

COBHTTPD

Servidor de Aplicaciones para COBOL.

Manual del Usuario

(Ver. 2.40)

INDICE

Descripción.....	3
Como trabaja este programa.....	4
COBHHTTPD Archivos y Programas	5
COBHHTTPD Directorios.....	6
Directorio de Instalación	6
Directorio de Documentos.....	6
Directorio de Reportes.....	7
Directorio de Temporales	7
Directorio XML.....	7
Configuración	9
General.....	9
Autorun Config.....	11
Configuración (Proyectos).....	13
Compiladores COBOL soportados.....	16
ACUCOBOL, interface httpacu.cbx.....	16
RMCOBOL, interface httpm5.cob	16
MSCOBOL 5, interface httpm5.exe	16
POWER COBOL 3, interface httpwc3.exe	17
POWER COBOL 85, interface httpw85.exe	17
POWER COBOL 5, interface httpw5.exe	17
Botón Edit XML.....	19
Formato del archivo de interfase XML	20
Editor XML	22
Ejecutando el programa Server COBHHTTPD	25
Ejecutando los programas COBOL	26
Generando Contenido HTML:	26
Generando Reportes PDF:	26
Acceso directo:	26
Llamando desde archivos HTML.....	26
Diferencia entre contenido HTML y PDF.....	27
Recibiendo datos de entrada en nuestro programa COBOL.....	28
HTTP-CONTROL.....	28
HTTP-DATA-INP	29
Como generar la salida a entregar al cliente.....	30
Manejo de sesiones	32
Incluir archivos externos	33
Introducción a la librería CGILIB	34

Descripción.

COBHTTPD es un servidor de Aplicaciones para COBOL. Con este programa puedes correr en forma remota algún programa COBOL.

Contiene también un Servidor de Reportes, para visualizar reportes por medio de Internet.

Este Servidor de reportes trabaja con una versión especial de COBVIEW para convertir los reportes a PDF y enviárselos al servidor COBHTTPD.

COBHTTPD trabaja también como servidor web, porque usa el protocolo http para recibir y enviar los datos del cliente y hacia el cliente navegador.

COBHTTPD no es solo un programa puente para ejecutar programas COBOL, COBHTTPD es un programa completo, no necesitas de algún programa externo para que funcione de manera total, y no necesita de algún DLL o librería externa para trabajar.

COBHTTPD no necesita una instalación especial para trabajar, por que trabaja en Windows 98/2000/XP y 2003, y la instalación es manual, por que tu copias el programa donde tu deseas y el directorio es definido por ti, De esta manera, tu siempre conocerás donde se colocan los archivos del programa (Muy importante para los programadores).

COBHTTPD puede regresar tanto contenido HTML como reportes en formato PDF.

Como trabaja este programa.

Primero un pequeño ejemplo:

- 1) El usuario visita la pagina para ver un reporte, y recibe una forma html para llenarla con los datos necesarios para crear el reporte.
- 2) El usuario llena la forma, y la postea hacia el servidor.
- 3) El servidor corre el programa COBOL apropiado, y convierte el archivo a PDF.
- 4) El servidor regresa el archivo PDF a el usuario y este es visualizado en el I explorer o Netscape.

COBHTTPD siempre regresa los reportes en formato PDF por las siguientes razones:

- PDF es universal
- Existe un plugin para tu browser preferido.
- No necesitas ningún programa especial para ver tus reportes.
- Puede ser impreso en cualquier impresora.
- Puede ser visualizado en diferentes plataformas: Windows/Linux/Unix

COBHTTPD te ayuda a convertir tus programas COBOL para trabajar en Internet, porque todos los parámetros que tu programa necesita son recibidos por la LINKAGE-SECTION.

Por que el protocolo http trabaja con datos, pero estos datos no son formateados, no contiene el tipo de información y no son delimitados por un tamaño, por esto COBHTTPD necesita conocer los programas que serán publicados en Internet, y necesita conocer el tamaño de los datos que recibirá de entrada para trabajar.

Así que, antes de que un programa pueda ser usado, es necesario definir en COBHTTPD, y es necesario que sean definidos los datos de entrada para este programa, todo puede ser echo con el configurador de COBHTTPD.

COBHTTPD Archivos y Programas

Los programas que son parte de la suite cobhttpd son:

CFGCOBHTTPD.EXE - Este programa es el Configurador de cobhttpd. Con este programa puedes modificar toda la información para que cobhttpd pueda trabajar. Puedes modificar el puerto TCP/IP a escuchar, directorios de trabajo, programas y campos de entrada.

COBHTTPD.EXE – Este programa es el que se necesita ejecutar para correr el servidor y atender peticiones de los clientes. Cuando este programa corre, el servidor estará listo para trabajar.

COBHTTPD_SVC.EXE – Este programa es una versión especial para funcionar en Windows 2000/XP como servicio. Un servicio no necesita que Windows entre a una sesión, por que el programa corre desde que Windows inicia. Así si un servidor Windows es reiniciado y nunca se acceda a una sesión de usuario, el servidor COBHTTPD estará listo para trabajar.

COBHTTPD_INT.EXE – Este programa es una interface entre el COBHTTPD cuando este funcionando como servicio, ya que un servicio no tiene intervencion con el usuario, este programa nos ayudara para monitorear los accesos que se hacen y que programas se ejecutan, nos sirve para cuando estemos haciendo pruebas.

COBVIEWHTTP.EXE – Este programa convierte los reportes generados por tus programas COBOL a PDF y regresarlos al servidor, para que este a su vez los regrese al Navegador del cliente.

COBHTTPD.DAT – Este archivo contiene la información de la configuración del programa cobhttpd. Contiene toda la información general.

Todos estos programas necesitan existir en el mismo directorio, ejemplo:

C:\COBHTTPD

Pero tu puedes poner tu instalación de COBHTTPD donde tu desees. Para COBHTTPD no es importante donde sea instalado.

COBHHTTPD Directorios.

El programa COBHHTTPD necesita de los siguientes directorios para trabajar.

Directorio de Instalación

Este directorio contiene todos los programas de cobhttpd y el archivo de configuración. Por default y para este documento usaremos: C:\COBHHTTPD.

Directorio de Documentos

Este directorio contendrá todos los archivos que serán publicados en el servidor web. Estos archivos publicados podrán ser: archivos html, archivos gif, jpg, y todos los archivos que se necesiten para crear tus formas html.

