

Manual

Para crear Aplicación Multi-Dll

Con Clarión 6.3

Introducción

Ante de comenzar con este pequeño instructivo sobre la creación de aplicaciones Multi-Dll, quiero advertir sobre la existencia de algunos errores comunes que se presentan en el desarrollo de la misma, y en donde muchos programadores quedan frustrados con el tema y lo dejan abandonado, hago este aclarando por que yo fui uno de ellos, pero retome el tema con dedicación y con deseo de aprender y no me dejé vencer por la impotencia al no encontrar la forma de solución a estos errores, por eso comienzo limpiándote el camino y sacando las dudas de tu corazón.

Lo que me motiva hacer este manual o instructivo, es que tuve la oportunidad de leer varios manuales para la creación de aplicaciones muti-Dll y ningunos de ellos llenaron mi expectativa, porque no me supieron explicar con detalle los pasos a dar, ni los inconvenientes que me iba a encontrar en la creación de aplicación multi-dll. Por eso quiero ir explicando paso a paso todo los referentes al tema. Lo que espero es que este humilde manual llene tu expectativa y pueda adquirir los conocimientos básicos sobre el tema. Te prometo que si sigue cada paso como está establecido en este manual no será un programador frustrado sobre el tema de la Dlls.

Errores que se presentan en la creación de aplicación Multi-Dll

Unos de los errores más comunes que se les presenta a muchos programadores y se convierte en un dolor de cabeza es este error **cskeys.clw**, donde el compilador de clarión no dice que no encuentra este archivo, que no es más que un código con extensión Clw de clarión. Este error es producido por el témpate ACBfree y que venía con el témpate para la versión 5.0 y 5.5, aunque hay una versión del témpate para 6.3 no trae el archivo y ocasiona el error y no se encuentra por ningún lado, solamente en la versión viaja de clarión 5.0 o 5.5 se puede encontrar.

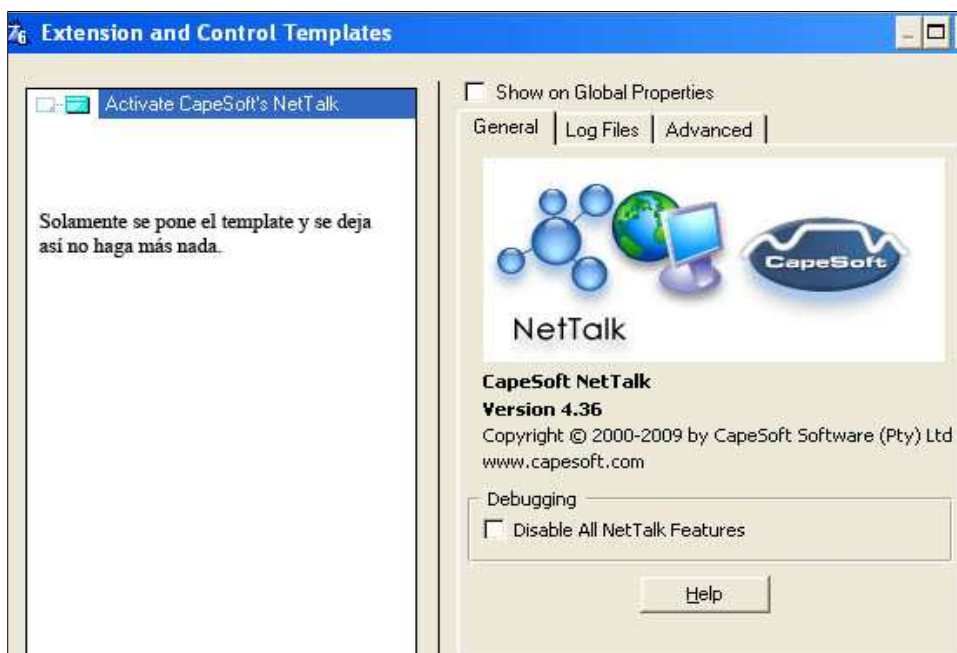
El témpate trae una extensión para agregar en tu DLL de datos la extensión se llama **"Exclude goblal classes generation in DLL"**. Se coloca en la propiedad global del Dll de datos o app de datos. Cuando se agrega esta extensión en el Dll de datos o app de datos el témpate no da errores. Pero si da errores en los Dll de procedimientos y en el exe o app de menú, donde presenta este error de errorStatusclass y otros errores.

En mi caso le puedo decir que hice todos lo posible para solucionar estos errores producido por este témpate y no pude dar con dicha solución, gracia a la sugerencia de un colega me dijo que la solución más adecuada era desinstalar el témpate, así mismo hice y no tuve solución, la alternativa final que se me ocurrió fue reinstalar clarión de nuevo partiendo de cero y así tuve solución, aunque te recomiendo que el trabajo es tedioso porque son muchas cosas que hay que volver a justar, Te digo como dice la Biblia
" **Los que aman a Dios todas las cosas le ayudan a bien.** Romanos 8:28

Ing. R. Pacheco
Analista Programador
Pachecoramon@hotmail.com
Santo Domingo, Rep. Dominicana

Mis recomendaciones es evitar instalar este template si piensa trabajar con sistema dll, si ya lo tiene por favor desinstálelo desde cero, para que te evite grande dolor de cabeza.

Por otro lado si tiene instalado el template de NETTALK de capesoft también procede muchos errores, pero si le agrega la extensión global Activate dll de datos o app de datos, los erros desaparecen, solo **activate** no hay que hacer mas nada. Observe la imagen.



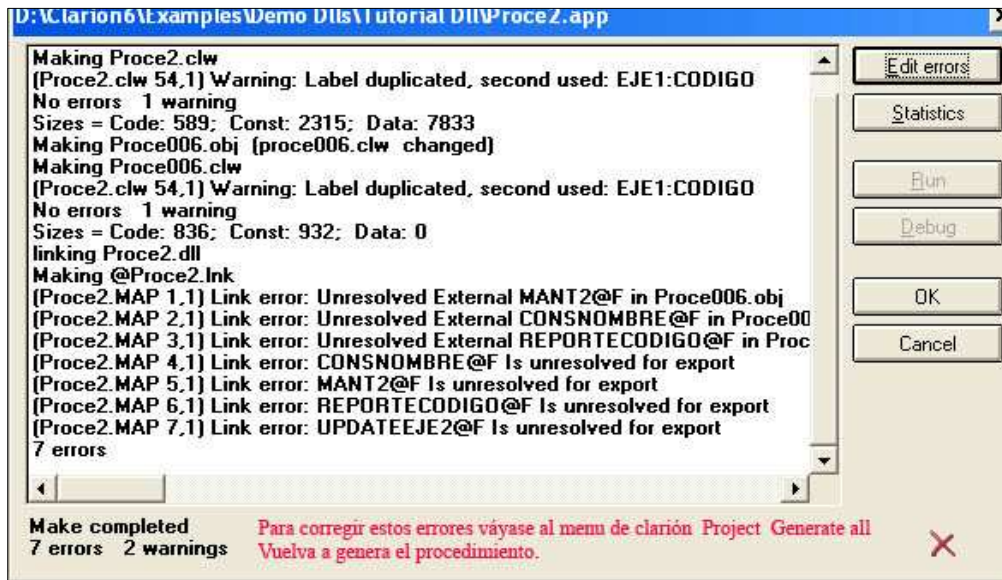
Error al crear menú con frame en dll de Procedimiento.

A la hora de programar aplicaciones grandes con varios módulos que tengan Menú y sub Menú debemos tener en cuenta de no usar el template frame de clarion En los Dll de procedimiento, sino usar Window o ventana luego crear en ella el menú Con el constructor de menú, esto te evitaría tener este error que se muestra en la imagen.



Error en la modificaciones de la DLL de Procedimiento

Cada vez que actualice modifique tu DLL de procedimiento puede ser que se te presente un error como el que vez en la imagen siguiente.



Para solucionar este error váyase al menú de clarión Project Generate all, esto significa vuelva a generar todos el proyecto o el sistema de nuevo.

Mantener las aplicaciones

Cuando hagamos alguna modificación a algún procedimiento sólo necesitamos recompilar la aplicación donde se encuentra el procedimiento. Si agregamos un procedimiento que es llamado en otra aplicación, debemos recompilar tanto la aplicación donde está el nuevo procedimiento, como aquella que lo llama. Cuando hacemos algún cambio en el diccionario o en las variables globales debemos recompilar todas las aplicaciones en estricto orden, iniciando con el *DLL de datos*, siguiendo con aquellos *DLLs* que son llamados por otros *DLLs*, después los *DLLs* que no son llamados por otros *DLLs* y finalmente los ejecutables

He tratado de limpiarte el camino para que vaya seguro a la creación de una aplicación multi-DLL, ahora sigue los pasos que describe este manual.

