

COBHTTPD

Servidor de Aplicaciones para COBOL.

Manual del Usuario

(Ver. 4.80)

INDICE

Descripción.....	4
Como trabaja este programa.....	5
COBHTTPD Archivos y Programas	6
COBHTTPD Directorios.....	7
Directorio de Instalación.....	7
Directorio de Documentos.....	7
Directorio de Reportes.....	8
Directorio de Temporales	8
Directorio XML.....	8
Configuración.....	10
General.....	10
Autorun.....	12
DynDNS	13
Mime.....	15
Alias.....	16
Auth	17
Configuracion	17
Usuarios	17
Carpetas	17
COBAPP.....	18
General.....	18
Proyectos	19
Compiladores COBOL soportados	23
ACUCOBOL, interface httpacu.cbx	23
RMCOBOL, interface httpm5.cob	23
MSCOBOL 5, interface httpm5.exe.....	23
POWER COBOL 3, interface httpwc3.exe	24
POWER COBOL 85, interface httpw85.exe	24
POWER COBOL 5, interface httpw5.exe	24
Botón Editar Programas.....	26
Formato del archivo de interfase XML	27
Editor de Programas	29
Send Mail.....	32
HTTPS (nuevo a partir de versión 4).....	34
Configuración General:	34
Pestaña: Self Signed	36
CACHE (nuevo a partir de versión 4)	38
Manejo de Black List de IPs sospechosas	40
Ejecutando el programa Server COBHTTPD.....	44
Ejecutando los programas COBOL	45
Generando Contenido HTML:.....	45
Generando Reportes PDF:	45
Acceso directo:	45
Llamando desde archivos HTML	45
Diferencia entre contenido HTML y PDF	46

Recibiendo datos de entrada en nuestro programa COBOL.....	47
HTTP-CONTROL	47
HTTP-DATA-INP	48
Como generar la salida a entregar al cliente	49
Manejo de sesiones	51
Incluir archivos externos.....	52
Introducción a la librería CGILIB	53

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 desees 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.

COBHTTPD Directorios.

El programa COBHTTPD 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:\COBHTTPD.

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:\COBHTTPD, pondrás el directorio de documentos hacia D:\COBHTTPD\HTTPDOC. Así cuando necesites hacer una copia completa de tu sitio, solo copias el directorio D:\COBHTTPD, 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 seria: 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 COBHTTPD.
- 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:

Puerto: (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)

Documento Inicial: 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.

Dir Documentos: 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.

Dir Temporales: Define el directorio de archivos temporales.

Default: c:\temp

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

Muestra Directorio – Si se marca esta opción, se mostrara 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 desctivarla en los servidores que trabajan en linea.

Muestra Errores en HTML – 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.

Graba 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.

Autorun

Bloque **Instalación de Autorun**

Ejecutar como Servicio – 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: “Autoejecutar al arrancar Windows”, y “Poner en la bandeja del Sistema” son desactivadas, y todos los datos del bloque “Ejecutar como Servicio” se activan.

Auto ejecutar al arrancar Windows – 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 se auto ejecutan inician cuando el usuario inicia sesión.

Poner en la bandeja del Sistema – 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).

Bloque **Ejecutar como Servicio**

Boton Instalar – 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.

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

Boton Iniciar – Inicia el servicio, cuando se haya instalado previamente, y no se este ejecutando en este momento.

Boton Detener – Detiene el servicio cuando se este ejecutando.

Usuarioe – 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.

DynDNS

Este modulo nos permite actualizar nuestra IP en algun servidor de DNS dinámico Gratuito como DYNDNS.ORG o NO-IP.COM

Estos servicios son utiles si por ejemplo tenemos una conexion DSL que nos cambia la IP dinámicamente, y deseamos accesar nuestro equipo desde fuera usando algún Nombre como por ejemplo: miserver.dyndns.org

Para esto, primero nos registramos en alguno de estos servicios gratuitos, y luego configuramos los datos de nuestra cuenta en el Web Server.

Tipo de DDNS - Actualmente podemos trabajar con dyndns.org y no-ip.com

Usuario - Es el nombre de usuario con el que nos registramos en el servicio de DNS dinamico.

Password - El password que usamos cuando registramos nuestra cuenta.

