



Grupo Josenet

Manual hbnetio y mod_harbour y harbour

Índice:

Índice	1
Operatividad	1
Diagramas Técnico	3

Hbnetio -> primeros pasos:

El programa (libreria escrita por master Przemyslaw Czerpak (druzus/at/poczta.onet.pl))

Hbnetio nos permite conectarnos vía internet sin necesidad de una interfaz con solo aperturar un puerto donde se haya instalado y ejecutado el programa hbnetio.exe el mismo que se lo arranca de una forma como esta

```
c:\hb32\bin\hbnetio -port=2941 -pass=topsecret -iface=127.0.0.1 -rpc  
-rootdir=c:\hb32\bin\data
```

donde

-port=2941 es el puerto por donde se reciben las peticiones

-pass=topsecret es la clave que se usa en las comunicaciones que tienen lugar entre el cliente (un programa Harbour o un programa Mod-Harbour) y el servidor (hbnetio.exe)



Grupo Josenet Manual hbnetio y mod_harbour y harbour

-iface=127.0.0.1 es la dirección donde se encuentra HbNetio.exe puede ser una ip local, una ip publica en internet un localhost o un dominio en internet

Cuando cargamos hbnetio queda funcionando y empieza a recibir las peticiones y ejecuta de una manera multi hilada, cualquier función que queramos ejecutar en el server debe ser linkeada en el ejecutable de hbnetio cuyo código fuente se halla en c:\hb32\contrib\hbnetio\utils\hbnetio\ expongo lo que yo tengo modificado en el para que se ejecuten en el server los

Mis modificaciones están en Azul como se mira

en hbnetio.prg

he aumentado esto

```
/*
 * Harbour NETIO server daemon
 *
 * Copyright 2010-2012 Viktor Szakats (vszakats.net/harbour)
 * Copyright 2009 Przemyslaw Czerpak <druzus / at / priv.onet.pl>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
 * (or visit their website at https://www.gnu.org/licenses/).
 *
```



Grupo Josenet

Manual hbnetio y mod_harbour y harbour

```
*/

/* TODO: - on the fly change of RPC filter modules
          - listing open files
          - listing active locks
          - gracefully shutting down server by waiting for connections to
close and not accept new ones
          - pausing server
          - sort out console UI from server side output
          - add support for subnet masks in allow/block lists, f.e.
172.16.0.0/12, and same for IPv6 */
//ini modificaciones de wilson gamboa a
memvar GLB_Work
memvar empresa
#define __CON_ADS__
#ifdef __CON_ADS__
    #include 'ads.ch'
#endif
//fin modificaciones de wilson gamboa a
#include "fileio.ch"
#include "inkey.ch"
#include "hbhrb.ch"
#include "hbsocket.ch"
#include "hbver.ch"
#include "hbnetio.ch"
#define _NETIOSRV_IPV4_DEF "0.0.0.0"
#define _NETIOSRV_PORT_DEF 2941
#define _NETIOMGM_IPV4_DEF "127.0.0.1"
#define _NETIOMGM_PORT_DEF 2940
#define _RPC_FILTER "HBNETIOSRV_RPCMAIN"
/* enable this if you need all core functions in RPC support */
#ifdef HB_EXTERN
REQUEST __HB_EXTERN__
#endif

#define _NETIOSRV_cName 1
#define _NETIOSRV_nPort 2
```



Grupo Josenet

Manual hbnetio y mod_harbour y harbour

```
#define _NETIOSRV_cIFAddr      3
#define _NETIOSRV_cRootDir    4
#define _NETIOSRV_lRPC        5
#define _NETIOSRV_cRPCFFileName 6
#define _NETIOSRV_hRPCFHRB    7
#define _NETIOSRV_lEncryption 8
#define _NETIOSRV_lAcceptConn 9
#define _NETIOSRV_lShowConn  10
#define _NETIOSRV_lQuit      11
#define _NETIOSRV_pListenSocket 12
#define _NETIOSRV_hConnection 13
#define _NETIOSRV_mtxConnection 14
#define _NETIOSRV_hAllow      15
#define _NETIOSRV_hBlock      16
#define _NETIOSRV_mtxFilters   17
#define _NETIOSRV_hNotifStream 18
#define _NETIOSRV_mtxNotifStream 19
#define _NETIOSRV_MAX_        19
#define _NETIOSRV_CONN_pConnection 1
#define _NETIOSRV_CONN_nThreadID 2
#define _NETIOSRV_CONN_tStart    3
#define _NETIOSRV_CONN_hInfo     4
#define _NETIOSRV_CONN_MAX_     4
PROCEDURE Main( ... )
    jnet_ambiente()
    netiosrv_Main( .T., ... )
    RETURN
```

la funcion `jnet_ambiente()` es esta

```
function jnet_ambiente()
    REQUEST ADS
    #ifdef __CON_ADS__
        RDDREGISTER("ADS",1)
        RDDSETDEFAULT('ADS')
        //SET SERVER LOCAL
```



Grupo Josenet Manual hbnetio y mod_harbour y harbour

```
SET SERVER REMOTE
SET FILETYPE TO NTX
ADSLocking( .t. )
//SET WRAP ON
//SET DATE FRENCH
//SET SCOREBOARD OFF
SET DELETE ON
SET EXCLUSIVE OFF
//SET RIGHTS CHECKING OFF
//SET DECIMALS TO 2
//SET EPOCH TO 1950 //year( date() ) //1960
#endif
return nil
```

y tenemos un prg adicional donde hemos puesto todas nuestras funciones que trabajaran en el lado del servidor en el ejemplo se llama jn_netio_funciones.prg en mi caso del ejemplo uso Advantage Database Server por eso los cambios

con lo que el hbnetio.hbp me queda asi

```
{HB_MT='no'}-stop
hbnetio.hbc
rddads.hbc
-mt
-w3 -es2 -l
-o${hb_name}
hbnetio.prg
_console.prg
netiomgm.hb
\\wilson\texvalle\tv_venta\jn_netio_funciones.prg
_winsvc.prg{win}
-main=WinMain{win}
hbwin.hbc{win}
#-prgflag=-DHB_EXTERN
```