Este directorio podrá ser visto con tu browser, cuando visites el sitio, ejemplo:

<http://www.tudominio.com>

Si tu puerto es el 80, entonces esta dirección mostrara los archivos que existen en el directorio de documentos.

El primer documento visualizado será el “**Documento de default**”, registrado en la configuración de cobhttpd, por default nosotros lo llamamos: index.html

Si tienes un subdirectorio “website” dentro del Directorio de documentos, y el Directorio de documentos es “c:\cobhttpd\httpdoc”, entonces podrás colocar archivos especiales dentro del directorio: “c:\cobhttpd\httpdoc\website”, y podrías visualizar los archivos con la siguiente dirección:

<http://www.tudominio.com/website>

Por default, y como mejor practica, este directorio estará dentro del **Directorio de Instalación**.

Así, si tienes el Directorio de instalación hacia D:\COBHHTTPD, pondrás el directorio de documentos hacia D:\COBHHTTPD\HTTPDOC. Así cuando necesites hacer una copia completa de tu sitio, solo copias el directorio D:\COBHHTTPD, y todos los archivos podrás restaurarlos después.

Directorio de Reportes

Este directorio es para almacenar los archivos PDF que son visualizados por los browsers. Porque este directorio necesita ser accesado desde Internet, este directorio necesita existir dentro del **Directorio de Documentos**.

Por default nosotros llamamos a este directorio “reports”, así que, si nosotros tenemos nuestro **Directorio de Documentos** hacia D:\COBHHTTPD\HTTPDOC, y nuestro **Directorio de Reportes** es reports, el directorio completo en nuestro sistema sería: D:\COBHHTTPD\HTTPDOC\REPORTS

Y todos los reportes podrían ser accesados desde Internet de la forma:

<http://www.tudominio.com/reports/reporte.pdf>

Este directorio es depurado por cobhttpd, algún reporte generado y publicado en este directorio tiene solo 5 horas de vida antes de que cobhttpd borre este archivo.

Directorio de Temporales

Este directorio es necesario para crear archivos propios para la operación del programa. Este directorio contiene pequeños archivos creados por cobhttpd y accesados por tus programas COBOL para la conversión de datos, y para la creación del reporte que posteriormente será convertido a PDF.

Por default nosotros usamos: C:\TEMP o D:\TEMP para esto, es mejor tener un directorio de 8 letras como máximo, para los compiladores COBOL de MSDOS.

Este directorio es depurado por cobhttpd, igual que con el Directorio de Reportes.

Directorio XML

Este directorio contiene la definición XML de los proyectos y programas que son publicados en el Internet. Recuerda que cada programa necesita ser definido antes de que se pueda trabajar con él en Internet. Cada proyecto contiene un archivo con todos los programas.

Por default, el directorio necesita existir dentro del **Directorio de Instalación**, y necesita sea llamado xml.

Los archivos xml son mantenidos automáticamente por el programa de configuración del cobhttpd.

Instalación:

Este programa no contiene un programa de instalación, este programa fue hecho para “Desarrolladores de software”. Y como programador necesitaras conocer que cambios hace el programa a tu sistema operativo. Y si necesitas copiar la instalación solo copias el directorio a la misma ruta en otra computadora y corres el servidor y estarás seguro que te funcionara.

Para una instalación rápida, sigue las instrucciones.

- 1) – Desempaca el archivo cobhttpd.zip en tu directorio preferido
“c:\cobhttpd” es una buena opción. Con esta instalación obtienes los siguientes directorios:
 - httpdoc – Directorio para todos los documentos e imágenes
 - httpdoc/reports – Directorio para los reportes generados por cobhttpd.
 - Xml – Directorio para los archivos definiciones de proyectos y programas..
 - Cobol – El directorio con los programas COBOL que sirven de interfase entre tu aplicación COBOL y COBHHTTPD.
- 2) – Define tu directorio temporal, como por ejemplo: c:\temp.
- 3) – Esto es todo para una instalación, a continuación, realiza la configuración del programa, en la siguiente sección podrás encontrar toda la información necesaria para esta tarea.

Configuración

General

Para la configuración, necesitas correr el programa `cfgcobhttpd.exe`, toda la configuración se hace con este programa.

Este programa no necesita ningún parámetro, solo ejecútalo.

Primero iniciaremos con los datos generales para el programa, estos son:

Web Server port: (Puerto para Server) – Este dato define el puerto TCP/IP por el cual el servidor servirá las peticiones de los clientes. Si ahora tienes instalado ya un servidor web corriendo en tu computadora, entonces necesitas definir un puerto diferente al 80. El puerto 80 es el puerto de default para los servidores http, tu puedes definir algún puerto que no este siendo usado por otro programa.

Si defines el puerto 8080, entonces podrás acceder por medio de tu browser a la siguiente dirección:

<http://www.tudominio.com:8080>

<http://tu.direccion.ip:8080> (si lo accesas desde la lan)

<http://localhost:8080> (si lo accesas desde la propia computadora)

Default document: Es el primer documento que enviara el servidor cuando no se indique ningún archivo a leer desde el browser.

Default: `index.html`

Ejemplo: si visitas la siguiente dirección:

<http://www.tudominio.com/directorio>

El cobhttpd primero buscara el archivo `index.html`, si este existe en este directorio entonces lo regresara como resultado al browser del lado del cliente.

Document Dir: Este es el directorio de default para los archivos html y los reportes auto-generados.

Default: `c:\cobhttpd\httpdoc`

Nota: Puedes cambiar el Document Dir hacia otro diferente, ejemplo: `c:\httpdoc`, pero recuerda que cuando copies la instalación actual a otra computadora, necesitas copiar también todos los directorios para que trabaje correctamente.

TEMP Directory: Define el directorio de archivos temporales.

Default: c:\temp

Este directorio no es creado por el programa de configuración, necesita ser creado manualmente.

Allow external Development (OBSOLETA)– Habilita a COBHTTTPD para trabajar con el editor COBEDIT, este editor tiene la particularidad de que puedes crear proyectos web y automáticamente actualizar tu archivo de proyecto agregando los programas que crees en COBEDIT y además actualizar tu lista de variables para que no necesites crearlas manualmente con el programa de configuración. Por seguridad, debes deshabilitar esta opción cuando instales un servidor web en una máquina conectada a Internet.

