



Bienvenidos al mundo de ooHG!!!

Intentaré explicar el funcionamiento de la clase **ooSQL** para acceder mediante el control **xBrowse** a consultas de **MySql**.

Primeramente cabe aclarar que esta clase surge por una inquietud para facilitar la navegación sobre consultas de **MySql**. Así pues, con Vicente logramos (entre nos..., él la hizo y Yo fui su Tester!! Jejeje), desarrollar esta pequeña clase que asociada al control **xBrowse**, mejora enormemente los famosos y nunca bien ponderados ABMs.

Para ello, he creado un pequeño proyecto para ser incluido en los ejemplos de ooHG donde se visualiza el uso completo de esta Clase. Es decir: la navegación por el **xBrowse**, Altas, Bajas y modificaciones.

Entorno de desarrollo y ejecución

Sistema operativo: Windows XP SP2 Professional

OoHG: Versión 2010-04-27 (debería funcionar con versiones superiores)

PSPad: Editor de fuentes (conveniente instalarlo, ya que el Proyecto.ppr, esta incluido)

QAC: Entorno de compilación (Gracias Carozo!)

Navegador: Mozilla Firefox

MySQL: Versión 5.0.24^a (debería funcionar con versiones superiores)

MiniGui IDE: Versión 2005 © Roberto López (Si, utilizo una versión antiquísima y no la más adecuada que es **oIDE2 Plus** de Ciro y adaptada a ooHG. El motivo: herencia de las primeras aplicaciones de MiniGui). Pero esto también demuestra la versatilidad de ooHG o cualquiera de sus distribuciones.

Notas:

- En un entorno distinto, no podría asegurar su correcto funcionamiento.
- Se da por descontado la instalación de MySql
- El programa no altera la configuración predeterminada de MySql, pero utiliza la configuración de puerto, host, usuario y pass por defecto.
- No me hago cargo de ningún mal funcionamiento, OK? ... agradeceré las correcciones pertinentes (nos benefician a todos!)...

Características generales

La intención de este proyecto, es transmitir de forma lo mas simple posible el uso de MySql con un ABM básico desde la creación misma de la Base de datos y las Tablas necesarias, hasta la configuración de parámetros dentro de una aplicación. Por tanto, puede resultarle muy útil a los principiantes y aburrido a los entendidos.

La conexión a **MySql** se hace a través de la clase **TMySQLServer()** y para ello incluyo en el proyecto las librerías **LibMySql.DLL** (propia de **MySql** y que debe estar en el mismo lugar donde esta el EXE), y las librerías **LibDllMySql.A** y **LibFunMySql.A** ambas generadas para el compilador MINGW (Observar el proyecto QAC, donde se encuentran agregadas).

Algunos puntos interesantes:

- Conexión a MySql
- Creación de Base de datos y tabla
- Altas de registros
- Bajas de registros (en este caso, Habilitaciones/Inhabilitaciones)
- Edición de distintos tipos de campos (Numéricos, Texto, Date)
- Utilización de transacciones en MySql
- Creación y edición de archivos INI
- Utilización de ventanas (Form) de tipo CHILD
- Efectos de transparencia (Ventana Parámetros)
- Proyecto completo (fuentes, proyecto QAC, proyecto Pspad, recursos)
- Utilización de recursos gráficos
- Uso de una ToolBar dentro de un SplitBox
- Ayuda en formato CHM y PDF (para este último utilice el sitio Web <http://www.doc2pdf.net/>)
- Acceso a aplicaciones externas
- Utilización de múltiples **PRGs** y múltiples **FMGs**
- Definición de colores
- Los PRGs están auto documentados dentro de lo posible

- Utilización de distintos Iconos

Funcionamiento del ABM propiamente dicho

Lo primero a observar es el inicio de la aplicación en **Sistema.PRG**, donde veremos los pasos y verificaciones para establecer la conexión a **MySql** y la creación de la Base y Tabla si fuera necesario (lo hace la primera vez que arranca la aplicación y luego nunca más, a menos que borremos la Base de MySql). Este PRG esta bastante bien documentado así que no creo que amerite más que mirarlo. Y verán lo simple que resulta la conexión!!.