Grupo Josenet

Manual hbnetio y mod_harbour y harbour

Mis modificaciones están en Azul como se mira

Una vez resuelto lo del server tenemos la opción de conectarnos desde un cliente escrito en Harbour puede ser con Harbour Puro, Harbour + FiveWin , Harbour + Xailer, Mod_harbour etc

vamos a poner un simple ejemplo desde mod_harbour

```
//estos define son para poder trabajar cómodamente con las funciones en
el servidor
#xtranslate net:<!func!>( [<params,...>] ) => ;
    netio_FuncExec( #<func> [,<params>] )
#xtranslate net:[<server>]:<!func!>( [<params,...>] ) => ;
    netio_FuncExec( [ #<server> + ] ":" + #<func>
[,<params>] )
#xtranslate net:[<server>]:<port>:<!func!>( [<params,...>] ) => ;
    netio_FuncExec( [ #<server> + ] ":" + #<port> + ":" +
#<func> ;
                                [,<params>] )

#xtranslate net:exists:<!func!> => ;
    netio_ProcExists( #<func> )
#xtranslate net:exists:[<server>]:<!func!> => ;
    netio_ProcExists( [ #<server> + ] ":" + #<func> )
#xtranslate net:exists:[<server>]:<port>:<!func!> => ;
    netio_ProcExists( [ #<server> + ] ":" + #<port> + ":" +
#<func> )

function main()
local hCfg := Config_Netio()
local lConnect
local hRet
lConnect := netio_Connect( hCfg[ 'ip' ], hCfg[ 'port'],hCfg[
'ntimeout'], hCfg[ 'psw' ] )
    if lConnet
        ?'conectado'
```



Grupo Josenet Manual hbnetio y mod_harbour y harbour

```
//ejecuto la funcion que esta en el server
//y en este caso me regresa un hash con los datos que necesito
hRet = net:seekcliente(1234)
      ? hret['codcli'], hRet['nomcli']
      netio_Disconnect( hCfg['ip'], hCfg['port'])
    else
      ? 'no conectado'
    end
return nil

function Config_Netio()
return { 'ip'      => '192.168.0.2' ,;
        'port'    => 2491      ,;
        'psw'     => 'topsecret' ,;
        'ntimeout' => 3000    }
end
```

NOTA en el server esta incluida la función

```
function seekcliente( nCodCli )
  local cPath := 'z:\sistemas\' //camino en el server de los datos
  local hData := { 'codcli' => 0, 'nomcli' => 'no hellado' }
  use ( cPath + 'clientes' ) index ( cPath + 'clientes' ) alias
  clientes shared
  if dbseek( nCodCli )
    hData['codcli'] = clientes->codcli
    hData['nomcli'] = clientes->nomcli
  end
  dbclosetable()
return hData
end
```

Por demás está decir que si usamos Mod_harbour éste debe tener incluida la librería hbnetio en este caso particular el mod harbour NO NECESITA tener el soporte de ADVANTAGE DATABASE SERVER porque eso está en el lado del servidor, por ello se compila el HBNETIO en el lado del server ese sí con



Grupo Josenet Manual hbnetio y mod_harbour y harbour

soporte de ADVANTAGE, lo mismo seria si accedemos con Mysql o Sqlserver si nos damos cuenta eso en el lado del programa de mod_harbour no habría que cambiar absolutamente nada ya que el server me regresa hash con la info que pido de la misma forma puedo enviarle la info a que me la actualice con lo que hemos independizado el cliente (mod_harbour) de la parte servidora y será esta la que resuelva lo de accesos a los datos

Se debe indicar que estas ideas fueron creadas conjuntamente con el Master Carles Aubia muy conocido en nuestro mundo de la programación pues ha dado vida a muchas criaturas admirables como Génesis, Mercury, tWeb que han puesto al alcance de los profanos como nosotros el mundo de la Web

Hemos creado un proyecto llamado AppTruck mismo que ya funciona y está en fase beta para un manejo de entrega mercaderia por los camiones de una empresa bastante grande en Ecuador

La forma de trabajo acá explicada ha hecho que nos dividamos el trabajo así

Charly

Escribe toda la parte Web con html5 + mod_harbour + mercury + tweb + javascript + hbnetio

Wilson (yo)

he escrito toda la parte servidora que se encuentra embutida en hbnetio.exe y que accede a un conjunto de datos en un server que se halla en producción en tablas Dbf + Advantage database server me he encargado de escribir simples funciones que abren las tablas y regresan consultas o realizan modificaciones de acuerdo a lo que Charly necesita en mi humilde opinión con esto nos hemos ahorrado más de la mitad del tiempo de desarrollo ya que Charly se dedica a lo que más sabe y yo a lo mio

Bueno es mi experiencia con esto

saludos

Att

Wilson Gamboa A



Grupo Josenet Manual hbnetio y mod_harbour y harbour

Créditos

Harbour fue creado por master Antonio de Linares y puesto a dominio publico hace mas de 20 años por hoy es un super lenguaje 4G lleno de características modernas al alcance de todos nosotros

Mod_Harbour fue creado por master Antonio de Linares y puesto a dominio público

Mercury, tWeb son propiedad de Carles Aubia y puestos a dominio público