

Tutorial

“Utilizar la conexión a Firebird de un componente FIBplus en Report Manager”

Descripción:

En el siguiente tutorial se detalla la utilización de los componentes FIBplus en su versión 6.9.6, los cuales se utilizan para conectarse a una base de datos Firebird 2.1.3, y luego utilizar esta conexión para cargar los datos de una tabla específica a un reporte diseñado con Report Manager 2.9b; todo esto en Delphi 2007.

Requerimientos:

1. Tener instalados los siguientes componentes en Delphi 2007:
 1. FIBplus 6.9.6
 2. Report Manager 2.9b
2. Diseñar un reporte con Report Manager, utilizando una conexión “FBCONNECTION”.

Inicio:

Teniendo instalados los componentes en Delphi, creamos un nuevo proyecto “VCL Forms Application” como se detalla en la figura No. 1:

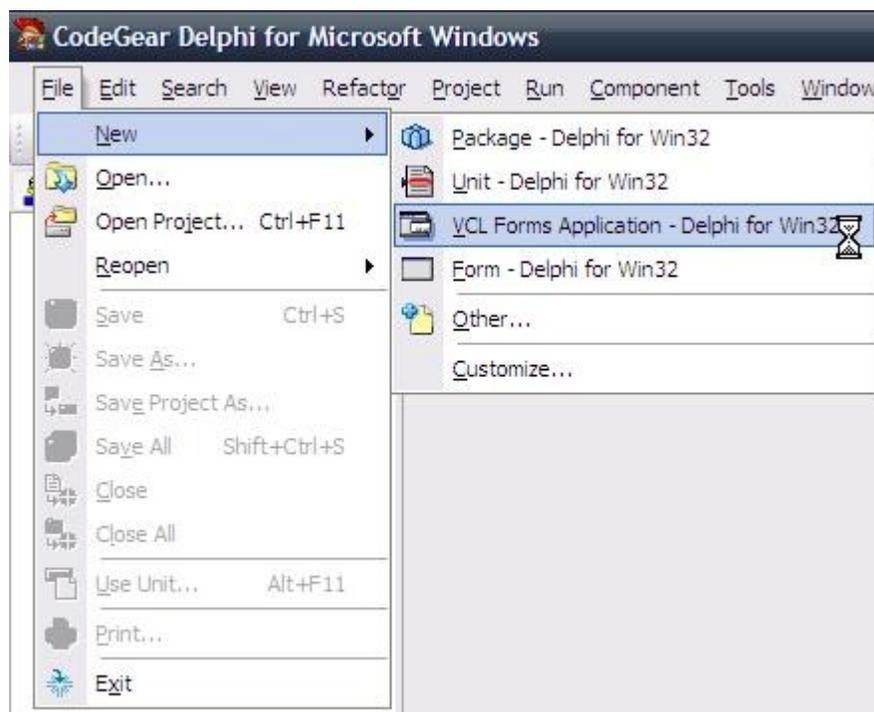


Figura No. 1. Nueva Aplicación “VCL Form”.

Luego, teniendo listo el formulario por defecto que nos crea el IDE, según la figura No. 2, procedemos a guardar la unidad y el proyecto, como se detalla en las figuras No.4 y No 5.:

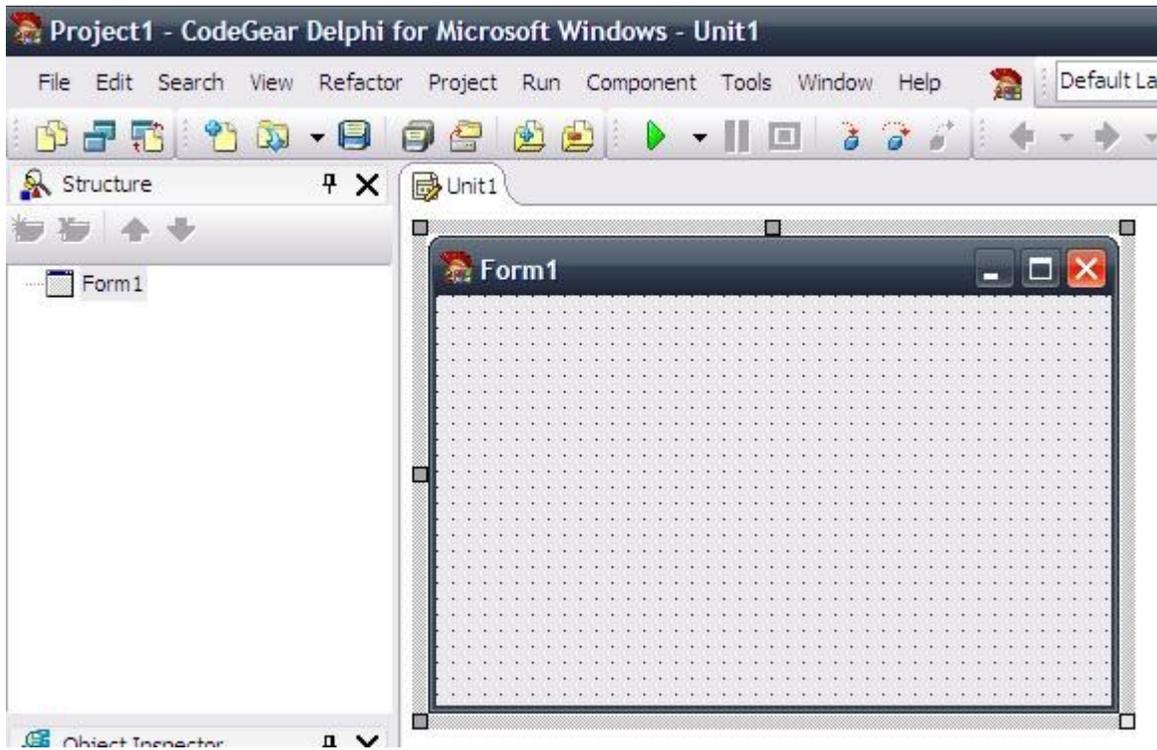


Figura No. 2 Formulario creado por el IDE

Siguiendo, guardarnos el proyecto:



Figura No 3. Icono "Save All" para guardar todo el proyecto.

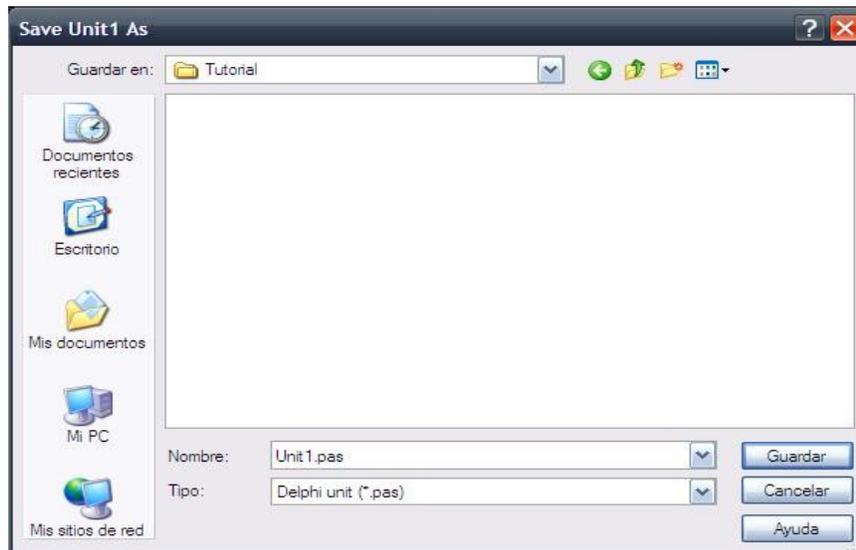


Figura No. 4 Guardar la unidad "Unit1.pas"