Allow show Directory on Web – Si se marca esta opción, se mostrará el directorio de archivos y carpetas en el web, en caso de que no existiera el archivo de Default a mostrar. Solo usarla en la fase de desarrollo, y desactivarla en los servidores que trabajan en línea.

Show warnings in HTML mode – Muestra los mensajes de error en el browser, si es que algún error ocurre, errores como **no se encontró el archivo XML, Directorio de proyecto no existe, programa no existe.**

Si este dato no es marcado el error no es mostrado, y la ejecución del programa es detenida.

Save Activiy LOG – Si se marca esta casilla, cobhttpd genera un archivo de log con los eventos ocurridos, como un GET a un archivo, ejecución de un programa, un POST de una forma html, etc. El nombre del archivo es cobhttpd.log y se aloja en el Directorio de Instalación.

Bloque Cookies

COBHTTTPD necesita manejar cookies para poder mantener variables permanentes por sesión. Una sesión almacena un pequeño dato en el explorador del cliente, y el explorador debe tener la opción de aceptar cookies, de lo contrario las variables permanentes no se podrán reusar y siempre tendrán un valor inicial de cero.

Expire Cookies (mm) – Este dato indica el tiempo de vida que tendrá un cookie, el tiempo que se configure aquí es en minutos, es muy importante definir un tiempo de vida adecuado. Por ejemplo, si una instalación tiene un tiempo de vida de 30 minutos, cuando una persona accese a una aplicación con su password, esta sesión quedaria abierta durante los 30 minutos, hay que también colocar una opción para cerrar la sesión manualmente y asegurarnos que los usuarios la seleccionen para que se cierre la sesión y no esperar 30 minutos para que el COBHTTTPD lo haga automáticamente.

Not Autoexpire cookies – Si marcamos esta casilla, le decimos a COBHHTTPD que siempre que se use la aplicación se renueve el tiempo de vida de la sesión, así, si definimos un tiempo de vida de 10 minutos, mientras el usuario este usando la aplicación se le renovaran los 10 minutos nuevamente. Si no se marca esta opción, entonces la sesión durara 10 o 30 minutos aun cuando el usuario siga usando la aplicación, y se morirá la sesión después de ese tiempo y tendría que volver a hacer LOGIN con su usuario y password.

Domain – Este dato actualmente no se utiliza, la idea es poder definir un nombre del dominio para identificar los cookies en el explorador, pero actualmente no debe de llevar nada. Es para uso futuro.

Autorun Config

Autorun on windows start – Si se marca esta opción, cobhttpd automáticamente se ejecuta al iniciar Windows. Esta función necesita que el usuario inicie sesión en Windows, por que todos los programas que sé auto ejecutan inician cuando el usuario inicia sesión.

Put in systray – Esta opción permite que cuando se ejecuta el programa cobhttpd se coloque de manera invisible en la barra de tareas en ejecución (systray). Si no se marca, el programa se ejecuta y se abre la ventana de ejecución (en modo visible).

Run as NT Service – Esta opción aparece solo si la computadora donde es instalado cobhttpd es Windows 2000/2003 o XP.

Si se marca esta opción, entonces el programa cobhttpd correrá como un servicio de NT. El servidor cobhttpd se ejecutará cuando Windows inicie, aun si el usuario no a iniciado sesión. Cuando esta opción es marcada las opciones: “Autorun on windows start”, y “put in systray” son desactivadas, y todos los datos del bloque “Run as service” se activan.

Bloque **Run as service**

Install button – Instala el programa cobhttpd como servicio en esta computadora. Solamente podrá ser activado una vez. Para hacer clic en este botón se necesitan los datos username y password.

Uninstall button – Con este botón se desinstala el cobhttpd como servicio, cuando se haya instalado anteriormente.

Start button – Inicia el serivicio, cuando se haya instalado previamente, y no se este ejecutando en este momento.

Stop button – Detiene el servicio cuando se este ejecutando.

Username – Es el usuario para ejecutar el servicio. El usuario necesita existir en el sistema actual. Se puede poner de la forma: **.Usuario**

Password – Es el password para el usuario.

Recuerda o anota tu usuario y password, por que cuando se salve la configuración para el servicio, nunca volverá a ser mostrada por razones de seguridad. Y este dato no es grabado en el archivo de configuración cobhttpd.dat.

Boton Projects – Este botón permite acceder a la configuración de los proyectos, la cual será explicada en la siguiente sección.

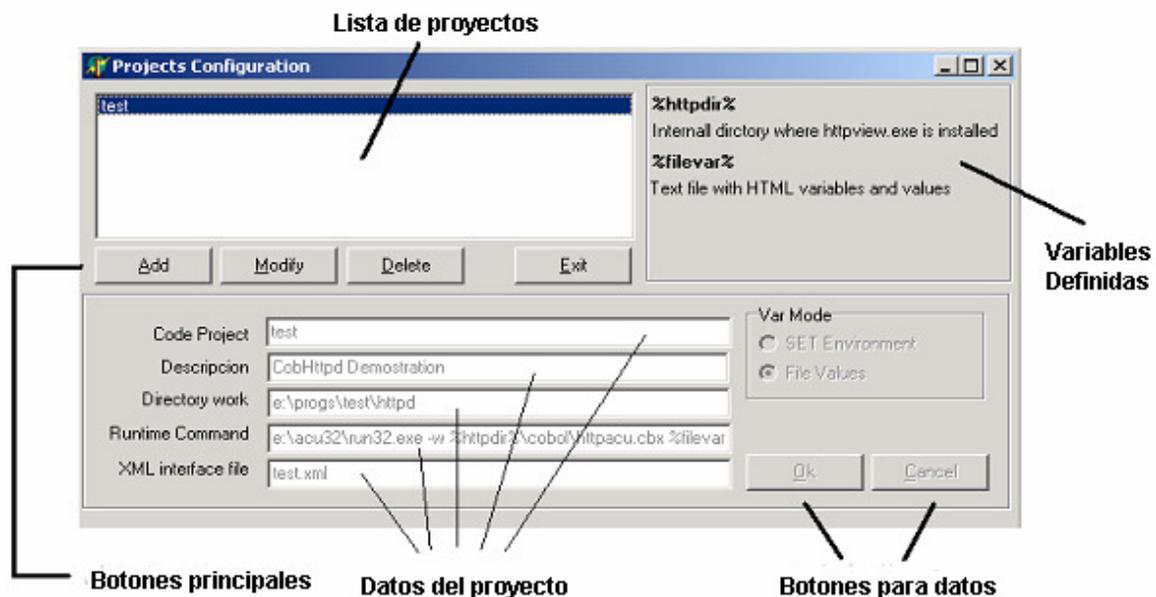
Configuración (Proyectos)

Los proyectos son grupos de programas que necesitan ser accedidos desde el Internet. Un proyecto contiene datos para los programas COBOL que son miembros de la misma aplicación COBOL.