Creación de un sistema dll.

Cuando una aplicación tiene muchos procedimientos, es difícil darle mantenimiento. Lo mejor es dividir la aplicación en pequeñas aplicaciones (*DLLs*) ligadas por una aplicación maestra (*EXE*).

¿Qué es un DLL?

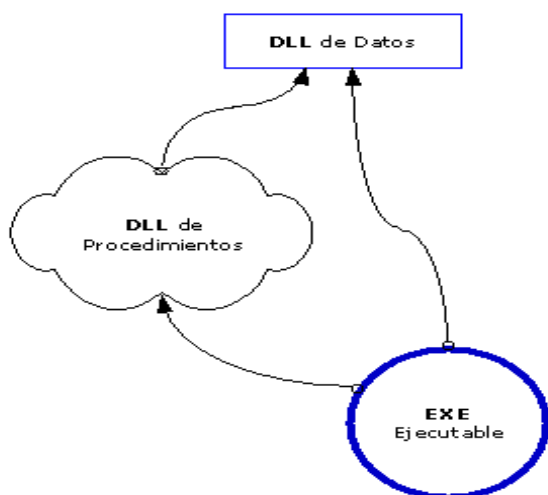
Dynamic Link Library – Librería de Ligado/Acceso Dinámico. Podemos definirlo de manera simple como un conjunto de procedimientos, disponibles para ser llamados por otros *DLLs* o *EXEs*.

Un sistema dll esta compuesto por tres partes principales.

Un DLL de datos o App de Datos. Contiene las definiciones de las variables globales y de las tablas del diccionario. Además, incorpora las clases usadas en la aplicación. NO puede contener procedimientos.

Un DLL de procedimientos o App de procedimiento. Donde se definen los procedimientos, ya sean browses, formas, reportes, procesos, etc.. Debe estar ligado al DLL de datos.

Un EXE. O App exe: es un ejecutable, que solo requiere un menú que llame a los procedimientos del DLL de procedimientos. Puede, adicionalmente, contener procedimientos. Debe estar ligado tanto al DLL de datos, como al DLL de Procedimientos.



Ventajas

En un proyecto grande, es usual que haya más de un programador involucrado. Al estar dividido en más de una aplicación, es muy fácil dividir el trabajo.

Facilidad de administración.

Compilación más rápida al sólo regenerar y recompilar los procedimientos de la aplicación que se modifique (Naturalmente, si se hace algún cambio en el diccionario, es necesario recompilar todas las aplicaciones asociadas).

Distribución de mejoras o actualizaciones de una manera más eficiente, ya que únicamente se redistribuyen los módulos que cambiaron.

Desventajas

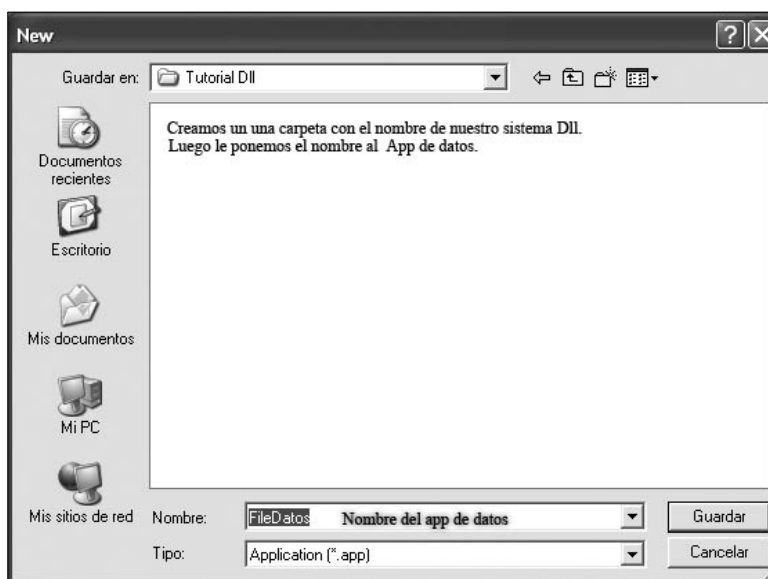
Puede salirse de control la administración si no se tiene cuidado en la división de los procedimientos (sobre todo, saber en cual aplicación está determinado procedimiento), pero no es una desventaja en los *DLLs* en sí, y con una buena planeación y administración no hay ningún problema.

Cuidado con los nombres

Normalmente, todas las aplicaciones de un proyecto (*DLLs* y *EXEs*) se almacenan en un mismo directorio. Debemos asegurarnos que los cinco primeros caracteres de cada aplicación sean diferentes. La razón es que al generar los archivos fuente, el compilador toma sólo los cinco primeros caracteres del nombre de la aplicación y le agrega tres dígitos consecutivos.

Creación del Dll de Datos o app de datos.

Los primero que debe hacer es crear una carpeta de trabajo en dentro del directorio de clarión Ejemplo Tutorial dll, luego debemos crear un diccionario de datos con los campos que usted desee que tenga su sistema. Luego creamos el nombre de nuestro app de datos para eso váyase al menú de clarión file y seleccione New póngale un nombre en mi caso yo le puse filedatos, trate de utilizar nombre cortos cuando este trabajando con dll. La aplicación debe verse como se observa en la imagen.



Variables Globales

Si tenemos variables globales, lo recomendable es definir las en el diccionario, como una *tabla tipo Global*. Si nuestra aplicación original contiene variables globales, podemos (debemos) copiarlas, ya sea a nuestra DLL de datos, o directamente al diccionario.

Normalmente, a las variables globales anteponemos un prefijo *GLO*, pero nuestro archivo *TXD* ya incluye un prefijo *GLO* (en la línea que inicia con "*Globales FILE*". Debemos eliminar de cada variable el prefijo *GLO* manualmente (todas las marcas que aparecen después de la primera línea).

Cargando los datos de la aplicación

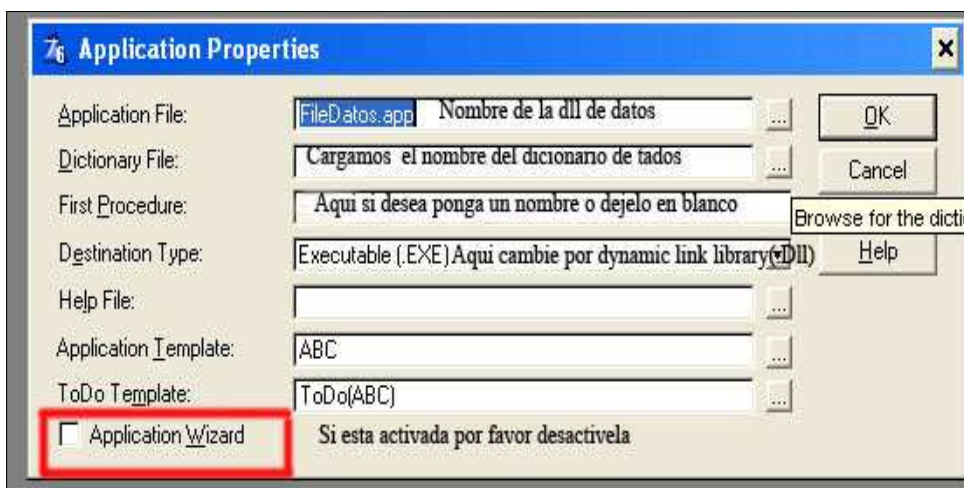
Application file: Se coloca el nombre de la dll de datos

Dictionary File: Se carga el nombre del diccionario de datos

First Procedure: Aquí se coloca el nombre del primer procedimiento, es opcional en este caso, lo puede dejarlo vacío como llenándolo.

