

# Componente ZipMaster

El único inconveniente que tiene este componente es que necesita tener una librería dinámica al lado de nuestro ejecutable: **DelZip179.dll**. Según sus autores, podemos definir una directiva para añadir la librería de manera estática, aunque yo no he podido conseguirlo como comentaré más adelante.

## DESCARGANDO EL COMPONENTE DE SU PAGINA WEB

La página oficial del componente **ZipMaster** es esta:

<http://www.delphizip.org/>



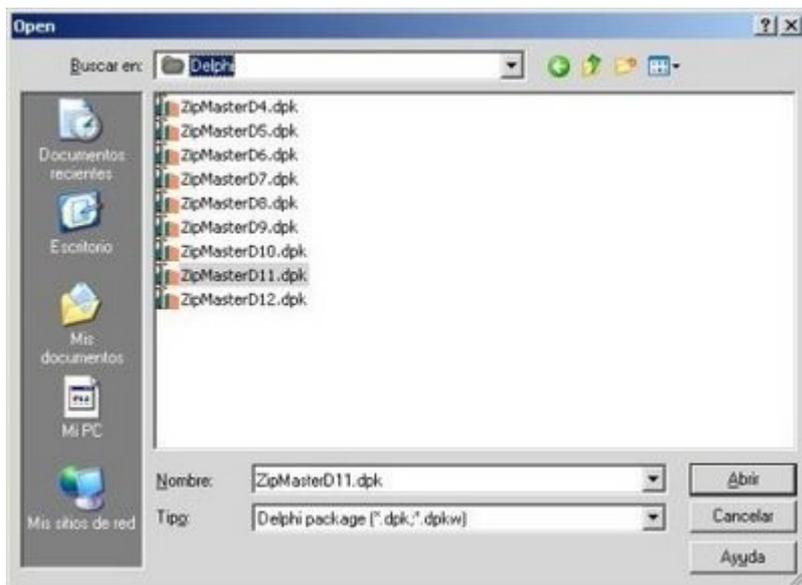
La última versión a fecha de este artículo es la **1.79**. Soporta desde Delphi 5 hasta Delphi 2007.

Nos bajamos el archivo [zm179setup1004.exe](#) que tiene un tamaño de 3,07 MB y lo instalamos:



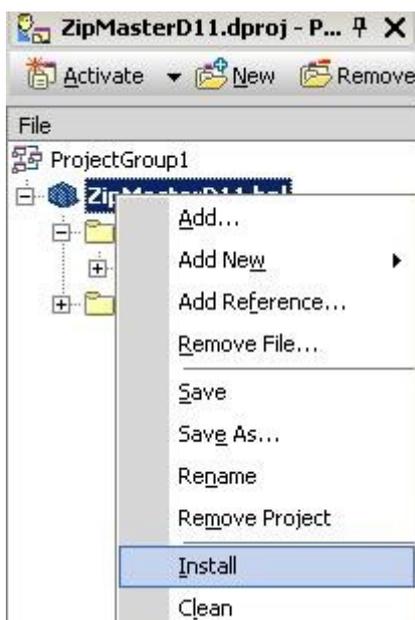
Después de instalarlo debemos instalar el paquete DPK en nuestra versión de Delphi habitual con los siguientes pasos:

1. Seleccionamos **File** -> **Open** y en el cuadro de búsqueda que aparece seleccionamos en el campo **Tipo** los archivos de tipo **DPK**:

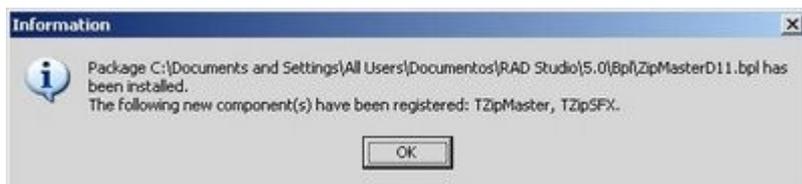


2. En mi caso voy a instalar la versión **ZipMaster11.dpk** que corresponde a la versión 2007 de Delphi.

3. Seleccionamos el paquete abierto con el botón derecho del ratón y seleccionamos **Install**:



4. Si todo ha ido bien aparecerá este mensaje:



Una vez instalado debe aparecer en la paleta de componentes:



## AÑADIENDO EL COMPONENTE A NUESTRO PROYECTO

Insertamos el componente **ZipMaster** en un formulario y lo llamamos **Zip** para abreviar:



También debemos vincular el directorio de búsqueda del proyecto al directorio donde se encuentra la unidad **ZipMaster**. Esto se hace seleccionando **Project -> Options** y en la sección **Directories/Conditionals** pulsamos el botón [...] a la derecha del campo **Search Path** y añadimos:

D:\CodeGear\RAD Studio\5.0\Componentes\ZipMaster\

Suponiendo que sea ese el directorio donde he instalado el componente **ZipMaster**. También sería una buena costumbre copiar la librería **DelZip179.dll** del directorio:

D:\CodeGear\RAD Studio\5.0\Componentes\ZipMaster\DLL\

a donde tengamos nuestro proyecto, aunque no es necesario ya que se encuentra en C:\Windows\System32\. Es bueno llevarlo al lado del ejecutable cuando tenemos que instalar el programa en otros ordenadores.

