



WDO

Web Database Objects

Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1



WDO

Web Database Objects

Harbour for Web





WDO

Web Database Objects

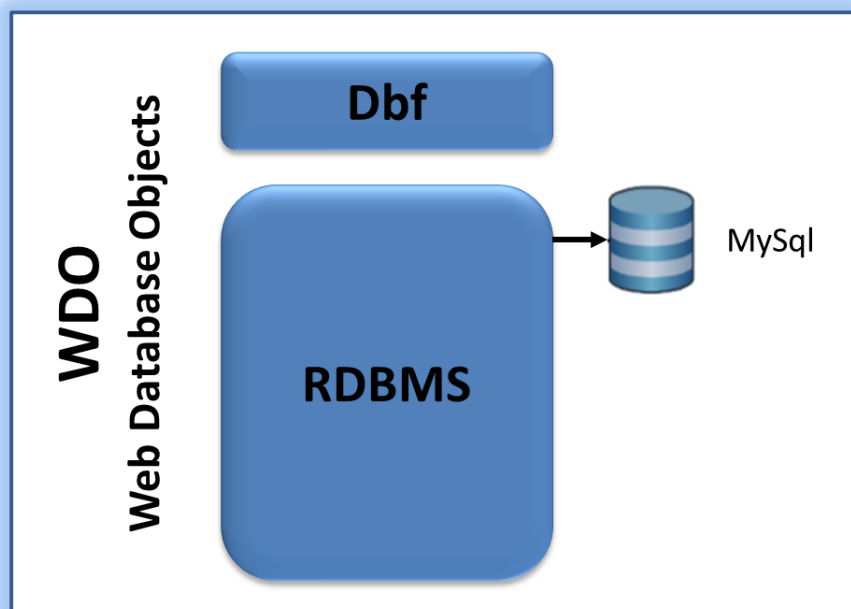
Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

Preámbulo	2
Carga de la librería	3
Hello WDO	4
DBF	5
RDBMS – MySql	8

Preámbulo

Web Database Objects (WDO) es una librería que te permitirá conectar con bases de datos desde mod_harbour de una manera fácil y potente. Básicamente podremos conectar a cualquier RDBMS y usar las tablas de datos de la misma manera independientemente de cual sea el motor de datos.

Habrà una forma particular de usar los DBF y otra de usar DB populares como MySql, Postgre, SQLite, Oracle... En esta primera versión ya tenemos disponible la conexión a Dbf y a MySql



A continuación se describe los pasos básicos para poder ejecutar y usar este plugin para Harbour for Web



Carga de la librería

Para poder usar la librería primero la habremos de descargar desde el repositorio <https://github.com/carles9000/wdo>. Imaginemos que tenemos el módulo wdo_lib.hrb en la carpeta \lib\wdo, lo primero que habremos de hacer desde nuestro programa es cargar el hrb y lo haremos con la función LoadHRB().

```
//      {% LoadHRB( '/lib/wdo/wdo_lib.hrb' ) %}
```

Observemos que tenemos al inicio de nuestro prg la barra de comentarios // y a continuación encerrado con los símbolos de macrosustitución {% ... %} hacemos la carga de la librería con LoadHrb().

A nivel de funcionamiento la línea está mentada y no se ejecutará como cualquier comentario en el programa, pero mod_harbour antes de ejecutar el código reemplaza y ejecuta lo que hay entre {%...%} como si se tratara de un pre-procesado. Esto nos da una gran potencia a nuestro sistema pudiendo ejecutar antes de nuestro prg diferentes funciones que nos ayudará a configurar fácilmente nuestro entorno.

Un caso también importante y que veremos más adelante es por ejemplo en el caso de MySQL, si queremos especificar donde tenemos instalada la dll, y lo podremos hacer de la siguiente manera:

```
//      {% HB_SetEnv( 'WDO_PATH_MYSQL', "c:/xampp/htdocs/" ) %}      //      Usuarios Xampp
```

Establecemos una variable de entorno que usaremos posteriormente desde nuestro prg.