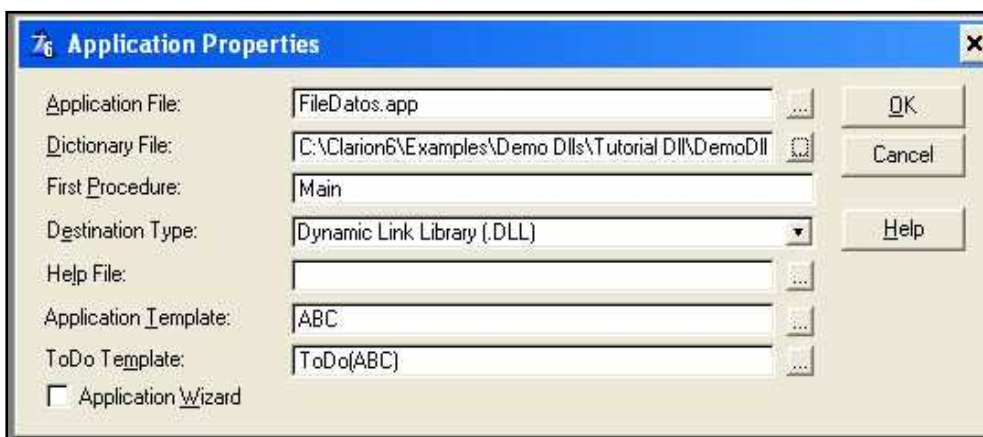
Destination Type: aquí se cambia ejecutable por dynamic link Library(DLL)

Todos lo demás se queda tal como se observa la imagen



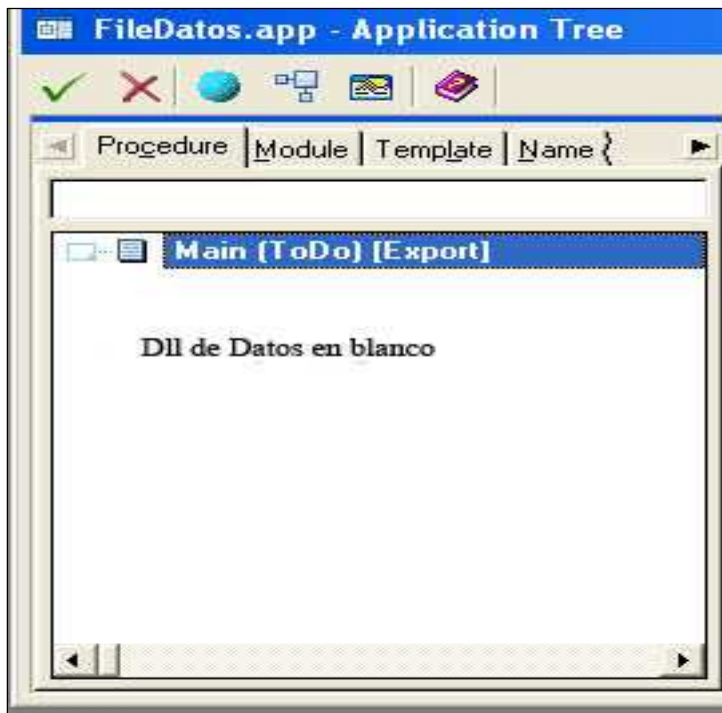
Luego de cumplir con las indicaciones presione el botón ok

Ahora su aplicación debe mostrarse como se observa en imagen de abajo.



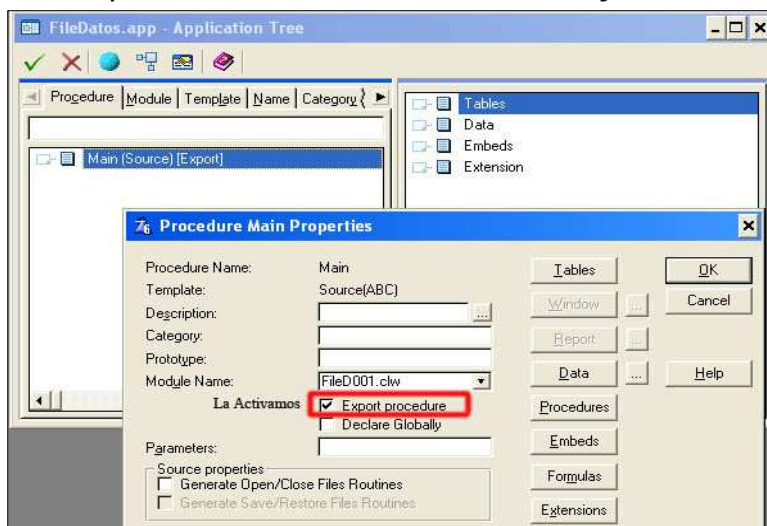
Si todo está bien presione el botón Ok.

Inmediatamente su aplicación debe aparecer como se observa en la siguiente imagen

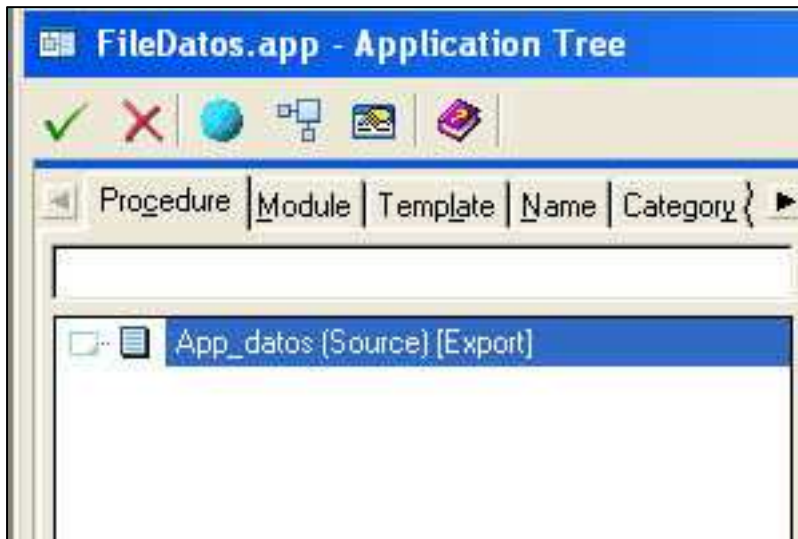


Aclarando

Como muestra esta imagen, aquí vemos el nombre del (**ToDo**) en blanco, a mi se me escapo y tome el nombre main por error, ahora quiero darle otro nombre lo llamaré App_datos, para cambiarle el nombre vayamos a la opción de clarión aplicación Propiedad, donde dice first procedure colocamos el nombre de App_datos , posiblemente le aparezca ahora app_datos como (Todo), seleccione una opción de témplate para ponerle en (Todo) déle doble clic en encima del todo para que aparezca el listado De témplate interno de clarión, en mi caso yo elegí source, ahora nuestra imagen debe haber quedado como se muestra más abajo.



Es recomendable activar la opción que dice export procedure, esto significa que Nuestra dll será llamada por el otro procedimiento, luego que tomamos source nuestra imagen debe estar como se muestra aquí



Configuración de la Propiedades Global

Para trabajar en modalidad dll es importante la configuración de la propiedad global en los tres procesos principales.

Desde el menú de clarión seleccionamos la opción del globito del planeta tierra en verde Que identifica la Global Propiedad, así como se vez en la imagen.



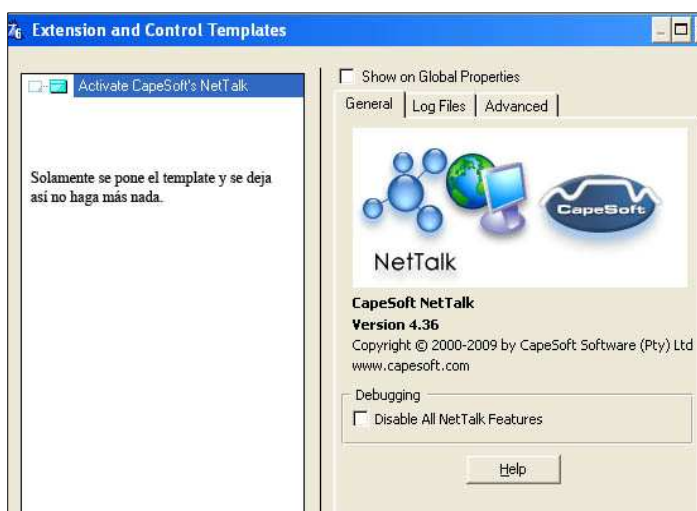
Como estamos trabajando con dll de datos o app de datos, elegimos directamente la opción **File control** de la propiedad global, entonces procedemos activar las dos opciones del círculo rojo, luego donde dice external activamos las dos casillas, como se muestra en la imagen de abajo.



Si todos esta bien presione el botón Ok.