Para indicar que proyecto usar en el url, el formato es:

`http://www.tudominio.com/cobpdf?proid=PROYECTO`

La función cobpdf en la llamada a este url será explicada mas adelante, lo más importante ahora es la manera como indicar el ID del proyecto. Este ID es único y global, siempre llamaremos a nuestro proyecto con este ID, y es universal y no podrá ser modificado después de registrarlo, sobre todo si ya lo usamos en nuestras formas html. La única manera de ejecutar un programa, es indicando a que proyecto pertenece. Este viene siendo como el nombre de tu aplicación, siempre es el mismo y nunca cambia. Cuando tu inicias a usar este ID en tus llamadas será difícil cambiarlo mas adelante.



La pantalla de configuración de proyectos contiene los siguientes componentes:

lista de proyectos

Esta es la lista de los proyectos definidos. Cuando haces clic en algún proyecto de la lista, todos los Datos del proyecto se muestran, pero no es posible modificarlos, porque el proyecto solo es seleccionado. Para hacer alguna acción es necesario hacer clic al botón apropiado antes.

- Add – Permite insertar un nuevo proyecto en la lista de proyectos. Cuando se selecciona este botón los datos del proyecto se ponen en blanco, y se activan para introducir datos.
- Modify – permite modificar los datos del proyecto actualmente seleccionado. Cuando se selecciona este botón los datos se activan para modificarse.
- Delete – Permite eliminar el proyecto actualmente seleccionado. Después de hacer clic en este botón el proyecto se elimina y se desaparece de la lista de proyectos.
- Exit – Cierra esta ventana y regresa a la pantalla principal de configuración.

Datos del proyecto:

Code Project

Es el ID del proyecto. Necesitar ser único, y de preferencia no cambiarlo mas adelante.

Description

Es una descripción corta del proyecto.

Directory work

Es el directorio donde tus programas COBOL se alojan. Siempre que tu programa COBOL se vaya a ejecutar, el directorio se cambia a este, antes de que tu programa se ejecute.

Runtime command

Es el programa que ejecuta tu programa COBOL. Si tu compilador COBOL genera programas auto-ejecutables, no hay ningún problema, solo tenemos que hacer una llamada directa al primer programa a ejecutar.

Normalmente se necesitan pasar parámetros especiales al runtime y al programa COBOL para ejecutar.

Existen dos variables que pueden ser usadas para hacer cambios especiales en este dato: `%httpdir%` y `%filevar%`. `%httpdir%` nos va a indicar el directorio de instalación de cobhttpd. Claro que también se podría usar el directorio directo, ejemplo: `c:\cobhttpd`, en lugar de usar la variable. Pero si cambiamos de directorio de instalación en otra computadora, tendremos que modificar nuestros proyectos para corregir la ruta. Este problema se resuelve con esta variable.

`%filevar%` es el archivo de datos que contiene los datos de entrada para nuestro programa COBOL. Nosotros no podemos saber el valor que tiene este archivo, pero podemos indicarlo como parámetro, y cobhttpd sustituirá esta variable con el nombre donde nos dejo los valores de los campos.

Para cada compilador de COBOL tenemos un pequeño programa que se usa como puente entre cobhttpd y nuestros programas COBOL. Este programa es diferente para cada compilador.

Ejemplos:

- Para ejecutar un proyecto en el directorio F:\APPS\ACCOUNT
- Desarrollado con Acucobol, el runtime es RUNCBL.EXE y este runtime esta en el directorio C:\ACU
- El ID del proyecto es: account

Code project: account

Directory work: F:\APPS\ACCOUNT

Runtime Command:

C:\acu\runcbl.exe -w %httpdir%\cobol\httpacu.cbx %filevar%

Si en nuestra instalación colocamos cobhttpd en c:\cobhttpd

Es lo mismo si ponemos

C:\acu\runcbl.exe -w c:\cobhttpd\cobol\httpacu.cob %filevar%

Pero recuerda, no siempre el directorio de instalación podrá ser c:\cobhttpd

Si en alguna otra instalación ponemos cobhttpd en d:\cobhttpd, tendremos que cambiar todos nuestros proyectos hacia este directorio.

El subdirectorio **cobol** es donde por default colocamos los programas internos para cada compilador.

Nosotros desarrollamos un programa para cada compilador.

En este caso, el dato *%filevar%* que el programa cobol recibe es como parámetro, y el programa puente hace varias cosas con el antes de ejecutar el programa COBOL de tu aplicación.

Ejemplo 2:

El mismo ejemplo, pero utilizando RMCOBOL en C:\RM

Runtime command:

C:\rm\runcobol %httpvar%\cobol\httpprm.cob a="%filevar%"

Compiladores COBOL soportados

A continuación se detallan los diferentes programas que se utilizan para cada compilador soportado por COBHTTPD, además ponemos un ejemplo de cómo quedaría la línea de **Runtime Command**.

ACUCOBOL, interface httpacu.cbx

El programa de interface a utilizar es httpacu.cbx, y fue creado usando acucobol ver. 3.2, se le pasa como parámetro la variable `%filevar%`

Ejemplo de "Runtime Command":

```
c:\acu\runcbl.exe -b -w %httpdir%\cobol\httpacu.cbx %filevar%
```

Puedes utilizar el parámetro `-b` para indicar que queremos ejecutar el programa como background, es decir si acucobol marca algún error este lo envía como salida de desplegado y lo podemos ver en la consola del COBHTTPD.

También puedes agregar el parámetro `-e` archivo, para crear un archivo de errores para monitorear si tu programa aborta de alguna manera y revisar el archivo y ver el error registrado.

RMCOBOL, interface httprm5.cob

Esta interface para RMCOBOL es para versión 5.36.

Ejemplo de "Runtime Command":

```
c:\rm\runcobol.exe %httpdir%\cobol\httprm5 K A="%filevar%"
```

En este compilador la variable `%filevar%` se pasa como parámetro.