Nombre del Host - Es el nombre del host que creamos con nuestra cuenta, y a el cual le estaremos actualizando la IP cuando nos vaya cambiando, ejemplo: myserver.dyndns.org

Autodetectar IP Wan - Esta opcion es importante, ya que muchas veces las conexiones DSL nos cambian la IP constantemente, y el mismo Web Server puede estar monitoreando el cambio de esta IP para que cuando suceda se lo reporte automáticamente a el Servidor de DNS y asi nuestro host apunte siempre hacia nuestra nueva IP.

Si esta opción no se marca, entonces se activa el dato: IP Fija, ya que si no es dinámica nuestra IP, entonces la IP Fija nunca cambiara, y se usara la que se ponga en este dato.

Use Router IP – Si no deseamos salir a Internet para investigar cual es la nueva IP, podemos intentar sacarla del propio router, ya que muchas veces en la configuración del estatus del router viene la ip nueva, y de ahí podemos también tomarla. Al marcar esta opción se activan dos datos: URL del Router y Texto antes de la IP.

IP Fija - Si tenemos algun servicio donde nuestra IP nunca cambia, entonces en este campo tecleamos la IP que nos dieron, y el Web Server actualizara la IP en el HOST solamente la primera vez.

URL del router - En este campo, podemos teclear la direccion para acceder a nuestro Router, y asi monitorear el cambio de IP, sin necesidad de estarlo haciendo cada 5 minutos, si no que con este metodo lo podemos checar cada 5 segundos, sin afectar el ancho de banda de nuestra conexión, ya que asi no haríamos consultas hacia el exterior. Para usar este método, necesitas saber la pagina que te regresa el estado de tu conexion ADSL, y que además incluya la IP publica de tu conexión.

Texto antes de la IP - Es algún texto que se encuentra un poco antes de donde aparece la IP publica de tu conexión, en la pagina del Router.

Sirve para identificar la IP pública, de alguna otra IP que también aparezca en la página de estado del router.

En algunos casos, dice: WAN IP, pero esto depende de el tipo de modem o de router que uses en tu conexión ADSL.

Mime

Se refiere un tipo de medio para internet, y es regresado en la sección de headers de las respuestas que arroja el servidor hacia los clientes.

Específicamente, es el resultado del campo Content-type.

Cada extensión tiene un tipo de Mime específico, y que es regresado al cliente cuando se solicita un archivo de X tipo.

Extension - Es la lista de extensiones, si hay varias para el mismo MIME, las separamos por espacio.

Mime Type - Es el tipo de Mime que se regresara cuando sea solicitado un archivo de este tipo.

Por ejemplo, el mas común, y que se usa en los documentos HTML, es el tipo text/html

Alias

El modulo de Alias nos ayuda a re direccionar un URL especial, hacia algún directorio en particular dentro del Sistema Operativo.

Por Ejemplo: `http://...domain.com/test/index.html`, el directorio `/test/`, podemos redireccionarlo hacia `c:\test\index.html`, en lugar de que el archivo que entreguemos sea el de `c:\WebServer\httpdoc`

Activar Alias - Esta opcion nos sirve para activar o desactivar el manejo de alias de directorios.

Directorio Alias - En esta lista podemos registrar varios directorios alias. Estos directorios deben empezar con /

Directorio Fisico - Es la ruta fisica en el servidor.
Ejemplo: Alias: `/test` Directorio: `c:\test`

Auth

Este modulo nos sirve para manejar una lista de directorios o carpetas que deseamos proteger con contraseña.

Este método se llama Autenticación Web.

Configuracion

Activar Auth - Este checkbox nos sirve para activar o desactivar el modulo de Auth.

Tipo - Actualmente el Web Server puede trabajar con los métodos BASIC y DIGEST.

Título - Aquí podemos colocar un titulo que aparecerá en los Browsers de los clientes, al momento que el Browser les solicita la identificación (Usuario y Contraseña)

Usuarios

Aquí podemos Agregar, Modificar y eliminar los usuarios que tenemos registrados y que pueden identificarse para entrar a los directorios protegidos.

Los datos que necesitamos para los usuarios son:

Usuario - Este vendría siendo el nombre de la cuenta

Password - La contraseña del Usuario

Nombre Completo - Este dato es solo para identificar al usuario, ya que realmente los únicos que se necesitan son el par Usuario y Password

Carpetas

En esta sección podemos configurar las diferentes carpetas o directorios que deseamos proteger con autenticación.