Agregando t  mplate si lo amerita

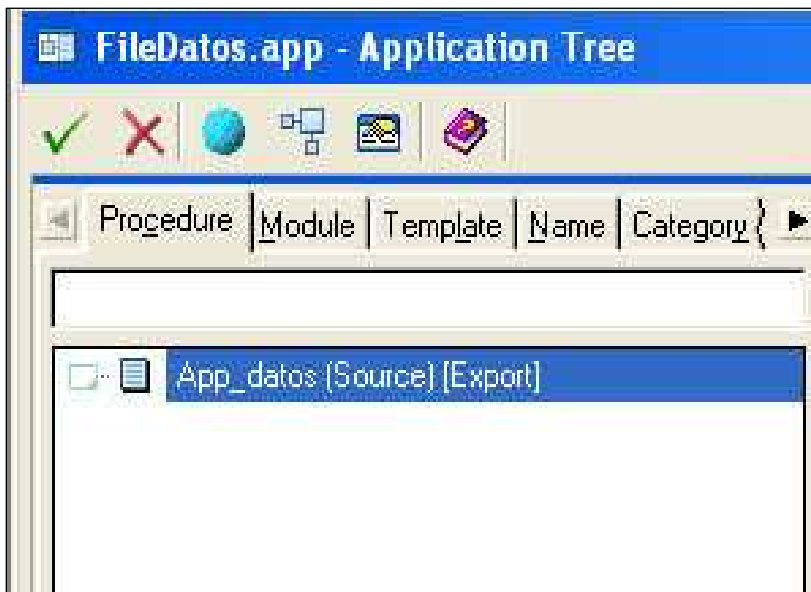
El app de datos es donde usted tambi  n debe colocar todos aquellos t  mplate globales Que lleva su sistema. En mi caso yo tengo instalado el t  mplate NetTalk de Capsoft, si compilo mi dll de datos o app de datos sin este t  mplate mi sistema va a arrojar varios errores, para evita eso coloco en la propiedad global la extensi  n active del t  mplate NetTalk, as   como se muestra en la imagen de abajo.



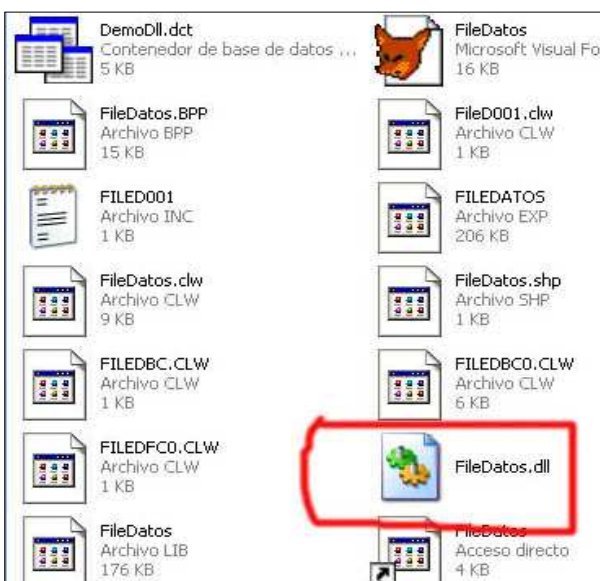
Solo coloque la extensión y no haga más nada. Ahora si usted no tiene instalado en el clarión este témpate no se preocupe su sistema no dará error.

Si tiene instalado el témpate de ABCFree hay muchas posibilidad de que su sistema le de error y no funcione.

Si ya terminamos de configurar nuestro Dll de datos o app de datos volvamos a nuestra aplicación como se muestra en la imagen.



Ya nuestro dll de datos o app de datos esta lista para ser compilada si usted ha seguido los pasos tal como se han descripto aquí, entonces compile su aplicación, luego que termine váyase a la carpeta de trabajo a ver su dll creada, tiene que tener algo parecido como se muestra en la imagen de abajo.

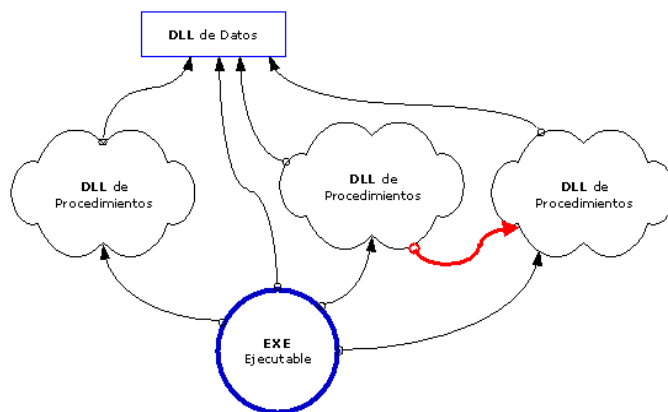


Ya hemos terminado nuestro Dll de dato o App de datos ahora procedemos a crear los Dll de procedimientos o App de Procedimiento.

El DLL de Procedimiento o App de Procedimiento

Un DLL de procedimientos o App de procedimiento. Es donde se definen los procedimientos, ya sean browses, formas, reportes, procesos, etc.. Debe estar ligado al DLL de datos.

Este esquema básico puede crecer para contener tantos *DLLs* de procedimientos como necesitemos, todos ellos ligados al *DLL de datos*.



En mi caso yo voy a crear dos dll de procedimientos, ya que en mi diccionario tengo Dos tablas diferentes quiero crear una dll para la tabla1 y otra para la tabla 2.

Pero ante quiero poner algo en claro, si usted ya tiene su sistema hecho en clarión y lo que quiere dividirlos en varias dlls, quizás el trabajo le sea más fácil porque solo tiene que importar las parte que desea convertir en dll de procedimiento, ahora en mi caso voy a crear un sistema nuevo.

DLL de Procedimientos Creamos una nueva aplicación llamada Proce1 (Va a contener un menú con opción de mantenimiento, browses, formas, otra opción de consultas y reportes. usando el mismo diccionario. Lo primero que hacemos es ir al menú de clarión en la opción Aplicación, New o nueva, póngale un nombre que sea corto luego tiene que verse como se muestra en la imagen de abajo.



Haciendo cambio a nuestra aplicación debe estar como se muestra en la imagen



Si todos esta correcto por favor presione el botón OK para continuar con nuestro desarrollo, luego nuestra aplicación debe estar así como se muestra en la próxima imagen.



Procederemos ahora a crear nuestro primer procedimiento, este será un menú que contiene el módulo de la tabla 1. Este constará de las siguientes opciones mantenimiento, consultas y reportes. Algo importante que se debe tener en cuenta es que si va a crear un menú en el dll de procedimiento no debe usar el frame del témpate de clarion porque le dará error, use una Windows o ventana y diseñe su menú a mano con el constructor del mismo y le quedara perfecto.

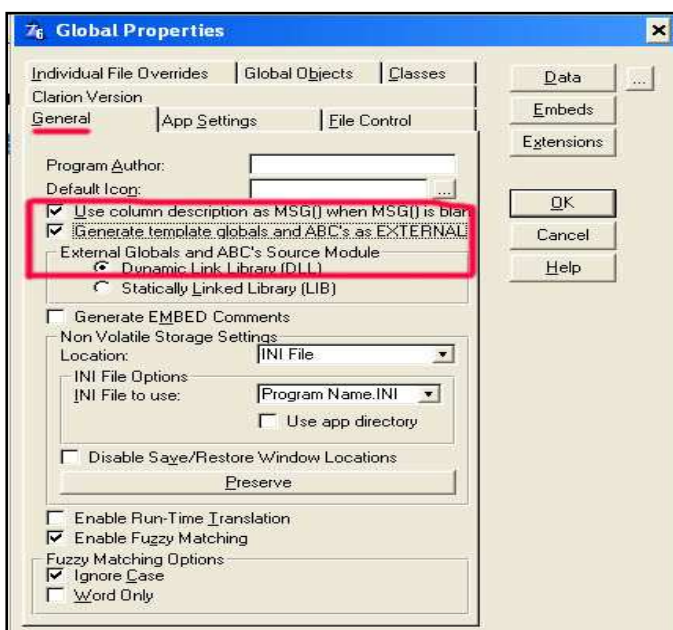
Hasta ahora si se llevo de mis consejos su aplicación de dll de procedimiento le debe haber quedado como se muestra en la siguiente imagen



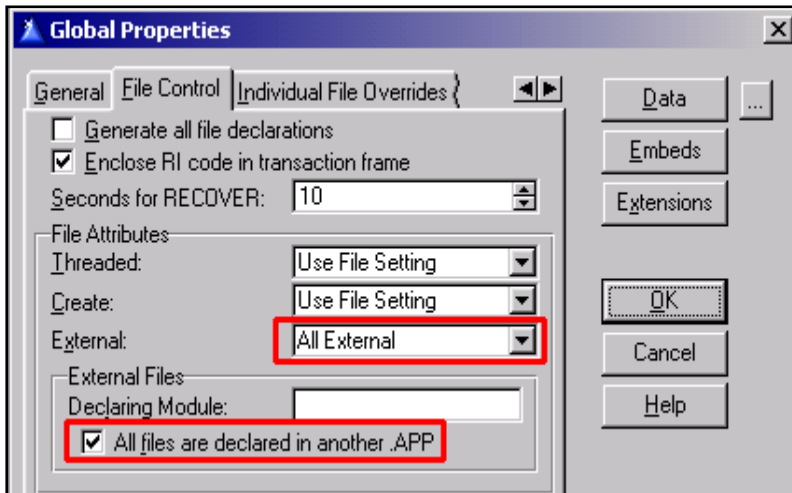
Aquí se muestra el árbol con el menú y todos los procedimientos del sistema, lo que no toca por ahora es darle un poco de estética a cada procedimiento.