WDO

Web Database Objects

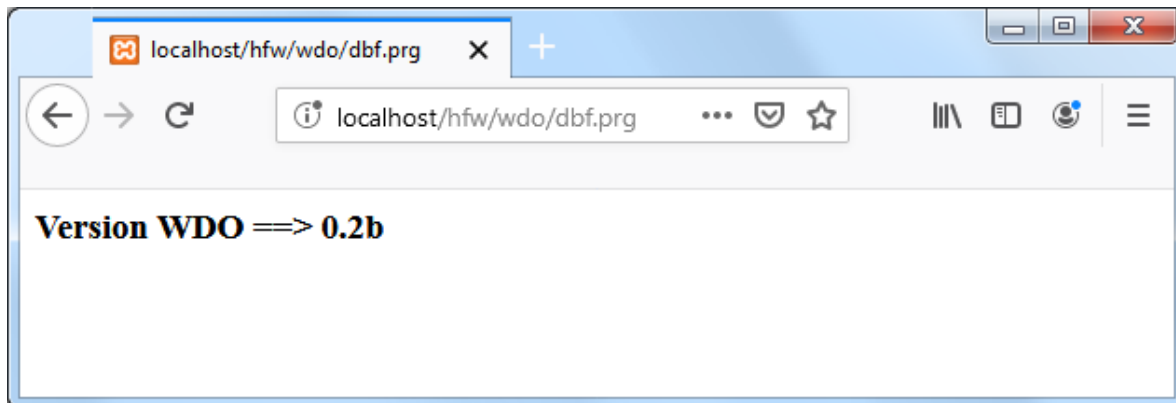
Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

Hello WDO

El primer ejemplo es comprobar si tenemos la librería cargada. Se tratará de cargar la librería y mostrar la versión de esta

```
// -----  
// Title.....: WDO Web Database Objects  
// Description: Test WDO  
// Date.....: 28/07/2019  
//  
// {% LoadHRB( '/lib/wdo/wdo_lib.hrb' ) %} // Loading WDO lib  
// -----  
  
FUNCTION Main()  
  
    ?? '<h3>Version WDO ==> ' , WDO():Version()  
  
RETU NIL
```

Si el sistema está listo y preparado nos habrá de aparecer por pantalla la versión de la librería





WDO

Web Database Objects

Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

DBF

Para cargar un objeto para el uso de Dbf's lo haremos de la siguiente manera

```
o := WDO():Dbf( <cTable>, <cIndex>, [<IOpen>] )
```

Por defecto el RDD usado es DBFCDX

Por defecto <IOpen> es .T. y lo que hace es abrir en la misma accion la tabla/indice indicado

El sistema abrirá el dbf y indice si se indica y devolverá una instancia de la clase a la variable.

```
oDb1 := WDO():Dbf( 'customer.dbf', 'customer.cdx' )
```

Podemos indicar al WDO una serie de datas que usara por defecto cada vez que abrimos una tabla:

```
::cDefaultPath  
::cDefaultRdd
```

La manera de usarlo es crear una instancia de WDO y asignar valores

```
//      Config Sistema...  
  
o := WDO():Dbf()  
    o:cDefaultPath := hb_getenv( 'PRGPATH' ) + '/data'  
    o:cDefaultRdd  := 'DBFCDX'
```

A partir de este momento cada vez que creemos un WDO ira a buscar en el path indicado y usará por defecto el Rdd

```
oDb1 := WDO():Dbf( 'customer.dbf', 'customer.cdx' )  
oDb2 := WDO():Dbf( 'states.dbf', 'states.cdx' )
```



WDO

Web Database Objects

Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

En el caso que hubiera alguna tabla fuera de los valores por defecto podemos indicar en el 3 parametro .F. (no abrir la tabla) y cambiar los valores. Posteriormente habremos de ejecutar el método ::Open()

```
oDb1 := WDO():Dbf( 'customer.dbf', 'customer.cdx' )
oDb2 := WDO():Dbf( 'states.dbf', 'states.cdx' )

oDb3 := WDO():Dbf( 'vendors.dbf', 'vendors.cdx', .F. )
oDb3:cPath      := hb_getenv( 'PRGPATH' ) + '/data/sales'
oDb3:cRdd       := 'DBFNTX'
oDb3:Open()

oDb4 := WDO():Dbf( 'users.dbf', 'users.cdx' )
```

Una vez abierta la tabla podremos ejecutar todos los métodos definidos en la clase y manejar la DBF como si fuera un típico objeto Tdatabase

Ver code fuente RDBMS_DBF para consultar todos los métodos disponibles.