Cada carpeta o folder, debe ser un path valido en el sistema de archivos, y debe ser registrado con la ruta completa, para la correcta validación.

A cada usuario, también le podemos configurar la lista de usuarios que podrán acceder a el.

Path - Aquí podemos registrar la lista de carpetas protegidas.

Podemos usar el botón Explorar, para abrir un explorador de carpetas, y navegar por el disco duro hasta seleccionar la carpeta a proteger.

Usuarios - En esta parte de la lista, podemos configurar los usuarios que tienen derecho a acceder a cada carpeta protegida. Si tenemos varios usuarios con derecho, los podemos teclear separadas por espacio.

Podemos usar el botón Usuarios para que nos aparezca una ventana con la lista de usuarios y seleccionar los que deseamos tengan derecho sobre la carpeta.

COBAPP

General

Activar interface COBAPP

Sirve para indicar si se activa o no la ejecución de aplicaciones COBOL.

Dir de Reportes – Es el directorio publico donde se colocaran los reportes generados y convertidos a PDF.

Debe de estar dentro de el directorio publico: httpdoc.

Allow uploads – activa la recepción de archivos por medio de formas web. Si no se activa no podrán hacer uploads desde la pagina web (ver template y programa ejemplo)

Allow keep-alive – activa el manejo de keep-alive en las conexiones. Acelera un poco la creación de sockets al reusar conexiones ya creadas. Si se desea usar lo recomendable es seleccionar la opción 3-Depend on request, ya que el cliente también debe de poder manejar keep-alive.

Bloque Cookies

COBHTTPD 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 COBHTTPD lo haga automáticamente.

Not Autoexpire cookies – Si marcamos esta casilla, le decimos a COBHTTPD 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.

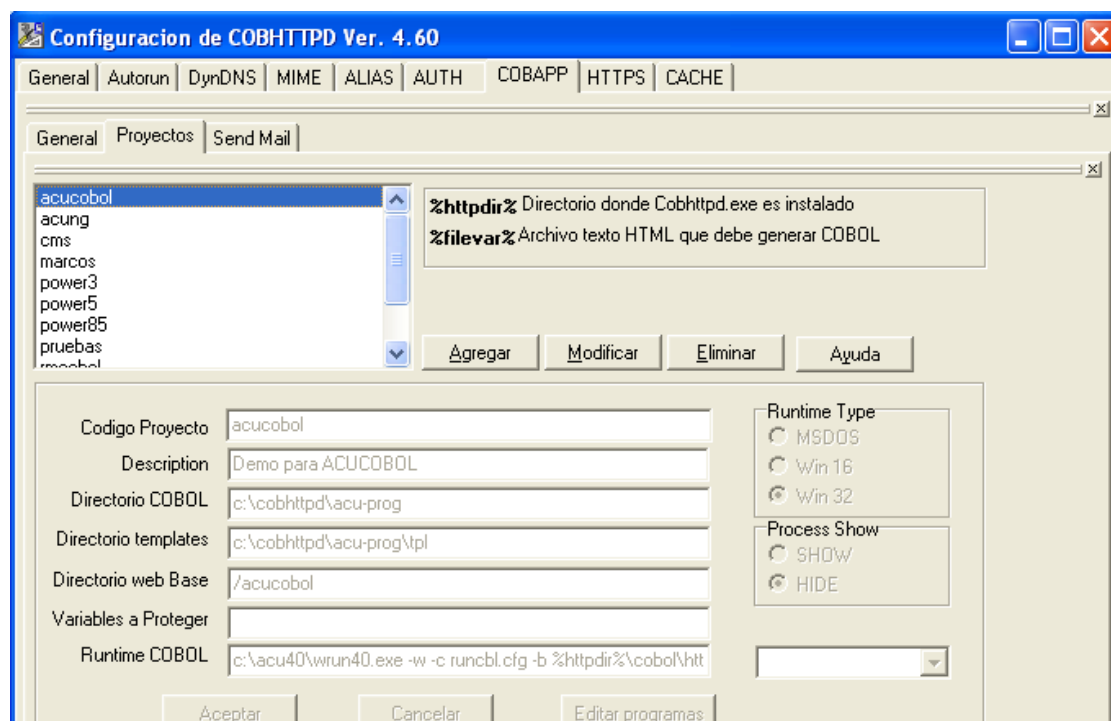
Proyectos

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.