Configuración de la Propiedad Global

una vez creada nuestra aplicación, presionamos el botón *Global* y activamos la opción **Generate template globals and ABC's as EXTERNAL**. Esta opción indica que tanto las variables globales declaradas por los t  mplates, como las clases ABC usadas en nuestra aplicaci  n sean definidas como "externas", es decir, declaradas en otra aplicaci  n. Debemos indicar que est  n declaradas en un *DLL* (que es nuestro *DLL de datos*). Si no activamos esta opci  n, al compilar marcar   una gran cantidad de errores debido a variables, propiedades y m  todos declarados por duplicado.

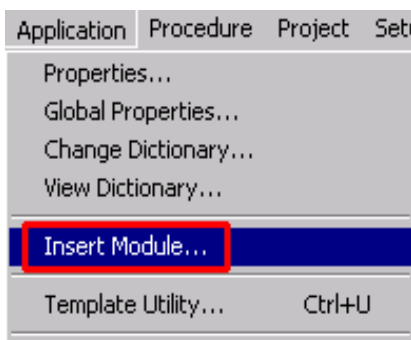


A continuación, en el **Tab File Control** definimos también que todas las tablas son externas (*All External*) y activamos la opción **All files are declared in another .APP** (Todos los archivos están declarados en otra aplicación), que es también nuestro *DLL de datos*, muestra aplicación debe quedar como se observa en la imagen.

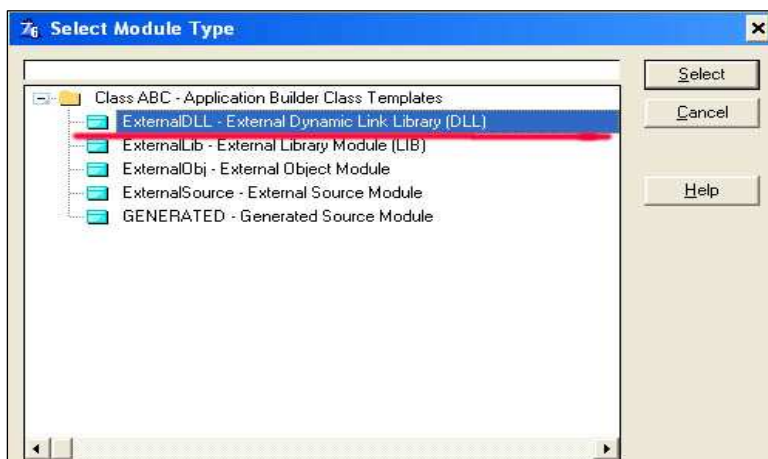


Insertar o Ligar al DLL de datos

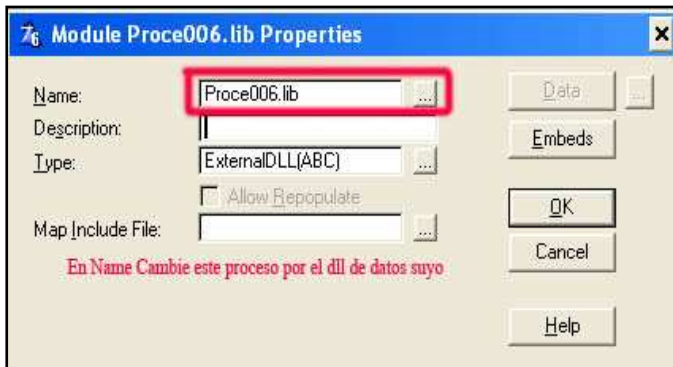
Para poder ligar nuestro *DLL de procedimientos* con el *DLL de datos*, seguimos estos pasos: En el menú *Application*, elegimos la opción *Insert Module...*



Inmediatamente aparece la ventana de selección de tipo de modulo



Elegimos la **opción externalDLL – External Dynamic Link library (dll)**
Inmediatamente sale otra ventana como se observa más abajo

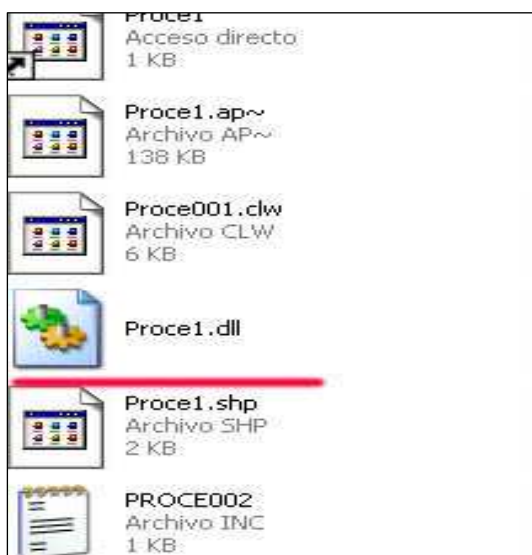


Cuando aparece esta ventana procedemos a hacer el cambio del proceso
En el name o nombre buscamos el nombre de nuestro dll de datos y lo colocamos aquí
Luego presionamos el botón Ok. Ahora nuestra aplicación debe quedar como se
Observa en la imagen de abajo.



Compilar nuestro dll de procedimiento

Una vez terminada nuestro dll de procedimiento procedemos a compilar la aplicación, obtendremos un *DLL* llamado *Proce1.dll* ligado a nuestro *DLL de datos* (*filedatos.dll*).
Le recomiendo que se ya a su carpeta de trabajo en mi caso seria:
D:\Clarion6\Examples\Demo DLLs\Tutorial Dll y observe su archivo dll creado como se muestra en la imagen.



Notas importantes

Quiero aclarar dos puntos importantes en el diseño de dll de procedimiento

1) Primer punto:

Si su aplicación ya esta creada en clarión y desea dividirla en varios procedimientos de dll, solo tiene que importar los datos creado en su nuevo procedimiento de dll.

Importar los procedimientos

Para traer de la aplicación original los procedimientos ya generados elegimos, del menú *File*, la opción *Import From Application...*



Escogemos de la aplicación original (en nuestro ejemplo, Minombre *.app*) y marcamos todos aquellos procedimientos que deseamos importar. En lo personal, prefiero hacerlo desde la vista *Name* (alfabético por nombre) o *Category* (por categoría de procedimiento). en la aplicación general ya creada, importe todos los procedimientos que desea que forman partes de su nuevo proce1.pp o dll de procedimiento.

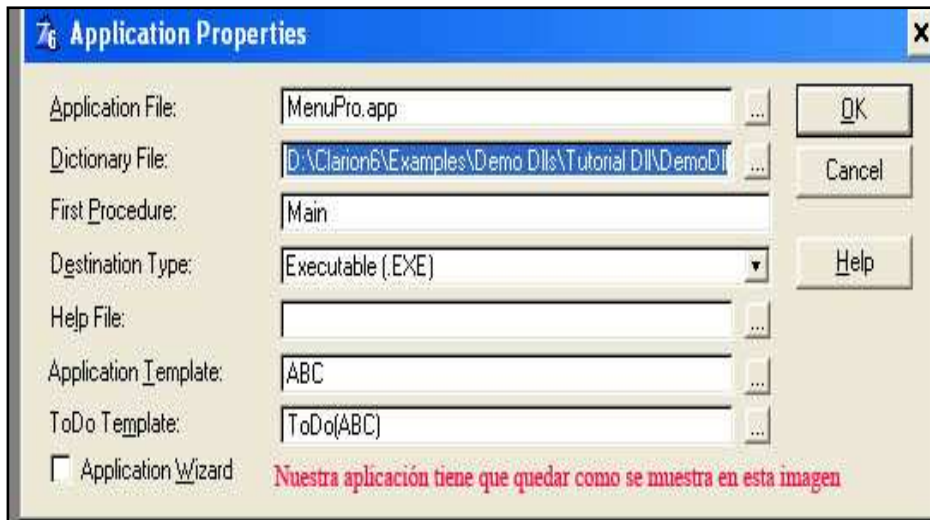
2) Segundo Punto

En este punto podemos compilar nuestra aplicación. Ahora como dije anteriormente Nuestra aplicación va ha tener dos procedimiento de dll, uno para la tabla 1 y otro para la tabla 2, serán dos módulos para que tenga una idea clara del futuro a la hora dividir su aplicación en varias Dll de Procedimiento.

