

# **EXPORTAR VB.NET A EXCEL XP**

Este artículo describe varios métodos para transferir datos a Excel XP desde Visual Basic .NET, Este artículo también presenta las ventajas y desventajas de cada método para que pueda seleccionar la solución que mejor se adapte a su situación.

# DESCRIPCIÓN

La técnica que se utiliza con más frecuencia para transferir datos a un libro de Excel es la Automatización. Con Automatización, puede llamar a métodos y propiedades que son específicas de las tareas de Excel, le ofrece la mayor flexibilidad para especificar la ubicación de sus datos en la hoja de cálculo, y la capacidad de formato de la hoja de cálculo como hacer diversos ajustes en tiempo de ejecución.

Con Automatización, puede utilizar varias técnicas para transferir sus datos:

- > Transferencia de datos celda por celda.
- > Transferencia de datos de una matriz a un rango de celdas.
- Transferencia de datos de un Recordset ADO a un rango de celdas utilizando el método CopyFromRecordset.
- Crear un objeto QueryTable en una hoja de cálculo de Excel que contiene el resultado de una consulta ODBC u OLEDB como fuente de datos.
- Transferencia de datos al portapapeles, y luego pegar el contenido del portapapeles en una hoja de cálculo de Excel.

También puede utilizar varios métodos que no requieren necesariamente de Automatización para transferir datos a Excel. Si está ejecutando un programa del lado del servidor, este puede ser un buen método para la adopción de la mayor parte del procesamiento de datos fuera de sus clientes.

Los siguientes criterios pueden ser usados para transferir sus datos sin Automatización:

- Transferir sus datos a un File o archivo de texto delimitado que Excel pueda analizar más adelante.
- Transferir sus datos a una hoja de trabajo usando ADO.NET.
- Transferencia de datos XML a Excel (versión 2002) para proporcionar ordenados en filas y columnas.

## **REQUERIMIENTOS:**

- Descargar Office XP Pias
- MS Office XP
- Visual .Net 2003 o 2005

#### **Técnicas**

#### UTILICE AUTOMATIZACIÓN PARA TRANSFERIR DATOS CELDA POR CELDA

Con Automatización, puede transferir datos a una hoja de trabajo una célula a la vez, de la siguiente manera.

No te olvides que tienes que agregar la referencia: *Microsoft Excel 10.0 Object Library* 

Crea una carpeta en C:\ExcelData\, En la parte general de la Clase declara lo siguiente: **Const sSampleFolder = "C:\ExcelData\"** 

Dim oExcel As Object

Dim oBook As Object

Dim oSheet As Object

'Start a new workbook in Excel.

oExcel = CreateObject("Excel.Application")

oBook = oExcel.Workbooks.Add

'Add data to cells of the first worksheet in the new workbook.

oSheet = oBook.Worksheets(1)

oSheet.Range("A1").Value = "Last Name"

oSheet.Range("B1").Value = "First Name"

oSheet.Range("A1:B1").Font.Bold = True

oSheet.Range("A2").Value = "Doe"
oSheet.Range("B2").Value = "John"

'Save the Workbook and quit Excel.
oBook.SaveAs(sSampleFolder & "Book1.xls")
oSheet = Nothing
oBook = Nothing
oExcel.Quit()
oExcel = Nothing
GC.Collect()

#### Observación

La transferencia de datos de celda por celda puede ser un criterio aceptable si no hay mucho a la transferencia de datos. También tienes la posibilidad de poner los datos en cualquier parte de la hoja de cálculo y aplicarle formato de las celdas en tiempo de ejecución condicional. Sin embargo, este método no se recomienda si usted tiene una gran cantidad de datos para transferir a una hoja de cálculo de Excel. Por lo que, la transferencia de datos ser lenta.

#### UTILICE AUTOMATIZACIÓN PARA TRANSFERIR MATRIZ EN UNA HOJA DE

#### CÁLCULO

Un conjunto de datos puede ser transferido a un conjunto de múltiples celdas, al mismo tiempo, de la siguiente manera.

Dim oExcel As Object
Dim oBook As Object
Dim oSheet As Object
'Start a new workbook in Excel.
oExcel = CreateObject("Excel.Application")
oBook = oExcel.Workbooks.Add

'Create an array with 3 columns and 100 rows.