- **Agregar** – 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.
- **Modificar** – permite modificar los datos del proyecto actualmente seleccionado. Cuando se selecciona este botón los datos se activan para modificarse.
- **Eliminar**– 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.

Datos del proyecto:

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

Descripcion - Es una descripción corta del proyecto.

Directorio COBOL - 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.

Directorio Templates – Es el directorio donde los templates HTML son colocados.

Directorio WEB Base - Es el directorio donde se colocan las imagenes y hojas de estilo (CSS), ej: Si necesitamos tenemos diferentes proyectos, y tenemos un directorio de imágenes para cada proyecto, podemos indicar el directorio base de nuestro proyecto, si tienes un proyecto EJEMPLO, podemos indicar el directorio base en: /EJEMPLO

Este directorio en tu webserver seria: c:\cobhttpd\httpdoc\ejemplo.

Si tu dejas este directorio en blanco, entonces los archivos necesarios para generar la pagina serian colocados en el directorio root c:\cobhttpd\httpdoc

Variables a Proteger - Son variables que por seguridad no deseamos que se pasen por el url, del tipo:

http://midominio.com?cobcgi&prog=PROYECTO&prog=TEST&USUARIO=1001

Esto nos ayuda por ejemplo, si quisiéramos guardar un valor especial en una variable de sesión, como por ejemplo el código de USUARIO, y que se almacene durante toda la sesión que esté conectado el usuario.

Pero, si algún usuario lo envía por la url, el programa COBOL no sabría de donde viene, si por variable de sesión o por la URL.

Al proteger una variable, jamás podrá ser introducida por el URL, únicamente recuperarla de una sesión o bien de una COOKIE.

Podemos poner tantas variables como necesitemos, separados por comas.

Runtime COBOL - 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

Codigo Proyecto: account

Directorio COBOL: F:\APPS\ACCOUNT

Runtime COBOL:

C:\acu\runchbl.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 revizar 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, escogeremos 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 Editar Programas

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 número lleva signo, valores: "Y", "1" o "Yes"