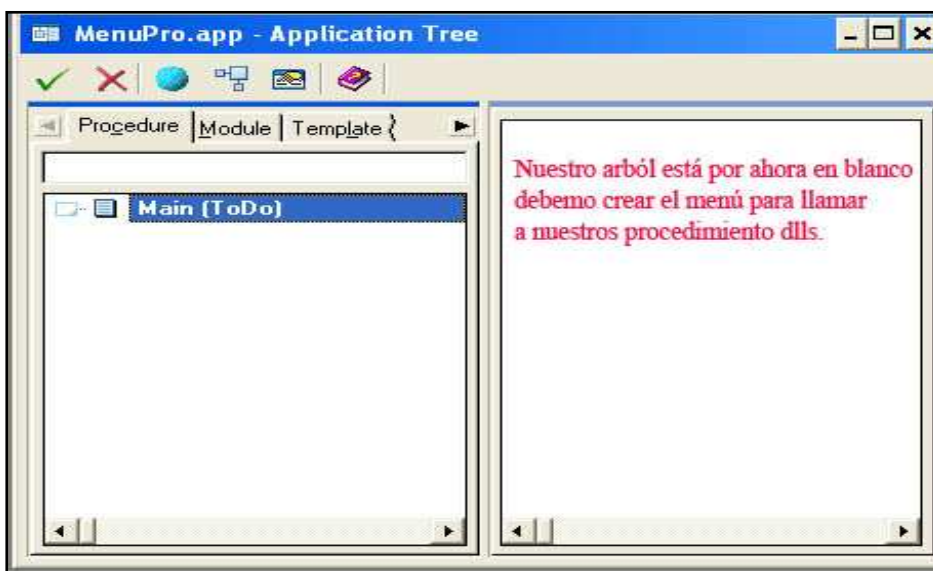
Quiero pedirle por favor que repita los mismos pasos para el procedimiento proce2.app iguales como se hicieron para el proce1. app.

Creación de la EXEs – Aplicación Ejecutable

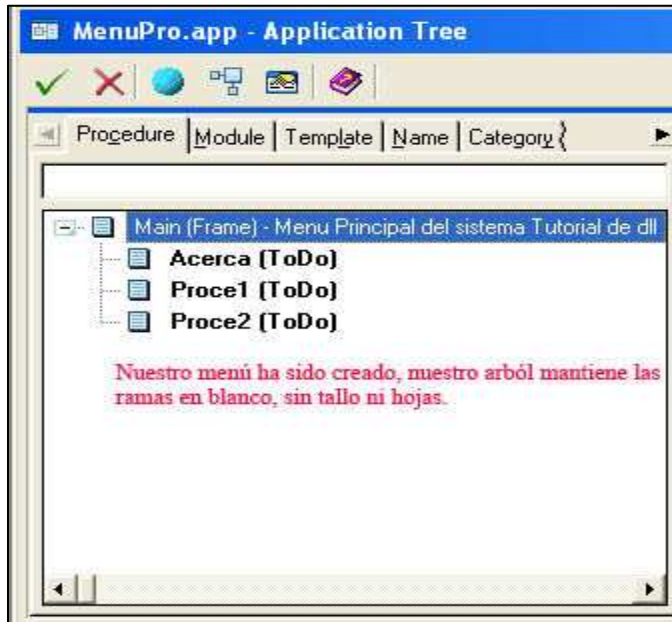
Creamos una nueva aplicación para convertirse en nuestro ejecutable, llamada *MenuPro.app* usando el mismo diccionario de datos, pero especificando un procedimiento como el primer procedimiento (*Main* es el nombre predeterminado) y nuevamente dejando apagada la opción de *Application Wizard*.



Luego presione el botón OK, para continuar con su exe.app en este caso lo es llamado MenuPro, su aplicación debe haber quedado como se muestra en la imagen de abajo.



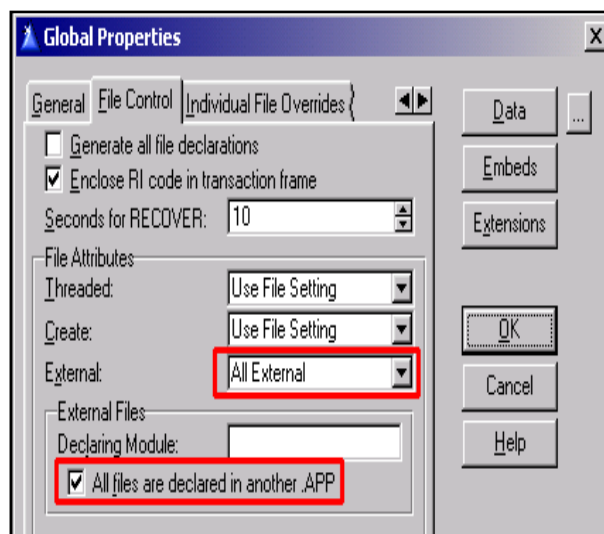
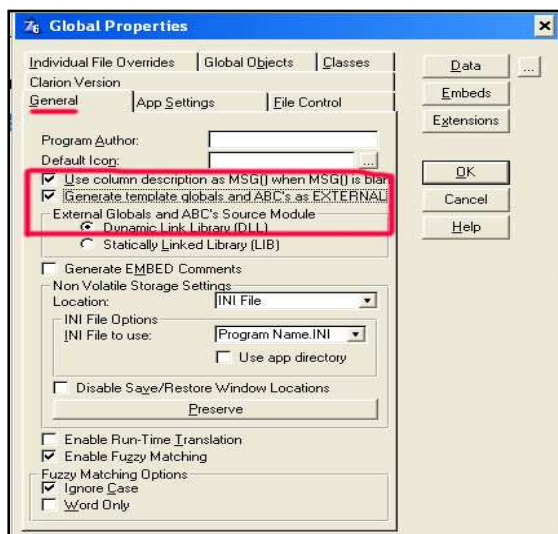
A continuación creamos nuestro menú con las opciones que deseamos que cargue, En este caso llamará a nuestros dos procedimientos proce1.dll y proce 2.dll Nuestro menú está quedando así como se muestra en la siguiente imagen



Cuenta con tres procedimientos, son: acerca será una ventana tipo Window en ella se describe brevemente el sistema sistema.
Proce1 y proce2 en estas opciones serán de tipo external, porque llamará a nuestras dos dll anteriormente creadas.

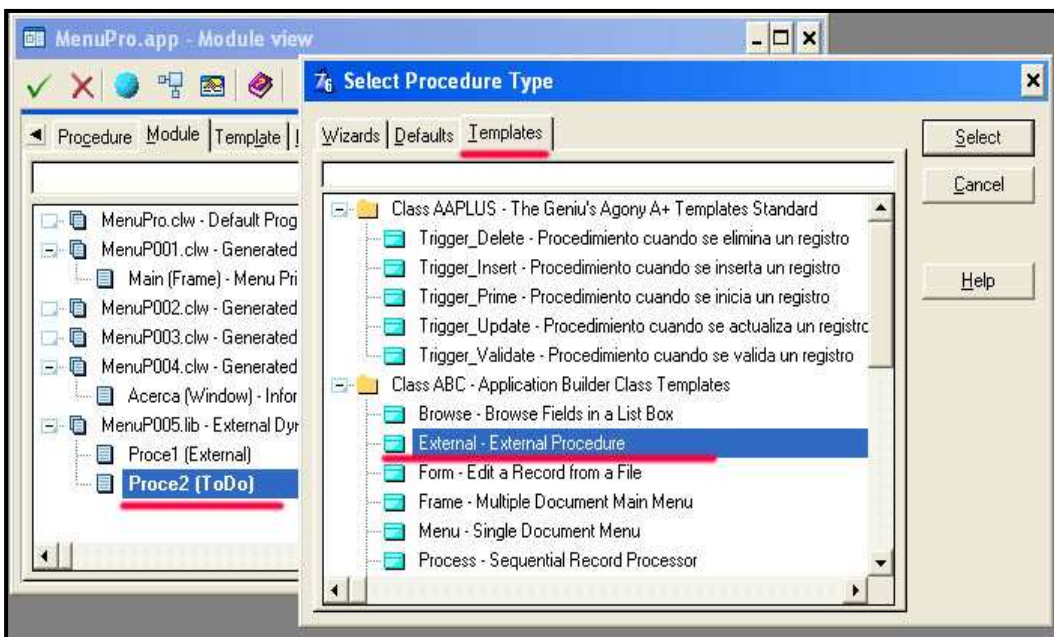
Configuración de la propiedad Global

Presionamos el botón *Global* y definimos exactamente las mismas opciones que en el *DLL* de procedimientos, tanto en el **Tab General**, como en **Control Files** y en la definición de variables (sí es que no las definimos en el diccionario) marque las casillas tal como se muestran en las dos imágenes.