Dim DataArray(99, 2) As Object

```
Dim r As Integer
For r = 0 To 99
DataArray(r, 0) = "ORD" & Format(r + 1, "0000")
DataArray(r, 1) = Rnd() * 1000
DataArray(r, 2) = DataArray(r, 1) * 0.07
Next
```

'Add headers to the worksheet on row 1.

oSheet = oBook.Worksheets(1)

oSheet.Range("A1").Value = "Order ID"

oSheet.Range("B1").Value = "Amount"

oSheet.Range("C1").Value = "Tax"

'Transfer the array to the worksheet starting at cell A2. oSheet.Range("A2").Resize(100, 3).Value = DataArray

```
'Save the Workbook and quit Excel.
oBook.SaveAs(sSampleFolder & "Book2.xls")
oSheet = Nothing
oBook = Nothing
oExcel.Quit()
oExcel = Nothing
```

GC.Collect()

Si la transferencia de datos mediante el uso de un Array en lugar de la celda por celda, se puede realizar una gran ganancia de rendimiento con un gran volumen de datos. Considere esta línea en el código anterior, que transfiere datos a 300 celdas de la hoja de trabajo.

OSheet.Range ( "A2"). Resize (100, 3). Value = DataArray

Esta línea representa dos solicitudes de interfaz: una para el área de distribución objeto que devuelve el método Range, y otro para el área de distribución que devuelve el método de cambio de tamaño. En cambio, la transferencia de datos celda por celda requiere solicitudes para 300 interfaces de los objetos Range. Siempre que sea posible, usted puede beneficiarse de la transferencia de sus datos y la reducción del número de solicitudes que realice la interfaz.

#### TRANSFERENCIA DE DATOS DE UN RECORDSET ADO A UN RANGO DE CELDAS

El objeto modelos de Excel 2000 y Excel 2002 proporcionan el método CopyFromRecordset para la transferencia de un recordset ADO a un rango en una hoja de trabajo. El siguiente código muestra cómo automatizar Excel para transferir el contenido de las órdenes en la base de datos de ejemplo Northwind utilizando el método CopyFromRecordset. Para esto nescitamos declarar una variable Const cn la ruta de la base de datos Norwind.

```
Const sSampleFolder = "C: \ ExcelData \"
Const sNorthwind = "C:\Archivos de programa\Microsoft Office\Office10\
muestras\Northwind.mdb"
```

#### El código a incluir es el siguiente:

```
'Establish a connection to the data source.
Dim sConnectionString As String
sConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=" & sSampleFolder & _
"Book7.xls;Extended Properties=Excel 8.0;"
Dim objConn As New System.Data.OleDb.OleDbConnection(sConnectionString)
        objConn.Open()
'Add two records to the table.
Dim objCmd As New System.Data.OleDb.OleDbCommand()
objCmd.Connection = objConn
objCmd.CommandText = "Insert into [Sheet1$] (FirstName, LastName)" &
            " values ('Bill', 'Brown')"
objCmd.ExecuteNonQuery()
objCmd.CommandText = "Insert into [Sheet1$] (FirstName, LastName)" & _
            " values ('Joe', 'Thomas')"
objCmd.ExecuteNonQuery()
'Close the connection.
objConn.Close()
```

**CopyFromRecordset** sólo funciona con los objetos de ADO Recordset. Un DataSet que se crea mediante el uso de ADO.NET no se puede utilizar con el método CopyFromRecordset.

# EJEMPLO INTEGRADO:

- Cree una nueva carpeta para mantener los archivos de Excel que se crearán para usted, y luego el nombre de la carpeta C:\Exceldata\.
- Inicie Visual Studio. NET. En el menú Archivo, haga clic en Nuevo y, a continuación, haga clic en Proyecto. En proyectos de Visual Basic, seleccione Windows Application.. Por defecto, se crea Form1.
- Añadir una referencia a la biblioteca de objetos de Excel. Para ello, siga estos pasos:
  - a. En el menú Proyecto, haga clic en Agregar referencia.
  - b. En la ficha COM, busque Microsoft Excel 10,0 Object Library y, a continuación, haga clic en Seleccionar.
  - c. Nota Si aún no lo ha hecho, Microsoft recomienda que se descargue y luego instalar el Microsoft Office XP Primary Interop Asambleas (PIAs).
  - d. En la ficha COM, busque Microsoft ActiveX Data Objects 2,7 Library y, a continuación, haga clic en Seleccionar.
- 4. Haga clic en Aceptar en el cuadro de diálogo Agregar referencias para aceptar sus selecciones.
- 5. Añadir un Combo Box de control y un botón de control a Form1.
- 6. Añadir el siguiente código a Form1.

## DESCARGAR CODIGO MUESTRA

Nota Si no se instala Office en la carpeta por defecto (C: \ Archivos de programa \ Microsoft Office), constante cambio en el código de muestra para que coincida con su ruta de instalación para Northwind.mdb.

Añadir el siguiente código a la parte superior de Form1.vb.

Imports Microsoft.Office.Interop

Presione F5, y divertase!

## **Fuentes:**

- > http://algoritmia.blogspot.com/2008/01/tabajar-con-office-xp-y-vb-net.html
- http://msdn2.microsoft.com/en-us/library/aa188489(office.10).aspx