Lo importante es la variable **oServer** que contendrá la conexión a MySql, y por ello deberá ser pública. Para los que se inician, es como el establecimiento del RDDCDX para poder manejar DBFs. También se puede observar, que de lo que se trata SQL, es precisamente consultas (queries) para TOODOOO!!.. Esta es la parte que puede llegar a confundir un poco al principio, sobretodo para los que venimos del más puro **Clipper** y sus archiconocidas tablas DBFs súper poderosas. Pero con las pruebas verán que es muy fácil su entendimiento y empezarán a descubrir la potencia de SQL. Además...en la Interneeeee, existen ejemplos para cualquier cosa!.

En **FuncionesSql.PRG** tenemos por un lado las funciones generales de uso de la aplicación, entre ellas algunas curiosas como **CambioBarritaSQL()** o **CambioAcentoSQL()** que son necesarias en algunos puntos, ya que **MySql** es bien quisquilloso con algunos caracteres que son considerados especiales. O por ejemplo **d2c()** para el tratamiento de fechas (gracias Gustavo Asborno!), donde **MySql**, hace un tratamiento especial de las mismas. También existe la inclusión de un programa en C, que realiza el efecto de transparencias.

Merece un capitulo aparte, la famosa clase **ooSQL** que por no estar incluida todavía en la última versión de **ooHG**, la he colocado aquí. Por supuesto que en versiones futuras deberemos sacarla –o haré una actualización a este Ejemplo–, ya que estará dentro del núcleo de **ooHG** propiamente dicho. Aquí se puede observar toda la potencia de **ooHG** basada en objetos.

En **Parámetros.PRG**, tenemos la creación y edición de archivos INI. Algo muy útil si queremos parametrizar valores en nuestra aplicación de manera muy fácil. No es necesaria más explicación, solo verlo es más que suficiente.

Y finalmente **AbmClie.PRG**. Este es el núcleo principal de este proyecto.

Aquí podemos ver como iniciamos el formulario de ABM, donde lo importante es primero asegurarnos la consulta (query) que generará nuestro ABM (es como la DBF filtrada por lo que queramos). Observar que luego de la comprobación de que no hay error en la consulta, cargo la variable objeto oBaseClie, que será la que enlazará al control xBrowse para sincronizar la navegación del mismo.

```
oBaseClie:=ooSQL():New(oQuery)
```

Aquí radica la ventaja de esta clase, ya que la navegación se hace mucho más fácil dentro del xBrowse. Otro punto interesante a observar, es la actualización de oBaseClie, luego de modificar, agregar o incluso habilitar/deshabilitar un nuevo registro, donde es necesario aplicarlo para que el xBrowse muestre la edición realizada.

```
nRec:=oBaseClie:Recno()  
oBaseClie:oQuery:=oQuery  
oQuery:Goto(nRec)
```

También es importante mirar la búsqueda, ya que se pueden ver las características de ooSQL para modificar los punteros automáticamente a medida que se realiza la misma.

Con estas explicaciones básicas, se puede ver la facilidad que genera la aplicación de la clase ooSQL para ABMs relativamente simples. El resto será cuestión de mirar detalladamente los fuentes.

Conclusiones finales

Este proyecto solo pretende mostrar el funcionamiento básico de la clase ooSQL con MySql en ABMs simples. No pretende enseñar nada nuevo a programadores experimentados, ni considera que esta sea la mejor forma de administrar consultas. Solo pretende y ese es mi objetivo, esclarecer un poco más el funcionamiento de MySql con ooHG utilizando esta clase.

Soy un ferviente usuario de ooHG, y humildemente intento colaborar de la mejor forma posible con este fabuloso trabajo que es esta herramienta. Soy un convencido, que si todos podemos colaborar con 'algo', ya sea un ejemplo de código, o ayudando en los foros, o participando activamente, o incluso haciendo aportes económicos

voluntarios –que son tan necesarios–, sin duda llevaremos a ooHG a una potencia inimaginable.

Creo necesario agradecer a...

Roberto López,
Vicente Guerra,
Ciro Vargas,
Carozo de Quilmes,
José Miguel,
Gustavo Asborno,
Gerardo Pérez,
Ricardo Sassy,
Lucho Miranda,
Bruno Luciani,

Y tantos otros –perdón, no me alcanzarían las páginas–, incluyendo a las otras versiones de MiniGUI, que hacen muy poderoso este hermoso lenguaje!!!

Saludos cordiales,
Sergio Castellari