valor a las variables.

```
Public Sub Sales_SetVariables(ByVal Sales As PrefGest.ISales, ByVal hJob As Long, ByVal SalesDoc As PrefSales.ISalesDoc, ByVal DocType As SalesPrintDocumentType, ByVal Layout As String)

    'Abrimos una conexión a la base de datos
    Dim cn As New ADODB.Connection
    cn.CursorLocation = adUseClient
    cn.Open Application.ConnectionString

    'Niveles del diseño
    dim strQuery as String
    strQuery = "select Nombre " & _
    "from PAF " & _
    "where Numero = " & CStr(SalesDoc.Fields("Number").Value) & " and version = "
    & CStr(SalesDoc.Fields("Version").Value)

    Dim rsPaf As ADODB.Recordset
    Set rsPaf = New ADODB.Recordset
    rsPaf.Open strQuery, cn, adOpenStatic, adLockReadOnly

    Sales.DefineVariable hJob, "VarPAF.Nombre", Trim(rsPaf.Fields("Nombre")),
    PLL_TEXT

    rsPaf.Close
    cn.Close

End Sub
```

4. Añadir campos mediante VB al List & Label

4.1. Declaración de campos

En el siguiente ejemplo, mediante la instrucción **DefineField** se añadirán los campos Nivel1, Nivel2, Nivel3, Nivel4 y Nivel5 dentro del nivel "Niveles".

El código suministrado a continuación ha de situarse en el evento Sales_DefineVariablesAndFields del módulo SalesModule del proyecto VB.

```
Public Sub Sales_DefineVariablesAndFields(ByVal Sales As PrefGest.ISales, ByVal
hJob As Long, ByVal SalesDoc As PrefSales.ISalesDoc, ByVal DocType As
SalesPrintDocumentType, ByVal Layout As String)

    Sales.DefineField hJob, "Niveles.Nivel1", "", PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel2", "", PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel3", "", PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel4", "", PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel5", "", PLL_TEXT

End Sub
```

4.2. Asignación de valores a los campos

Este evento se ejecutará tantas veces como líneas contenga el presupuesto y asignará los valores a los campos creados.

```
Public Sub Sales_SetFields(ByVal Sales As PrefGest.ISales, ByVal hJob As Long,
ByVal SalesDocItem As PrefSales.ISalesDocItem, ByVal DocType As
SalesPrintDocumentType, ByVal Layout As String)

    'Abrimos una conexión a la base de datos
    Dim cn As New ADODB.Connection
    cn.CursorLocation = adUseClient
    cn.Open Application.ConnectionString
    Dim strQuery as String

    'Niveles del diseño
    strQuery = "select Nivel1,Nivel2,Nivel3,Nivel4,Nivel5 " & _
"from ContenidoPaf Inner join Dibujos on Concepto=Codigo " & _
"where Numero = " & CStr(SalesDoc.Fields("Number").Value) & " and version = "
& CStr(SalesDoc.Fields("Version").Value) & _
" and orden = " & CStr(SalesDoc.Fields("SortOrder").Value)

    Dim rsDibujos As ADODB.Recordset
    Set rsDibujos = New ADODB.Recordset
    rsDibujos.Open strQuery, cn, adOpenStatic, adLockReadOnly

    Sales.DefineField hJob, "Niveles.Nivel1", Trim(rsDibujos.Fields("Nivel1")),
```

```
PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel2", Trim(rsDibujos.Fields("Nivel2")),
PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel3", Trim(rsDibujos.Fields("Nivel3")),
PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel4", Trim(rsDibujos.Fields("Nivel4")),
PLL_TEXT
    Sales.DefineField hJob, "Niveles.Nivel5", Trim(rsDibujos.Fields("Nivel5")),
PLL_TEXT

    rsDibujos.Close
    cn.Close
End Sub
```

5. Añadir campos mediante VB a la Hoja de Trabajo

5.1. Declaración de campos

Para pasar campos a la Hoja de Trabajo, se utiliza el evento `Worksheet_DefineVariablesAndFields` del módulo `WorksheetModule`, utilizando la siguiente instrucción:

```
WorkSheet.Application.DefineField hJob, "Carpeta.Campo", "", PLL_NUMERIC
```

(`PLL_NUMERIC` o el tipo que corresponda)

5.2. Asignación de valores a los campos

Para dar valor a los campos añadidos, se utiliza el evento `Worksheet_NewWindow` del módulo `WorksheetModule`, utilizando la siguiente instrucción:

```
WorkSheet.Application.DefineField hJob, "Carpeta.Campo", valor_campo, PLL_NUMERIC
```

(`PLL_NUMERIC` o el tipo que corresponda)

Los tipos de datos que se pueden utilizar para pasar variables/campos a los distintos informes posibles son:

`PLL_NUMERIC`

`PLL_TEXT`

`PLL_DATE_YYYYMMDD`