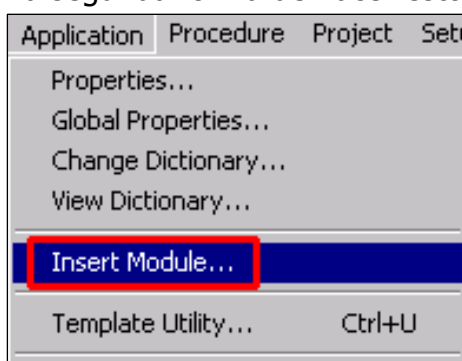
Creando nuestro procedimiento externo

Para crear los procedimientos externos hay dos formas de hacerlo, la primera es Posicionándonos encima del procedimiento en blanco (ToDo) y dándole doble clic con el botón izquierdo del Mouse, inmediatamente sale una ventana para que usted seleccione el tipo de procedimiento a elegir, como se observa en la siguiente imagen.



Se elige external Procedure, esto quiere decir que nuestro procedimiento será interno y automáticamente crea un código procedimiento así **MenuP005.lib**.

La segunda forma de hacer esto es yéndose por la opción application Insert Module

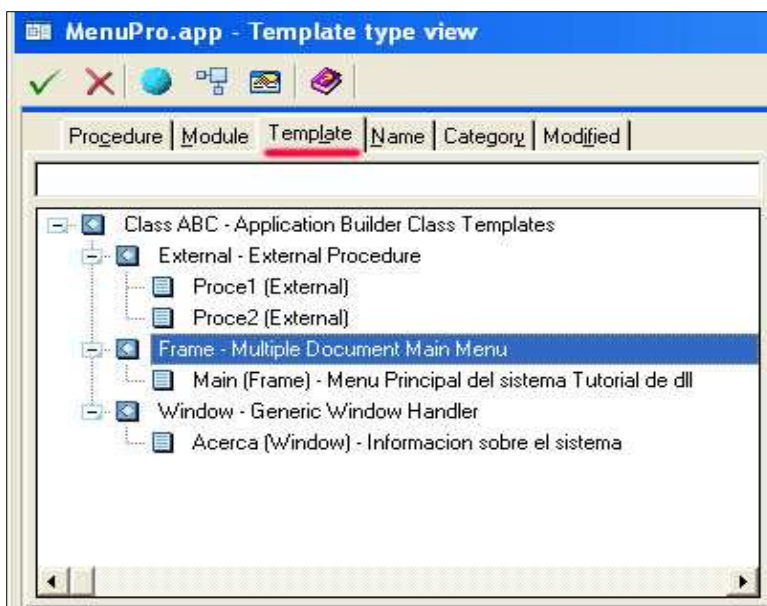


Insertar o ligar el procedimientos dll al proceso external.

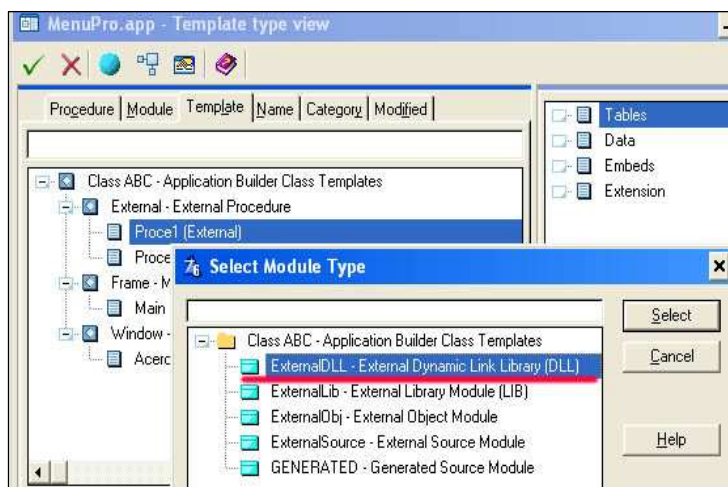
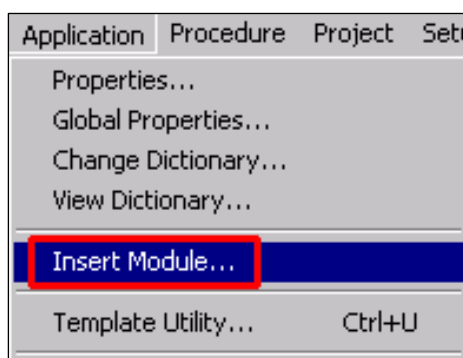
Ya hemos creado los procedimientos externos, pero aun no hemos terminado, no falta la parte de mayor confusión a la hora de agregar las dlls de procedimientos y la dll de datos a nuestros procedimientos externos, para lograr esto desde el árbol de clarión váyase a la opción template así como se observa en la imagen.

Insertar o ligar el procedimientos dll al proceso external.

Ya hemos creado los procedimientos externo, pero aun no hemos terminado no falta la parte de más confusión a la hora de agregar las dlls de procedimientos y la dll de datos a nuestros procedimientos external, para lograr esto desde el árbol de clarión váyase a la opción témpate así como se observa en la imagen.



Posesiónese encima del primer procedimiento externo, en este caso sería en Proce1(external), ahora váyase a la opción del menú de clarión Aplicación Insert Module inmediatamente sale la ventana selección el tipo de modulo.



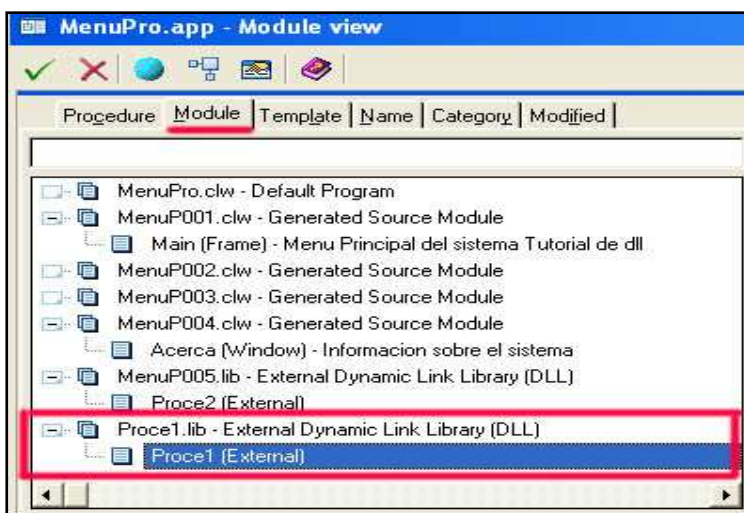
Seleccione la opción ***externalDLL(ExternalDynamic Link Library(DLL)***
Luego sale la siguiente ventana.



Presione el puntito en name para abrir el DLL de Procedimiento correspondiente en este caso, seleccione Proce1.dll. Su procedimiento debe quedar como se observa en la siguiente imagen.



Ya le hemos insertado las dll de procedimiento a nuestro proceso externo, si queremos ver el resultado, váyase al árbol de clarión seleccione module y se muestra como en la siguiente imagen.

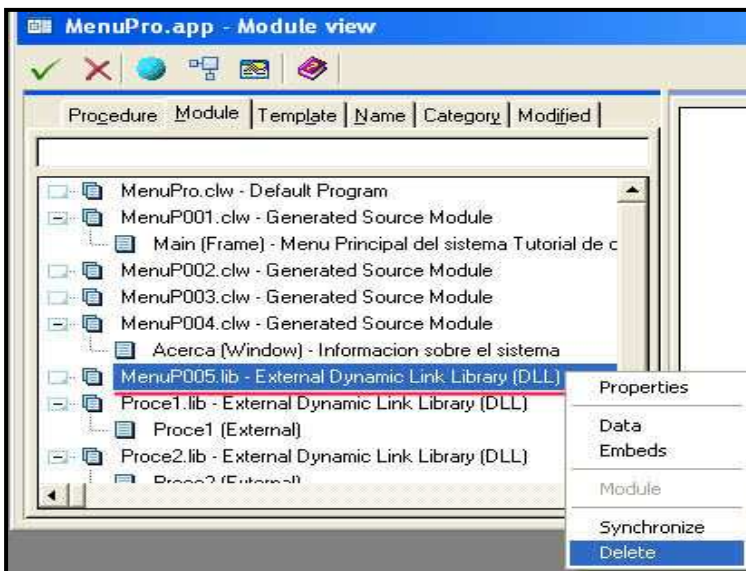


Ya he terminado con ligar o insertar el dll de procedimiento al nuestro proceso externo,

Ahora repita los mismos pasos para el proce2.dll, iguales como en el primero Proceso externo proce1.dll.