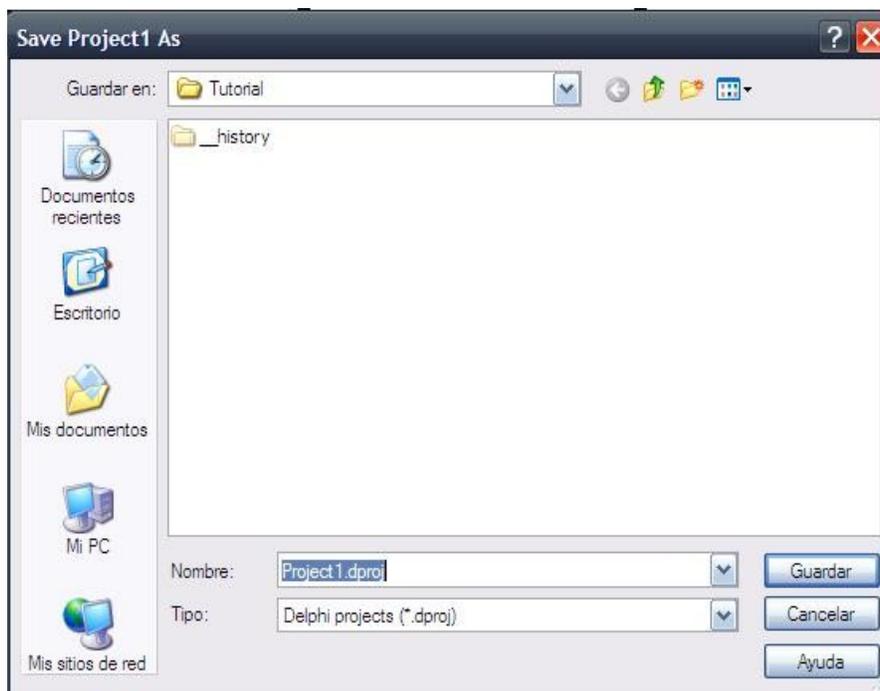


Figura No. 5 Guardar el proyecto “Project1.dproj”

Ahora debemos agregar al formulario los siguientes componentes, según las figuras No. 6, 7 y 8; los cuales se encuentran en las paletas de herramientas: FIBPlus, Reportman y Standard respectivamente.



Figura No. 6 Componentes TpFIBDatabase, TpFIBDatase y TpFIBTransaction

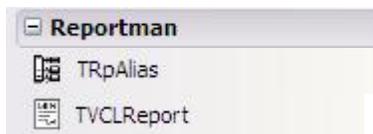


Figura No. 7 Componentes TrpAlias y TVCLReport



Figura No. 8 Componente TButton

Teniendo los componentes agregados al formulario, los ordenamos como se muestra en la figura No. 9, y vamos a proceder a configurar las propiedades de cada uno de ellos, según las que vamos a utilizar

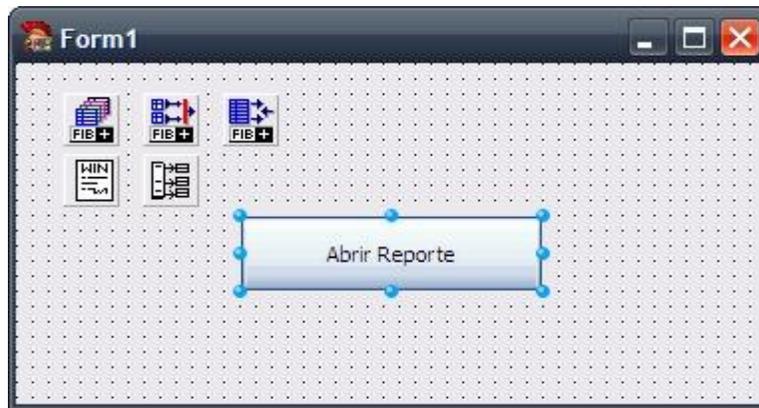


Figura No. 9 Formulario con los componentes

Configurando los componentes:

Cambiar las siguientes propiedades de los componentes y en el orden que sigue:

pFIBDatabase1:

1. DefaultTransaction = pFIBTransaction1
2. DBName = <ruta de tu base de datos Firebird>
3. ConnectParams
 1. Password = <tu password>
 2. UserName = SYSDBA
4. LibraryName = fbclient.dll
5. Connectd = True

