

A 3D icon of a database cylinder with four horizontal blue bands.

WDO

Web Database Objects

Harbour for Web

mod Harbour.V2





WDO

Web Database Objects

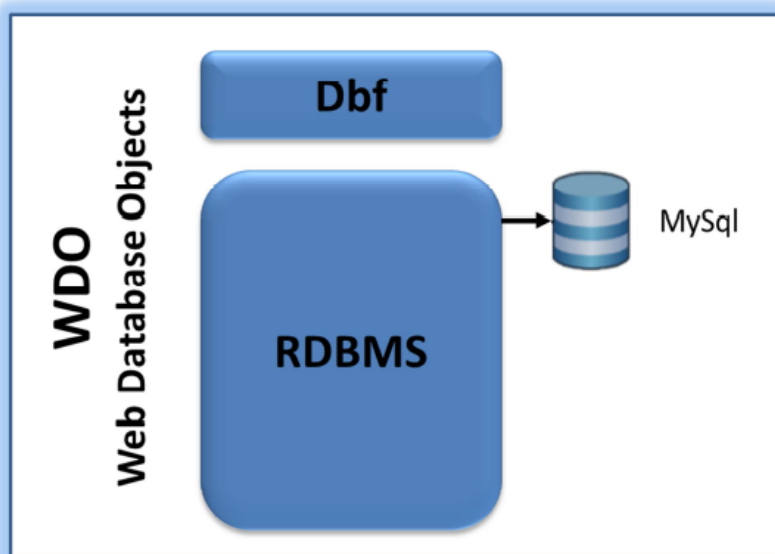
Autor Carles Aubia
Fecha 11/05/2022
Versión Rev. 2.1

Preámbulo	2
Carga de la librería	3
DBF	5
RDBMS – MySql	8
Conclusiones	11

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 modHarbour.V2



Carga de la librería

Para poder usar la librería primero la habremos de descargar desde el repositorio <https://github.com/carles9000/wdo.v2> . Imaginemos que tenemos el módulo wdo.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 mh_LoadHRB().

```
// {% mh_LoadHRB( '/lib/wdo/wdo.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 mh_LoadHrb().

A nivel de funcionamiento la línea esta 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 preprocesado. 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 mas 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

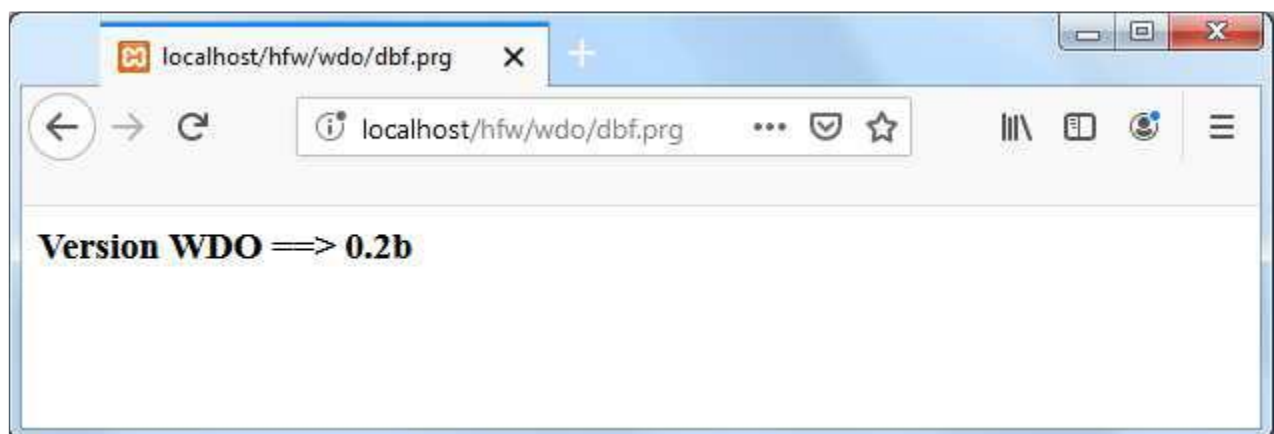
Autor Carles Aubia
Fecha 11/05/2022
Versión Rev. 2.1

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  
//  
// {% mh_LoadHRB( '/lib/wdo/wdo.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





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 creamos 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 11/05/2022
Versión Rev. 2.1

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 podría ser el siguiente

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

// Test Speed...

FUNCTION Main()

    LOCAL o
    LOCAL oDb1

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

    // Config Sistema...

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

    // Open Tables

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

        oDb1:Focus( 'state' )
        oDb1:Seek( 'LA' ) // KS, LA, MA, ...
```