MSCOBOL 5, interface httpms5.exe

Este compilador no necesita de runtime para ejecutar los programas COBOL. Debido a que es autoejecutable el programa interface, lo ejecutaremos directamente.

Ejemplo de "Runtime Command":

```
%httpdir%\cobol\httpms5.exe %filevar%
```

POWER COBOL 3, interface httpw3.exe

Este compilador no necesita de runtime para ejecutar los programas COBOL. Debido a que es autoejecutable, el programa interface, lo ejecutaremos directamente.

Ejemplo de "Runtime Command":

```
%httpdir%\cobol\httpw3.exe %filevar%
```

POWER COBOL 85, interface httpw85.exe

Al igual que el anterior, no necesita runtime, este se usa para programas compilados en Power COBOL 3, usando el COBOL 85.

Ejemplo de "Runtime Command":

```
%httpdir%\cobol\httpw85.exe %filevar%
```

Estas dos interfaces anteriores han sido creados para Power COBOL 3, los dos utilizan los programas que se van a llamar como DLL, de tal manera que hay que crear todos los programas en estos compiladores como DLL para que estas interfaces las puedan ejecutar.

POWER COBOL 5, interface httpw5.exe

Esta interface no necesita de runtime, ya que se manda a llamar directamente el programa de interface.

Ejemplo:

```
%httpdir%\cobol\httpw5 %filevar%
```

Con esto se define que el proyecto sera de POWER COBOL 5, y la ejecucion de los programas ya se llevara a acabo en el directorio donde se indicaron que estos seral alojados. Si descarga los ejemplos para este compilador, encontrara una guia de los pasos mas importantes para desarrollar un Proyecto WEB con Power COBOL 5.

Runtime Type

En este bloque debemos definir el tipo de Runtime que tenemos para correr nuestros programas, las opciones son:

MSDOS
WIN 16
WIN 32

Es muy importante seleccionar la opción adecuada, ya que COBHHTTPD ejecuta cada uno de estos tipos de runtime de manera diferente, y el seleccionar una opción incorrecta puede ocasionar que COBHHTTPD se quede colgado.

Si nuestro runtime es para MsDos y trabaja en modo texto, seleccionamos MSDOS. En cambio el Runtime es para Windows 3.11, debemos escoger WIN 16. Y por ultimo, es es un runtime para Windows 98 o superior, escojeremos WIN 32.

Process Show

Este bloque es para especificarle a COBHHTTPD como correrá el runtime, en ocasiones al ejecutar un runtime, este automáticamente muestra una ventana que se abre al ejecutar este programa. Si deseamos hacer que esta ventana no se muestre seleccionaremos la opción HIDE, algunos runtimes necesitan una configuración especial para indicarle que no muestre esta ventana inicial, tal es el caso de RMCOBOL.

Por otro lado, si deseáramos desplegar algún tipo de dato para poder depurar nuestro programa al estar haciendo pruebas, deberemos de escoger la opción SHOW momentáneamente, y de esta manera podremos ver los despliegues que hacemos con nuestro programa.

Hay que aclarar que en un servidor ya trabajando en linea, debemos escoger siempre HIDE.

Botón Edit XML

Con este botón se puede editar el archivo xml, con un pequeño editor de xml integrado en cobhttpd.

Cuando se hace clic en este botón, aparece el editor de xml, donde puedes realizar las siguientes operaciones:

- Insertar nuevos programas
- Modificar programas ya existentes
- Agregar campos a los programas
- Eliminar un programa o algún campo

La otra manera de modificar el archivo xml, es editándolo directamente con algún editor de textos.

En las siguientes dos secciones explicare primero como es el formato interno del archivo xml, posteriormente en detalle explicare las funciones completas del editor xml.

Formato del archivo de interfase XML

El archivo de interfase xml tiene primero un elemento root: httpview

ejemplo:

```
<httpview>  
  Definición de programas  
</httpview>
```

Definición de Programas:

Cada programa necesita tener el elemento <program> con los siguientes parámetros:

Name="nombre-programa"

ejemplo:

```
<program name="prog01">  
  definición de campos  
</program>
```

Campos de un programa.

Los campos se definen con la marca <field>.

Parámetros:

Type – tipo de campo, los valores son, 9-números, X-alfanumérico

Size – Tamaño del campo

Int – Cantidad de enteros

Dec – Cantidad de decimales

Sign – Indica si el numero lleva signo

El valor del elemento es el nombre del campo que se usara en la forma html.

Ejemplo:

```
<field type="X" size="10">NOMBRE</field>
```

Ejemplo completo:

```
<httpview>
  <program name="prog01">
    <field type="X" size="10">NOMBRE</field>
    <field type="X" size="20">APELLIDO</field>
  </program>
  <program name="prog02">
    <field type="9" int="3">PAGINI</field>
    <field type="9" int="3">PAGFIN</field>
  </program>
</httpview>
```

prog01 Fuente de la interfase:

```
. . .
01 DATA-INPUT.
   02 INPUT-NOMBRE      PIC X(10).
   02 INPUT-APELLIDO   PIC X(10).
. . .
```

Prog02 Fuente de la interfase:

```
. . .
01 DATA-INPUT.
   02 PAGINI           PIC 9(03).
   02 PAGFIN          PIC 9(03).
. . .
```

Editor XML

Este editor es parte de cobhttpd, y puede ser accesado desde el botón **“Edit xml”**.

Cuando se hace clic en este botón, el archivo xml del proyecto actual es abierto, y es cargado en el editor xml.

El archivo es mostrado en modo de Arbol.