Un ejemplo básico de uso podria ser el siguiente

```
// -----
// Title.....: WDO Web Database Objects
// Description: Test WDO
// Date.....: 28/07/2019
//
// {% LoadHRB( '/lib/wdo/wdo_lib.hrb' ) %}           // Loading WDO lib
// -----

FUNCTION Main()

    LOCAL o
    LOCAL oDb1, oDb2, oDb3, oDb4

    ?? 'Init ' + time() + '<hr>'

    // Config Sistema...

    o := WDO():Dbf()
    o:cDefaultPath      := hb_getenv( 'PRGPATH' ) + '/data'
    o:cDefaultRdd       := 'DBFCDX'

    // Open Tables

    oDb1 := WDO():Dbf( 'customer.dbf', 'customer.cdx' )

    oDb2 := WDO():Dbf( 'states.dbf', 'states.cdx' )

    oDb3 := WDO():Dbf( 'vendors.dbf', 'vendors.cdx', .F. )
    oDb3:cPath      := hb_getenv( 'PRGPATH' ) + '/data/sales'
```



WDO

Web Database Objects

Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

```
oDb3:Open()

oDb4 := WDO():Dbf( 'users.dbf', 'users.ntx', .F. ) // 3 param. Open
oDb4:cRdd := 'DBFNTX'
oDb4:Open()

? '<b>==> Open tables...</b>'

// Seek States...

? '<br><b>==> Seek States...</b>'

oDb2:Focus( 'CODE' )

IF oDb2:Seek( 'MA' )

    ? oDb2:FieldGet( 'code' ), oDb2:FieldGet( 'name' )

ENDIF

// List Vendors...

? '<br><b>==> List Vendors...</b>'

oDb3:First()

WHILE !oDb3:Eof()

    ? oDb3:FieldGet( 'id' ), oDb3:FieldGet( 'name' ), oDb3:FieldGet( 'phone' )

    oDb3:next()

END

// Info Customer...

? '<br><b>==> Info Customer...</b>'

? oDb1:cDbf, oDb1:cRdd

? 'lOpen', oDb1:lOpen
? 'First()', oDb1:first()
? 'Fieldget(1)', oDb1:FieldGet( 1 )
? 'Fieldget( "first" )', oDb1:FieldGet( 'first' )
? 'Count()', oDb1:Count()
? 'Fieldname(1)', oDb1:FieldName( 1 )
? 'Next(5)', oDb1:next( 5 ), oDb1:Recno(), oDb1:FieldGet( 1 )
? 'Goto(7)', oDb1:goto(7), oDb1:Recno(), oDb1:FieldGet( 1 )
? 'Prev(2)', oDb1:prev( 2 ), oDb1:Recno(), oDb1:FieldGet( 1 )
? 'Last()', oDb1:last(), oDb1:FieldGet( 1 )
? 'First()', oDb1:first(), oDb1:Recno()

IF oDb1:Rlock()
    oDb1:FieldPut( 'street', time() )
ENDIF

? 'Street: ', oDb1:FieldGet( 'street' ), oDb1:Recno()
? '<hr>'
```

RETU NIL



WDO

Web Database Objects

Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

RDBMS – MySql

WDO():Rdbms() te ayudará a conectar de manera facil con los distintos gestores de bases de datos. MySql ya está operativo y a continuación se describe su uso

MySql necesita de una dll que se proviene con la librería mod_harbour: libmysql.dll y libmysql64.dll para versiones windows

Si tenemos instalada esta dll en algun directorio especial, por ejemplo usuarios de xampp en windows, podremos indicar al principio del programa su ubicación:

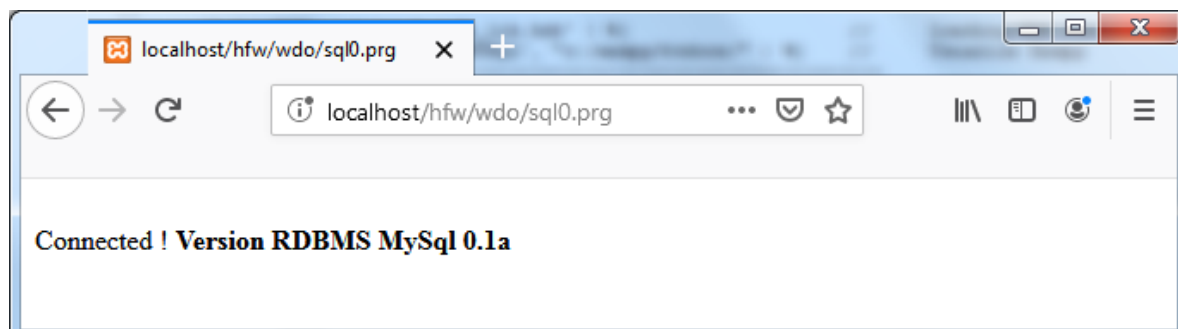
```
//      {% HB_SetEnv( 'WDO_PATH_MYSQL', "c:/xampp/htdocs/" ) %}      //
```

Usuarios Xampp

El ejemplo básico para comprobar si se ha cargado la librería y si ha conectado con mysql seria el siguiente:

```
//      -----  
//      Title.....: WDO Web Database Objects  
//      Description: Test WDO  
//      Date.....: 28/07/2019  
//  
//      {% LoadHRB( '/lib/wdo/wdo_lib.hrb' ) %}      //      Loading WDO lib  
//      {% HB_SetEnv( 'WDO_PATH_MYSQL', "c:/xampp/htdocs/" ) %}      //      Usuarios Xampp  
//      -----  
  
FUNCTION Main()  
  
    LOCAL o  
  
        o := WDO():Rdbms( 'MYSQL', "localhost", "harbour", "password", "dbHarbour", 3306 )  
  
        IF o:lConnect  
  
            ? 'Connected !', '<b>Version RDBMS MySql', o:Version()  
  
        ELSE  
  
            ? o:cError  
  
        ENDIF  
  
    RETU NIL
```

Si el sistema está bien parametrizado habria de aparecer una pantalla similar a esta:





WDO

Web Database Objects

Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

Una vez conectados la clase ofrece los siguientes métodos

METHOD Query(cSql)
METHOD Count(hRes)
METHOD FCount(hRes)
METHOD LoadStruct()
METHOD DbStruct()
METHOD Fetch(hRes)
METHOD Fetch_Assoc(hRes)
METHOD FetchAll(hRes, lAssociative)
METHOD Free_Result(hRes)



WDO

Web Database Objects

Autor Carles Aubia
Fecha 12/08/2019
Versión 0.1

Un ejemplo básico de uso podría ser el siguiente:

```
// -----  
// Title.....: WDO Web Database Objects  
// Description: Test WDO  
// Date.....: 28/07/2019  
//  
// {% LoadHRB( '/lib/wdo/wdo_lib.hrb' ) %} // Loading WDO lib  
// {% HB_SetEnv( 'WDO_PATH_MYSQL', "c:/xampp/htdocs/" ) %} // Usuarios Xampp  
// -----  
  
FUNCTION Main()  
  
    LOCAL o, oRs, a  
  
    o := WDO():Rdbms( 'MYSQL', "localhost", "harbour", "password", "dbHarbour", 3306 )  
  
    IF o:lConnect  
  
        ?? 'Version WDO', o:ClassName(), o:Version()  
  
        IF !empty( hRes := o:Query( 'select * from users' ) )  
  
            ? 'Count(): ', o:Count( hRes )  
            ? 'Fields: ', o:FCount( hRes )  
  
            ? '<br><b>Fields</b>'  
            for n := 1 to len( o:aFields )  
                ? o:aFields[n][1], o:aFields[n][2]  
            next  
  
            ? '<br><b>Data</b>'  
            //while ( !empty( a := o:Fetch( hRes ) ) )  
            //    ? valtochar( a )  
            //end  
  
            // Associative array  
            while ( !empty( a := o:Fetch_Assoc( hRes ) ) )  
                ? valtochar( a )  
            end  
  
        ELSE  
  
            ? 'Error: ', o:cError  
  
        ENDIF  
  
    ELSE  
  
        ? 'Error: ', o:cError  
  
    ENDIF  
  
    RETU NIL
```