Nota muy importante

Si ya termino de ligar o insertar su dll de procedimiento, a su app exe o ejecutable, Hay un punto muy importante que hay que tomar en cuenta ante de compilar para evitar multitudes de errores, anteriormente dije que cuando uno crea un procedure de tipo external se crea de forma automática un código de procedure así: **MenuP005.lib** de tipo librería, mis dos procedure external tenían este código y los cambiamos por nuestro proce1.dll, proce2.dll que son los dll de procedimiento. Por que les digo esto Por que este procedure **MenuP005.lib** se ha quedado creado como huérfano, para ver Lo que le digo váyase al árbol de clarión modulo y observe por usted mismo lo que le acabo de decir, observe la imagen.



Como vemos en la imagen marcada con la línea roja el procedure huérfano, tenemos que eliminarlo de nuestra aplicación, para eso posicione encima del que desea borrar, presione el botón derecho del Mouse para que salga la pequeña lista de opciones que usted vez en la imagen, entonces seleccione delete para que borre dicho procedure, con esto ya logrado doy por terminado el punto de insertar o ligar dll de procedimiento a nuestro programa ejecutable o exe.

Le recomiendo que observe bien su aplicación, para que se asegure que no hay procedure huérfanos.

Insertar o ligar la dll de datos al MenuPro.exe

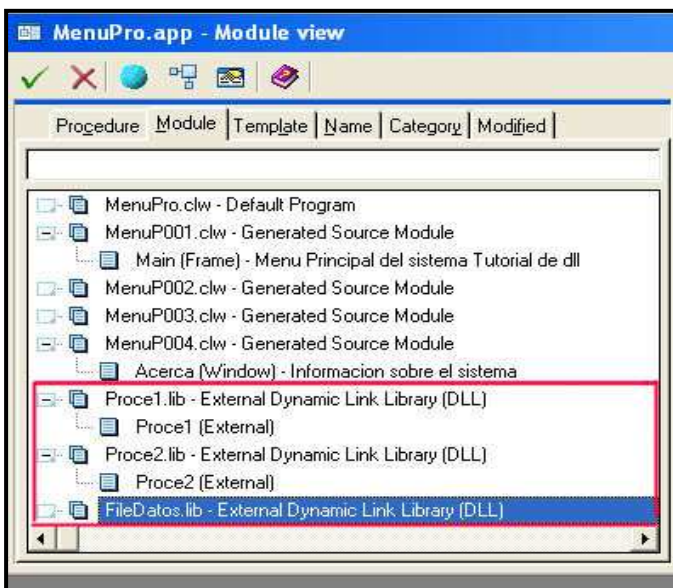
Exactamente igual que como hicimos al ligar nuestro *DLL de procedimientos* con el *DLL de datos*. Debemos ligar cada *DLL o EXE* con el *DLL de datos*. (Recordemos que en ese *DLL* se encuentran las definiciones de las variables globales de témples, las clases y de un modo u otro, nuestras variables globales). Para insertar o ligar la dll de datos a nuestro programa exe o ejecutable váyase al árbol de clarión module luego a la opción de clarión aplicación insertar module como se observa en la imagen.



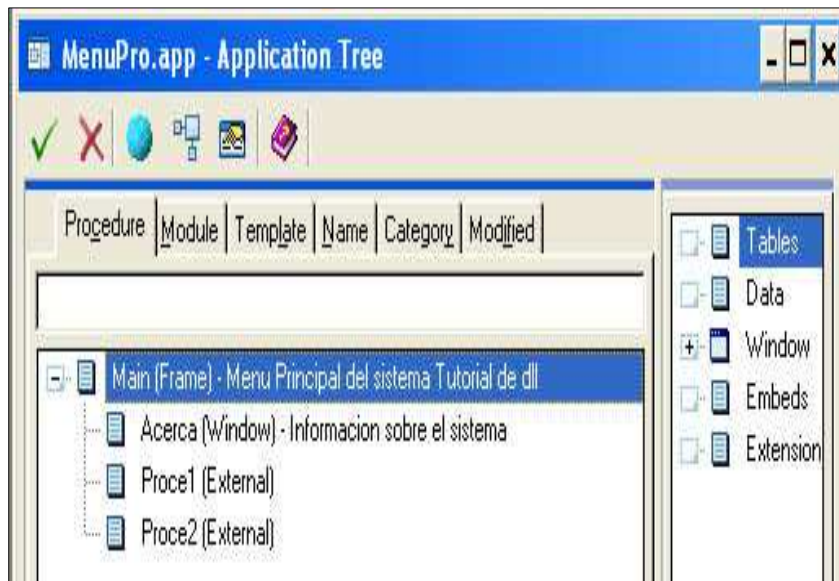
Se supone que ha seguido los pasos tal como le es ido diciendo, no es tiempo de dudas ni de errores, todos tiene que estar como está imagen.

Dándole un vitazo final a nuestra aplicación

Ya terminado este paso es bueno que le demos un vitazo final a nuestra aplicación para ver como ha quedado, para eso, váyase al árbol de clarión module y observe si está como la imagen siguiente.



Si su aplicación está como se vez aquí en esta imagen, quiero decirle que usted ha terminado ya su aplicación exe o ejecutable y que esta lista para ser compilada, pero ante le sugiero que se vaya a la opción producere, su aplicación debe verse como se vez en esta imagen.



Al ejecutar nuestra aplicación no notaremos ningún cambio comparado con la versión original (una sola aplicación), pero internamente ya es un EXE que llama para su ejecución a dos DLLs nuestros.

No se asuste compile su aplicación para que salga de las dudas.

Espero que todo le haya salido bien en su compilación, ami me salio así.

Suerte....

Nota Importante

Aquellos que lean este manual, si le interesan los ejemplos que se usaron en la elaboración del mismo, por favor solicitarlo vía email, que con mucho gusto se lo enviaré mi correo es: pachecoramon@hotmail.com

Conclusión

Finalmente mi objetivo principal en la creación de este manual o instructivo es que usted haya adquirido los conocimientos básicos para crear aplicaciones multi-dll, esperando que haya llenado su expectativa.

En mi poca experiencia trabajando con dll, puedo decir que lenguaje clarion tiene grande debilidades en esta parte, ya que cualquier template mal elaborado puede dañar las clases y las sub clases de clarion.

Recuerde que al trabajar con *DLLs* aparezcan de repente, al compilar uno de ellos, una Gran cantidad de errores (incluso cientos). Generalmente es debido a que se pasó por alto algunas de las opciones, o se activó otra indebidamente. Crear DLLs es básicamente algo rutinario (por lo menos las opciones que debemos activar por cada tipo de aplicación: *DLL de datos*, *DLL de procedimientos*, *EXEs*), por lo que normalmente basta repasar los pasos que se deben seguir para solucionar los errores.

Este manual solo alcanza los niveles básicos en la creación de aplicación multi-dll. En otro próximo trabajo hablaré de mi experiencia más avanzada.

Ing. R. Pacheco
Analista Programador
Pachecoramon@hotmail.com
Santo Domingo, Rep. Dominicana

Agradecimientos

Quiero agradecer aquellos protagonistas anónimos que de una u otra forma colaboraron con sus ideas en este proyecto, son ellos.

Nelson Ariel

Paraguay, gracia mi hermano, con usted se cumplió lo que dice la bérlica " El que da, recibe "

Héctor Daniel Madeira

Argentina, Gracia Héctor tu siempre me saca del hoyo cuando meto las patas

Heudy Félix

Rep. Dominicana, agracia a mi compueblano que siempre aporta su granito de arena, Y me saco de la oscuridad en este proyecto.

Heberto Barrio

México, Gracia Heberto porque siempre esta dispuesto dar un poco de luz en medio de la oscuridad.

Raulhm

Argentina, gracia mi hermano por excelente colaboración

Alejandro

Argentina, mi amigo Alejandro por pasarme los primero datos para del tema de la dll muchas gracias

Gustavo olmedo

De evolution, gracia Gustavo por tu gentileza y tu apreciada colaboración en foro de clarión

Que Dios continué bendición a cada unos de estos colegas.

Gracias muchachos..... reciban muchas bendiciones

Ing. R. Pacheco
Analista Programador