WDO

Web Database Objects

Autor Carles Aubia
Fecha 11/05/2022
Versión Rev. 2.1

```
Test( oDb1, 'state', 'LA' )

RETU NIL

FUNCTION Test( o, cField, uValue )

    LOCAL cHtml := ''

    cHtml += '<b>Focus </b>' + cField + ' = ' + uValue + '<br>'

    cHtml += '<table border="1">'

    WHILE !o:Eof() .AND. o:FieldGet( cField ) = uValue

        cHtml += '<tr>'
        cHtml += '<td>' + ltrim(str(o:Recno())) + '</td>'
        cHtml += '<td>' + o:FieldGet( 'first' ) + '</td>'
        cHtml += '<td>' + o:FieldGet( 'last' ) + '</td>'
        cHtml += '<td>' + o:FieldGet( 'city' ) + '</td>'
        cHtml += '<td>' + o:FieldGet( 'zip' ) + '</td>'
        cHtml += '<td>' + o:FieldGet( 'state' ) + '</td>'
        cHtml += '</tr>'

        o:next()

    END

    cHtml += '</table>'

    ? cHtml

RETU
```



RDBMS – MySql

WDO():Rdbms() te ayudará a conectar de manera fácil 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 algún 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  
//  
// {% mh_LoadHRB( '/lib/wdo/wdo.hrb' ) %} // Loading WDO lib  
// {% HB_SetEnv( 'WDO_PATH_MYSQL', "c:/xampp/htdocs/" ) %} // Usuarios Xampp  
// -----  
  
FUNCTION Main()  
  
    LOCAL o  
  
    ? "<b>==> Test Error de conexion...</b><br>"  
  
    o := WDO():Rdbms( 'MYSQL', "localhost", "harbour", "hb1234", "db_zzz", 3306 )  
  
    IF o:lConnect  
  
        ? 'Connected !', '<b>Versión RDBMS MySql', o:Version()  
  
    ELSE  
  
        ? o:cError  
  
    ENDIF  
  
    RETU NIL
```




WDO Web Database Objects

Autor Carles Aubia
Fecha 11/05/2022
Versión Rev. 2.1

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



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)

Esta clase, pretende ser muy simplista y a la vez efectiva. Es muy fácil poner capas de funcionalidades.

Otro ejemplo básico podría ser el siguiente:

```
// -----  
// Title.....: WDO Web Database Objects  
// Description: Test WDO  
// Date.....: 28/07/2019  
//  
// {% mh_LoadHRB( '/lib/wdo/wdo.hrb' ) %} // Loading WDO lib  
// {% HB_SetEnv( 'WDO_PATH_MYSQL', "c:/xampp/htdocs/" ) %} // Usuarios Xampp  
// -----  
  
FUNCTION Main()  
  
    LOCAL o, oRs, hRes  
  
    o := WDO():Rdbms( 'MYSQL', "localhost", "harbour", "hb1234", "dbHarbour", 3306 )  
  
    IF ! o:lConnect  
        RETU NIL  
    ENDIF
```



WDO

Web Database Objects

Autor Carles Aubia
Fecha 11/05/2022
Versión Rev. 2.1

```
?? '<h3>Data Base Customer</h3><hr>'

cSql  := "SELECT count(*) as total FROM customer"

hRes  := o:Query( cSql )
oRs   := o:Fetch_Assoc( hRes )

? '<b>Registros Totales: </b> ', oRs[ 'total' ]

cSql := "SELECT * FROM customer c " +;
        "LEFT JOIN states s ON s.code = c.state " +;
        "WHERE ( c.state = 'LA' OR c.state = 'AK' ) and c.age >= 58 and
c.age <= 60 and c.married = 1 " +;
        "ORDER by first      "

? '<br><b>Sql: </b>' , cSql

IF !empty( hRes := o:Query( cSql ) )

    ? '<br><b>Total Select: </b> ', o:Count( hRes )

    aData := o:FetchAll( hRes )

    o:View( o:DbStruct(),      aData )

    o:Free_Result( hRes )

ENDIF

RETU NIL
```

localhost/master/wdo.v2/sql5.p

localhost/master/wdo.v2/sql5.prg

Data Base Customer

Registros Totales: 100176

Sql: SELECT * FROM customer c LEFT JOIN states s ON s.code = c.state WHERE (c.state = 'LA' OR c.state = 'AK') and c.age >= 58 and c.age <= 60 and c.married = 1 ORDER by first

Total Select: 76

View table

id	first	last	street	city	state	married	age	state	code
35023	Aaron	Hensiek	Mary Baldwin Drive	Arlington	AK	1	60	Alaska	AK
88074	Achim	Knox	Parkside Avenue	Lake Worth	AK	1	58	Alaska	AK
83460	Adele	White	North Maple	Pearl Harbor	LA	1	58	Louisiana	LA
88239	Ahab	Eddy	Prince Charles Drive	Clovis	LA	1	58	Louisiana	LA





Conclusiones

WDO es una sencilla librería diseñada para ayudar a los programadores a conectar fácilmente con dbfs y DB, aun que esta primera versión que era para consolidar prototipo, solo lo hace a MySQL.

La idea era ofrecer una funcionalidad básica y potente, y que se sumaran diferentes usuarios para poderla ir escalando a medida que se usara.

De alguna manera los objetivos eran comprobar la carga de la librería a modo de plugin usando hrbs y ofrecer fácilmente esta funcionalidad.