## CÓMO COMPRIMIR ARCHIVOS CON ZIPMASTER

Ahora supongamos que quiero comprimir un par de hojas de cálculo que se encuentran en la carpeta **D:\prueba\**:

Nombre	Tamaño
calculo_cuotas.xls	228 KB
Prestamo Frances - Cuo...	382 KB

Este sería el proceso para comprimir todos los archivos de esa carpeta:

```

Zip.DLLDirectory := ExtractFilePath( Application.ExeName );
Zip.ZipFileName := 'D:\prueba\calculos.zip';
Zip.TempDir := 'D:\prueba\';
Zip.FSpecArgs.Clear;
Zip.FSpecArgs.Add( 'D:\prueba\*.*' );
try

```

```
Zip.Add;
except
  raise exception.Create( 'Error al comprimir los archivos.' );
end;
```

He realizado los siguientes pasos:

1º Le he especificado el directorio donde se encuentra la librería dinámica **DelZip129.dll** con la propiedad **DLLDirectory**.

2º Con la propiedad **ZipFileName** le indico como se va a llamar el archivo zip que voy a crear.

3º Aunque no es obligatorio, si es recomendable decirle el directorio temporal donde procesar los archivos temporales (**TempDir**). Lo normal es que sea el mismo directorio donde se están comprimiendo los archivos, por si se detiene el proceso y deja archivos temporales.

4º Por último vamos añadiendo los archivos que queremos comprimir:

```
Zip.FSpecArgs.Clear;
Zip.FSpecArgs.Add( 'D:\prueba\*.*' );
```

También podemos añadir una contraseña al archivo comprimido de este modo:

```
Zip.Password := '1234';
```

## CREANDO ARCHIVOS AUTOEXTRAIBLES

Para crear un archivo comprimido y autoextraible hay que introducir en nuestro formulario el componente **ZipSFX**:



Suponiendo que el componente se llame **ZipSFX** escribimos el siguiente código que convierte un archivo **zip** ya comprimido en uno ejecutable:

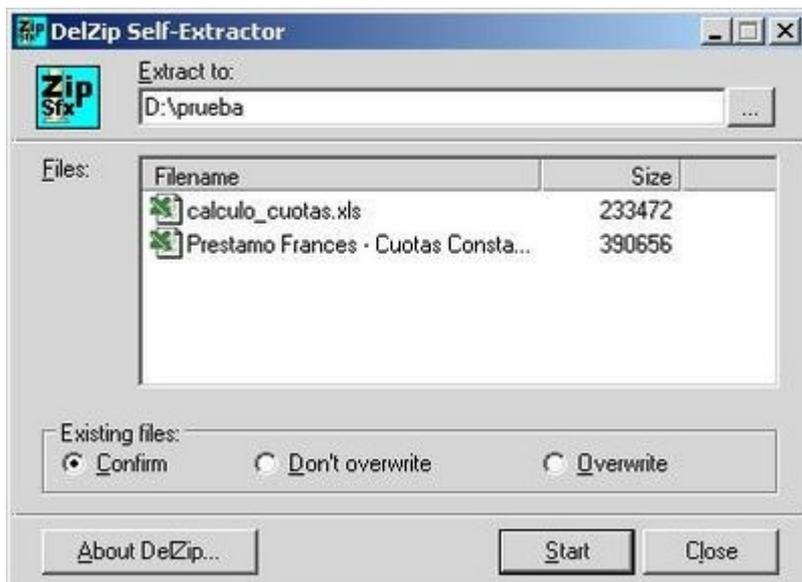
```
ZipSFX.SFXPath := 'D:\CodeGear\RAD Studio\5.0\Componentes\ZipMaster\Res\';
ZipSFX.SourceFile := 'D:\prueba\calculos.zip';
ZipSFX.TargetFile := 'D:\prueba\calculos.exe';
ZipSFX.Convert;
```

El archivo que hace de autoextraible procede del directorio:

```
..\ZipMaster\Res\
```

El cual contiene dentro el archivo **sfx\_std.zip** que hay que descomprimir en el mismo directorio.

Al ejecutar el archivo autoextraible aparece esta ventana:



Supuestamente también nos permite crear el archivo autoextraíble en varios idiomas, incluido el español. Esto es lo que habría que añadir:

```
ZipSFX.SFXLanguage := 'es';
```

Lo que ocurre es que al ejecutarlo da este error:



Y ya he comprobado de que se encuentran estos archivos en ese directorio:

**Dzsfxes.res**

**Dzsfxes.bin**

Pero no hay manera de que me haga caso.

## COMO DESCOMPRIMIR ARCHIVOS CON ZIPMASTER

Para descomprimir uno o más archivos con **ZipMaster** hacemos esto:

```
Zip.DLLDirectory := ExtractFilePath( Application.ExeName );
Zip.ZipFileName := 'D:\prueba\calculos.zip';
Zip.TempDir := 'D:\prueba\';
Zip.ExtrBaseDir := 'D:\prueba\';
Zip.FSpecArgs.Clear;
Zip.FSpecArgs.Add( '*.*' );
try
    Zip.Extract;
except
    raise exception.Create( 'Error al descomprimir los archivos.' );
end;
```

## INTENTANDO AÑADIR LA LIBRERÍA DELZIP179.DLL

En la documentación de este componente nos indica que si añadimos la directiva:

```
{ $DEFINE STATIC_LOAD_DELZIP_DLL }
```

en nuestro proyecto ya no necesitamos la librería **DelZip179.dll** al lado de nuestro ejecutable. Aunque he probado a quitar esta DLL y al ejecutarlo me da el error:



Que me lo expliquen.

## CONCLUSIONES

Pese a las dos cosas que he intentado hacer y que no me han funcionado he de reconocer que el rendimiento de este componente es excelente y tiene una cantidad de opciones impresionantes

---

<http://delphiallimate.blogspot.com/2008/11/el-componente-zipmaster.html>

21 noviembre 2008

Pruebas realizadas en RAD Studio 2007.