Para este componente, hay que disponer de la librería “fbclient.dll” en la ruta donde esta instalado Delphi, luego también debe incluirse al distribuir la aplicación. Si hay propiedades en que la información no corresponde, entonces en el último paso (5) al cambiar la propiedad a “True” nos mostrará un error.

pFIBTransaction1

1. Active = True

En este componente la propiedad “DefaultDatabase” cambia a *pFIBDatabase1* automáticamente cuando cambiamos la propiedad “DefaultTransaction” en el componente *pFIBDatabase*.

pFIBDataset1

1. Database = pFIBDatabase1
2. SQLs
 1. SelectSQL = <tu sentencia SQL, para este ejemplo es: “SELECT * FROM emple;”>
3. Active = True

Si existe algún error en la propiedad “SelectSQL”, al cambiar la propiedad “Active”, dará error.

VCLReport1

1. Filename = <ruta del reporte, para este ejemplo: “e:\db\datafb\prueba.rep”>

Las propiedades “ReportName”, “Title” son opcionales, en las cuales se puede cambiar el nombre del reporte y su respectivo título.

RpAlias1

Este componente es el que hará el enlace de datos desde el *pFIBDataSet1* hacia el *VCLReport1*, o sea que tomara los datos de la sentencia *SelectSQL* y los enviara al “conjunto de datos” que tengamos en nuestro reporte, entonces solo queda agregar una lista de *Alias*, como se muestra en la figura No. 10:



Figura No. 10 Propiedades del componente RpAlias1

Y luego de seleccionar la propiedad “List”, vamos a editar y agregar un *TrpAliaslistItem*, como se muestra en la figura No. 11.



Figura No. 11 Edición de TRpAliaslistItems

Teniendo agregado un *TrpAliaslistItem*, vamos a configurar las propiedades, como se muestra en la figura No. 12:

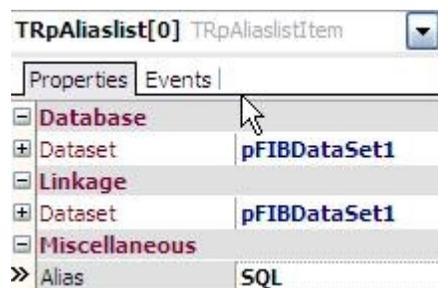


Figura No. 12 Propiedades de un TRpAliaslistItem

Cuando se configura la propiedad “Alias” del *TrpAliasListItem*, esta debe ser igual al nombre que le asignamos al *conjunto de datos* en el diseñador de Report Manager, para este ejemplo; según como se muestra en la figura No. 13, el *conjunto de datos* se llama “SQL”:



Figura No. 13 Conjunto de datos del informe en Report Manager

Teniendo configuradas las propiedades correspondientes de los componentes, vamos a codificar el evento *OnClick* del botón “Button1”, para esto, hacer doble clic sobre el mismo, y escribir el siguiente código, como se muestra en la figura No. 14:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    VC1Report1.Execute;  
end;  
end.
```

Figura No. 14 Código del botón “Button1”

Ejecutando el reporte:

Luego de haber configurado las propiedades de los componentes, procedemos a guardar el proyecto, haciendo click sobre el botón “Save All”, y ahora que ya esta listo lo podemos ejecutar haciendo sobre el botón “Run”, o presionando “F9”.



Figura No. 15 Reporte ejecutado con datos de la tabla emple

Conclusiones:

El anterior tutorial resalta la utilización del componente “*TRpAlias*” de la paleta de herramientas “Reportman”, el cual nos puede ser de mucha utilidad cuando no tengamos a disposición el componente de conexión a base de datos de los que dispone el componente “*TVCLReport*”, en su propiedad “*VCLReport1.Report.DatabaseInfo.Items[0]*”.

Para que la aplicación se ejecute, es necesario tener en la misma ruta las siguientes librerías: “*fbclient.dll*” y “*dbxfb.dll*”.

Autor:

José de Paz
Escuintla, Guatemala 2011.
Ahora en Medellin Colombia.
jose.depaz@gmail.com



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.