El editor de xml contiene las siguientes opciones de menú:

File

New - Crea un nuevo archivo xml en blanco, limpia el contenido actual

Open – Abre un archivo xml existente

Save – Salva el contenido actual en el archivo

Exit – Cierra el editor de xml

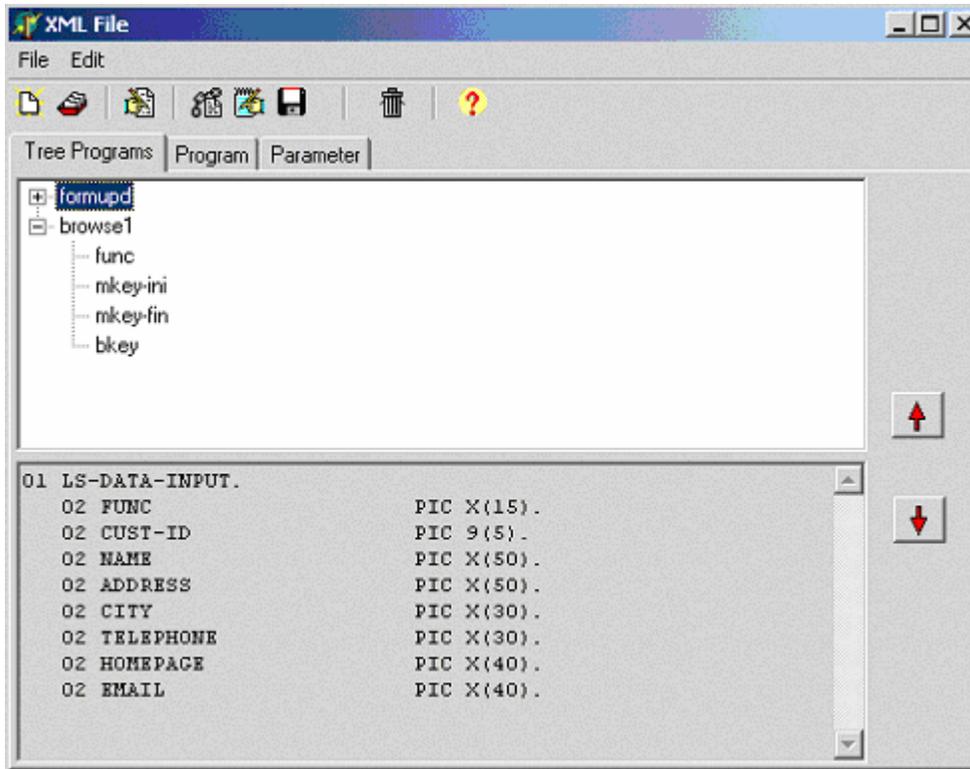
Edit

New program – Inserta un nuevo programa en el archivo xml actual

New parameter – Inserta un nuevo campo en el programa actualmente seleccionado

Modify – Modifica el programa o el campo actualmente seleccionado

La vista actual del editor de xml es la siguiente:



Cuenta también con un tool-bar superior, para realizar las operaciones mas importantes del programa:

New Program – Crea un nuevo programa, igual que la opción del menú.

New Parameter – Crea un nuevo campo parámetro, igual que en menú.

Modify program o parameter – Permite modificar el elemento seleccionado actualmente.

View Linkage – Con este botón podemos ver una estructura de la LINKAGE-SECTION, de como se recibirá en nuestro programa. Podemos incluso marcar y copiar el contenido, y pegarla en nuestro programa.

Edit code for import fields – Permite editar el registro de campos de entrada para el programa, utilizando la sintaxis COBOL para una fácil edición de campos, también es posible copiar el contenido del registro directamente desde el programa fuente COBOL.

Save file – Graba el archivo xml.

Delete program – Elimina el programa seleccionado.

Help – Muestra una ventana de ayuda.

Flechas arriba y abajo – Con estas flechas de dirección, puedes colocar un campo arriba o hacia abajo, para la definición correcta en tu registro de la LINKAGE SECTION.

Cuando seleccionas un programa diferente, aparece en la parte inferior el contenido completo del registro de campos, tal y como lo recibira nuestro programa COBOL.

Como agregar un nuevo programa:

Para esta operación, seleccionamos la opción del menú Edit->New program, o hacemos clic en el botón New Program.

Automáticamente cambiara hacia la carpeta Program.

En esta carpeta solo tenemos activo el dato name.

Después de teclear el nombre del programa, hacemos clic en el botón OK para agregar el programa al árbol XML.

Después de insertar nuestro nuevo programa, podremos seleccionarlo e insertarle los campos que necesitemos.

Como modificar un programa:

Primero necesitamos seleccionar el programa que queremos modificar.

Después, hacemos clic en la opción del menú Edit->Modify program.

El único dato que podemos cambiar es el nombre del programa: **Name**

Como insertar un nuevo campo parámetro:

Primero, seleccionamos el programa donde queremos insertar el campo.

Después, clic en el botón New Parameter, o en la opción del menú Edit->New Parameter

Llenamos la forma de datos para el nuevo campo.

Name, Type (X or 9), Size.

Si nuestro nuevo campo se define como numérico, entonces es necesario definir la cantidad de enteros y decimales que tendrá. También indicar si necesitamos el signo.

Al final hacemos clic en el botón OK para agregar el nuevo campo.

Como modificar un campo parámetro:

Primero, seleccionamos el campo que se desea modificar.

Hacemos clic en el menú Edit->Modify parameter.

Cambiamos la información del campo en la carpeta Parameter.

Hacemos clic en el botón OK.

Una vez que hemos terminado de hacer las modificaciones necesarias a nuestro archivo de proyecto xml, grabaremos el archivo con la opción del menú File->Save.

Para terminar de usar el editor xml, nos salimos con la opción: File->Exit.

Ejecutando el programa Server COBHHTTPD

Para ejecutar el programa cobhttpd, y ponerlo a escuchar peticiones de los clientes browser, se ejecuta uno de los siguientes programas.

Cobhttpd.exe, Si el Windows es Windows 98, NT/200/XP y actualmente no se ejecuta como servicio..

Cobhttpd_svc.exe – Se ejecuta automáticamente si el Windows es NT/2000/XP y se ha configurado como servicio. Este programa se ejecuta al arrancar el Windows.

En Windows 98, puedes configurar para que tu programa se corra automáticamente y se coloque en el systray. Para abrir la ventana del programa en este modo, hacer doble-clic en el icono y la ventana principal del programa será abierta.

Si se a optado para ejecutar cobhttpd como servicio, entonces arrancara sin necesidad de hacer LOGON en la computadora, pero al correr asi no tendremos manera de hacer un monitoreo de nuestros programas COBOL que se van ejecutando. Para hacer un monitoreo de los accesos cuando este funcionando como servicio, usamos el programa **cobhttpd_int.exe**, este se conectara con el programa servicio y mostrara unicamente los accesos que se realicen.

Ejecutando los programas COBOL

Existen dos maneras de ejecutar un programa COBOL. El modo a utilizar dependerá del tipo de contenido a regresar.

Generando Contenido HTML:

Si deseamos regresar contenido HTML usaremos:

`http://www.tudominio.com/cobcgi?proid=PRO&prog=PROG&F1=value`

Generando Reportes PDF:

Si deseamos regresar un report PDF a el navegador, utilizaremos el modo:

`http://www.tudominio.com/cobpdf?proid=PRO&prog=PROG&F1=value`

Acceso directo:

Desde el campo dirección, en el web browser, puedes correr un programa de la siguiente manera:

`http://www.tudominio.com/cobpdf?proid=PRO&prog=PROG&F1=value`

Con este dato, puedes correr un programa llamado PROG en el proyecto PRO, y recibirá el campo F1 con el valor indicado en value.

Recuerda que, si definiste un puerto diferente al 80, entonces necesitas indicar este puerto en tu Server, ejemplo, si defines el puerto 85:

`http://www.tudominio.com:85/cobpdf?proid=XXXXXXX`

Llamando desde archivos HTML

Para una llamada desde formas html, solo coloca la dirección en el ACTION de una marca FORM:

```
<form method="POST" action="/cobpdf?proid=PRO&prog=PROG">  
  ... definición de campos de la forma  
</form>
```

Para gente que conoce poco de html, si ves en el action, podrás ver que no escribí el url completo. Esto no es necesario, por que el browser se encarga de expandir este url con el nombre del dominio de donde descargo este html, y con el puerto del webserver. A esto se le conoce como dirección relativa.

Pero, si tienes el html de la forma en un web Server diferente, y solo buscas correr tus programas COBOL con el servidor cobhttpd, entonces tu POST podría verse así:

```
<form method="POST"
action="http://www.tudominio.com/cobpdf?proid=PRO&prog=PROG">
... definición de campos de la forma
</form>
```

A esta dirección se le conoce como dirección absoluta.

Si quieres pasar un dato, pero este dato no tiene campo INPUT dentro de la forma html, puedes pasar este dato como parte del url, ejemplo:

```
<form method="POST" action="http://www.tudominio.com/cobpdf?
proid=PRO&prog=PROG&func=add">
```

En este ejemplo, el campo **func**.

Este campo es pasado al programa PROGNAME, y este programa necesita tener definido este campo (func) en su lista de campos también.

Cuando nuestro programa COBOL se ejecute, func también le llegara, igual que los campos que están definidos en la forma HTML con las marcas INPUT.

Diferencia entre contenido HTML y PDF

Para contenido html usar **/cobcgi?proid=pro&prog=prog**

Para contenido pdf usar **/cobpdf?proid=pro&prog=prog**

En ambos casos el archivo a generar debe de llamarse tal y como nos indique la variable: HTTP-OUTPUT (esta variable se explica mas adelante en el registro HTTP-CONTROL).

Recibiendo datos de entrada en nuestro programa COBOL

Hasta este momento, nuestro programa COBOL recibe dos registros de datos:

```
01  HTTP-CONTROL .
    02  HTTP-PROJECT      PIC X(20) .
    02  HTTP-PROGRAM     PIC X(20) .
    02  HTTP-OUTPUT      PIC X(150) .
    02  HTTP-ERROR       PIC 9(04) .
    02  HTTP-ERROR-MSG   PIC X(50) OCCURS 5 TIMES .

01  HTTP-DATA-INP .
    02  HTTP-DATA-INPX   PIC X(4096) .
```

Estos dos registros se encuentran ya definidos en el archivo **LS-HTTP.FDT** que se encuentra en el directorio COBOL dentro de COBHHTTPD, también va en cada ejemplo ya que se necesita para compilar los programas.

HTTP-CONTROL

El primer registro: HTTP-CONTROL contiene algunos datos importantes para nuestro programa, y que son pasados de COBHHTTPD hacia nuestro programa COBOL que será ejecutado:

HTTP-PROJECT – es el código del proyecto al que pertenece nuestro programa que se está ejecutando. Nos puede servir dentro de nuestro programa para hacer enlaces hacia urls y colocar el mismo código de proyecto que en el archivo html externo.

HTTP-PROGRAM – es el nombre del programa que se está ejecutando, y nos puede servir para hacer librerías externas y hacer referencia hacia el programa que se está ejecutando, y no tener que poner el nombre del programa fijo, si es que nuestra librería sirve para diferentes programas.

HTTP-OUTPUT – Es el nombre del archivo que debemos de generar con nuestro contenido html que deseamos.

HTTP-ERROR – Es un código de error que podemos regresar a COBHHTTPD, cualquier número diferente de 0 es un error (no usado por el momento).

HTTP-ERROR-MSG – Es un arreglo de 5 renglones donde podremos colocar un mensaje textual para regresar al cliente en caso de que el programa necesite informar algún error, al ser este HTML es necesario regresar el error como parámetro. Trabaja en conjunto con HTTP-ERROR. (no usado por el momento).

HTTP-DATA-INP

Este segundo registro, es un solo campo de 4 kb de tamaño. En este campo nos llegarán los diferentes valores que nuestro programa COBOL necesita para funcionar.

Al ser de un máximo de 4 K podemos incluso recibir grandes cantidades de datos para realizar operaciones de Actualización de archivos.

Ya en nuestro programa hacemos la división de campos declarando un registro en la **WORKING-STORAGE**, y solo movemos este registro a nuestro registro interno para tener la división real. Ejemplo:

```
WORKING-STORAGE SECTION.  
  
01  REG-INTERNO.  
    02  MFUNC                PIC X(15) .  
    02  MKEY-INI             PIC 9(05) .  
    02  MKEY-FIN             PIC 9(05) .  
    02  BKEY                 PIC 9(05) .  
  
PROCEDURE DIVISION.  
INICIO.  
    MOVE HTTP-DATA-INP  TO REG-INTERNO.  
    MOVE HTTP-OUTPUT    TO WF-CGI-OUT.
```

De esta manera usamos los campos de REG-INTERNO en nuestro programa.

Como generar la salida a entregar al cliente

Ya sea generando contenido HTML o generando un reporte para ser entregado como PDF debemos de grabar el contenido en un archivo texto.

COBHHTTPD nos indicara como se debe de llamar el archivo texto que debemos generar. Este nombre de archivo nos llegara al programa en la variable HTTP-OUTPUT. Esta variable tiene el nombre del archivo. Si no generamos el contenido con este nombre el resultado no será entregado al cliente que solicito la consulta.

Ahora, desde COBOL podemos utilizar la siguiente declaración para crear este archivo:

```
... ..
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT IMPRESO
        ASSIGN TO INPUT-OUTPUT WF-REPORT
        ORGANIZATION IS LINE SEQUENTIAL
        FILE STATUS IS STAT-IMP.
... ..

DATA DIVISION.
FILE SECTION.
FD IMPRESO
    LABEL RECORDS ARE OMITTED.
01 LINEA          PIC X(500).

WORKING-STORAGE SECTION.

77 WF-REPORT      PIC X(150).
77 STAT-IMP        PIC X.

... ..
LINKAGE SECTION.

COPY "LS-HTTP.FDT".

PROCEDURE DIVISION USING HTTP-CONTROL HTTP-DATA-INP.
000-INICIO.
    MOVE HTTP-OUTPUT TO WF-REPORT.
    OPEN OUTPUT IMPRESO.

    WRITE LINEA FROM "<html><head>".
    WRITE LINEA FROM "<title>ejemplo html</title>".
    WRITE LINEA FROM "</head>".
    WRITE LINEA FROM "".
```

```
WRITE LINEA FROM "<body>".
WRITE LINEA FROM "<center>".
WRITE LINEA FROM "<h1>Primer prueba html</h1>".
WRITE LINEA FROM "</center>".
WRITE LINEA FROM "</body>".
WRITE LINEA FROM "</html>".

CLOSE IMPRESO.
STOP RUN.
END PROGRAM.
```

En este pequeño ejemplo hacemos solamente un desplegado de un letrero que debe de decir:

Primer prueba html

Todas las marcas utilizadas aquí son html, así que no debe de haber ningún problema con esa parte.

Lo interesante primero es:

- Utilizamos el archivo LS-HTTP.FDT para obtener todas las variables que nos entrega COBHTTPD
- Utilizamos el nombre del archivo que nos llega en HTTP-OUTPUT
- Creamos el archivo con el contenido deseado

Para crear contenido PDF es idéntico, solo que las marcas que usaremos para crear un reporte PDF deben ser las mismas marcas que para crear un reporte para COBVIEW. Hay que recordar que los reportes que se visualizan en Windows con COBVIEW pueden ser también creados para internet y visualizados en formato PDF que es la razón por la que se hace esta conversión.

Manejo de sesiones

COBHTTPD nos permite manejar sesiones, con el fin de recuperar un valor de una variable la siguiente vez que el usuario accese al mismo programa.

Como es sabido, el protocolo http solo regresa el contenido de una consulta. Una vez que el contenido es regresado al navegador se pierde la conexión entre el servidor y el cliente. De tal manera que la siguiente vez que se realiza la consulta a otro documento se hace una conexión nueva y los datos no se pasan de una consulta a otra. Aun cuando se trate del mismo documento.

Para evitar este problema COBHTTPD maneja sesiones, las cuales permiten grabar valores y pasarlos de consulta en consulta durante el tiempo que dure la sesión.

Las sesiones siempre se crean y se actualizan en cada consulta. Para grabar un valor en una sesión desde nuestro programa COBOL, debemos de crear una marca con el siguiente formato:

```
<@ SESSION SAVE variable=valor @>
```

Ejemplo:

```
<@ session save clave=1953 @>
```

Desde dentro de nuestro programa COBOL generaríamos la instrucción de la siguiente manera:

```
STRING "<@ session save clave=", MEM-CLAVE, " @>"  
      DELIMITED BY SIZE INTO MEM-LINEA.  
WRITE R-CGI-OUT FROM MEM-LINEA.
```

Para recuperar el valor que hemos grabado en la sesión, debemos de tomar el valor como un parámetro de entrada. Es decir registrarlo en la Configuración de COBHTTPD como variable de entrada de nuestro programa, y declararlo en el registro para poder tener el valor siempre que se entre a nuestro programa.

Dicho de otra manera, la variable siempre nos llegara como parámetro, y dentro de nuestro programa debemos de actualizar este valor con la instrucción SESSION SAVE para que la próxima vez que se entre al mismo programa nos llegue este valor.

Otra solución para resolver este tipo de problemas es que siempre enviemos el valor dentro del url del programa, así se pasa un valor como parámetro incluido en la misma dirección.

Incluir archivos externos

COBHTTPD también permite incluir archivos externos. Debido a que son de gran ayuda ya que podemos agrupar algunas librerías o rutinas en un archivo externo e incluirlo en cada página HTML que generemos.

Estos archivos pueden ser partes de código HTML o bien rutinas Javascript que podemos necesitar en nuestra página web que generemos.

Para hacer esto, usamos la siguiente marca especial de COBHTTPD:

```
<@ COPY "nombre-de-archivo.ext" @>
```

El nombre de archivo debe de ir encerrado entre comillas o apóstrofes.

La ubicación física del archivo siempre debe de ser a partir del directorio donde se está ejecutando el programa.

Es decir, el directorio que tenga configurado el Proyecto en la configuración del COBHTTPD para correr los programas del proyecto.

Si deseas meter por ejemplo todos tus archivos de JavaScript en un directorio llamado js, este debería de estar dentro del directorio de programas del proyecto.

Así, para incluir un archivo llamado formas.js que se guardó en el directorio js, la línea para COBHTTPD debe de ser:

```
<@ COPY "js\formas.js" @>
```

Con esta línea, COBHTTPD va a sustituir este texto por el contenido del archivo formas.js. Esto lo hace de manera automática.

Introducción a la librería CGILIB

CGILIB es una librería para crear programas tipo CGI desde COBOL y creada específicamente para usarse con COBHHTTPD.

Esta librería consta de una serie de archivos que se incluyen en nuestro programa COBOL, y con base en puras llamadas a párrafos COBOL podremos crear contenido HTML de manera dinámica.

Las capacidades de CGILIB son:

- Crear contenido HTML sin escribir ninguna marca de este lenguaje
- Poder crear tablas para dar formato a nuestras pantallas
- Crear formas para captura y presentación de datos en HTML
- Librería para crear un BROWSER y poder navegar a través de los registros de un archivo.
- Poder crear botones y links a urls.
- Creación de temas para poder cambiar la apariencia de nuestro programa con solo indicar el nombre del nuevo tema.
- Manejo fácil de Formas templates creadas con editores de html

Los archivos de esta librería se encuentran ubicados en el directorio CGILIB dentro del directorio de programas de COBHHTTPD, ejemplo C:\COBHHTTPD\CGILIB.

Necesitas estudiar también el manual de esta librería para que puedas comprender el uso de cada archivo de código que contiene, y cuando necesitas usar uno y otro.