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>
    <field type="9" int="12" dec="2" sign="Y">NUMERICO</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).
   02 NUMERICO          PIC 9(12)V99-.
. . .
```

Editor de Programas

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

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:

Archivo

Salirt – Cierra el editor de programas

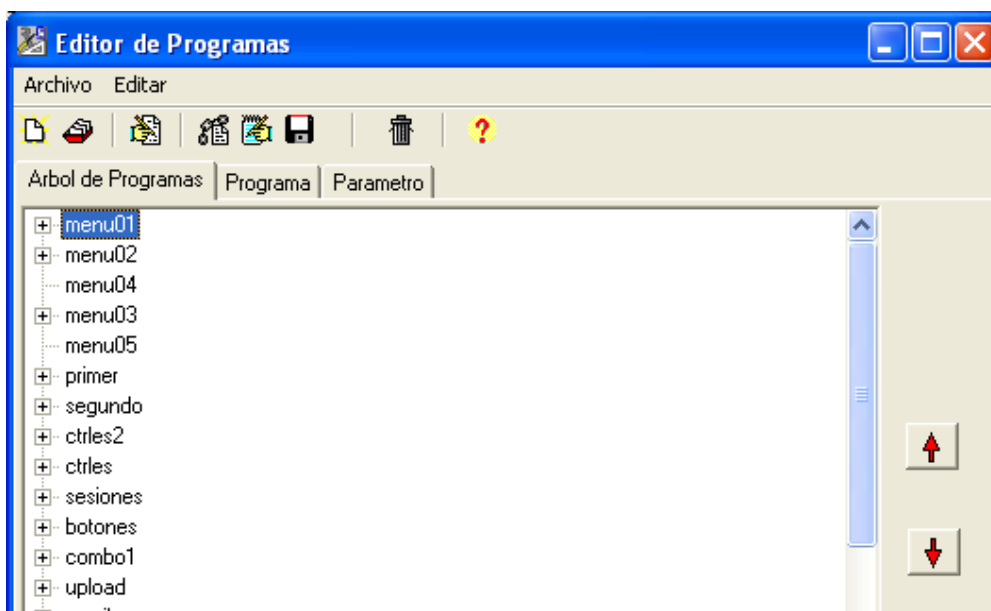
Editar

Nuevo programa – Inserta un nuevo programa en el archivo xml actual

Nuevo parametro – Inserta un nuevo campo en el programa actualmente seleccionado

Modificar – Modifica el programa o el campo actualmente seleccionado

La vista actual del editor de Programas es la siguiente:



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

Nuevo Programa – Crea un nuevo programa, igual que la opción del menú.

Nuevo Parametro – Crea un nuevo campo parámetro, igual que en menú.

Modificar Programa o Parametro – Permite modificar el elemento seleccionado actualmente.

Visualiza la Linkage Section – 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.

Editar código para Importar Campos – 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.

Guarda Configuracion – Graba el archivo xml.

Elimina programa – Elimina el programa seleccionado.

Ayuda – 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 Editar->Nuevo Programa, o hacemos clic en el botón “Nuevo Programa”.

Automáticamente cambiara hacia la carpeta Programa.

En esta carpeta solo tenemos activo el dato Nombre.

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 necesitamos.

Como modificar un programa:

Primero necesitamos seleccionar el programa que queremos modificar.

Después, hacemos clic en la opción del menú Editar->Modificar Programa.

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

Como insertar un nuevo campo parámetro:

Primero, seleccionamos el programa donde queremos insertar el campo.

Después, clic en el botón Nuevo Parametro, o en la opción del menú Editar->Nuevo Parametro

Llenamos la forma de datos para el nuevo campo.

Nombre, Tipo (X or 9), Tamaño.

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 Aceptar para agregar el nuevo campo.

Como modificar un campo parámetro:

Primero, seleccionamos el campo que se desea modificar.

Hacemos clic en el menú Editar->Modificar Parametro.

Cambiamos la información del campo en la carpeta Parametro.

Hacemos clic en el botón Aceptar.

Una vez que hemos terminado de hacer las modificaciones necesarias a nuestro archivo de proyecto xml, grabaremos el archivo con el botón “Guardar Configuración”.

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

Send Mail

En esta pestaña se puede configurar el COBHTTPD para el envío de correos, desde dentro una petición web.

O dicho de otra manera, un programa que es ejecutado con el cobhttpd, aparte de regresar contenido HTML puede disparar también el envío de un correo simultáneamente.

En los ejemplos que se pueden descargar de la pagina, hay una muestra de cómo enviar un email y regresar contenido html.

Tambien en la lista de Tips, hay una explicación de cómo funciona el ejemplo del correo.

Los datos que necesitamos configurar son los siguientes:

SMTP Host – Esta puede ser un nombre de dominio o la dirección IP del servidor de SMTP que se utilizara para el envío de nuestro email. Regularmente se registra el de nuestro proveedor de Internet, a menos de que la empresa donde vayamos a instalar este software cuente con el suyo propio.

SMTP Port – Es el puerto TCP donde se encuentra nuestro servidor SMTP escuchando las peticiones, por default podemos usar el puerto **smtp**, a menos de que nuestro servidor escuche en alguno diferente.

From Email – Aquí podemos especificar la cuenta de email desde la cual serán enviados los correos, sirve además para que las respuestas a nuestros emails sean enviados hacia esta cuenta. La cuenta FROM también puede ser especificada desde el archivo de envío del email, si esto sucede entonces no será usada la cuenta FROM de la configuración del programa si no la especificada en el archivo de email.

Authentication – Sirve para indicar si el servidor SMTP por el cual será enviado el correo necesita Autenticación, de ser así, deberemos de escoger el tipo de autenticación que utiliza, las opciones son:

- smtpAuthNone
- smtpAuthPlain
- smtpAuthLogin
- smtpAuthCramMD5

La opción que debemos de seleccionar dependerá de nuestro servidor SMTP al que nos conectaremos.

Auth Username – Es el nombre de usuario que será usado para hacer la autenticación.

Auth Password – El password del usuario con el que será autenticado en el servidor SMTP.

Save Log File – Este parámetro es nuevo, incluido en desde la version1.10, si lo marcamos, le indicamos que nos lleve un registro de los emails que se han enviado a travez de COBEMAIL, con fines de llevar un monitoreo.

Use SSL – sirve para indicar si el servidor de correos smtp, requiere el uso de seguridad SSL en las conexiones. Si es asi, hay que activar esta casilla.

Type – se refiere al tipo de SSL que utiliza el servidor. Los valores pueden ser:

None - que es el default cuando no se usa SSL,

Implicit (TLS connection)

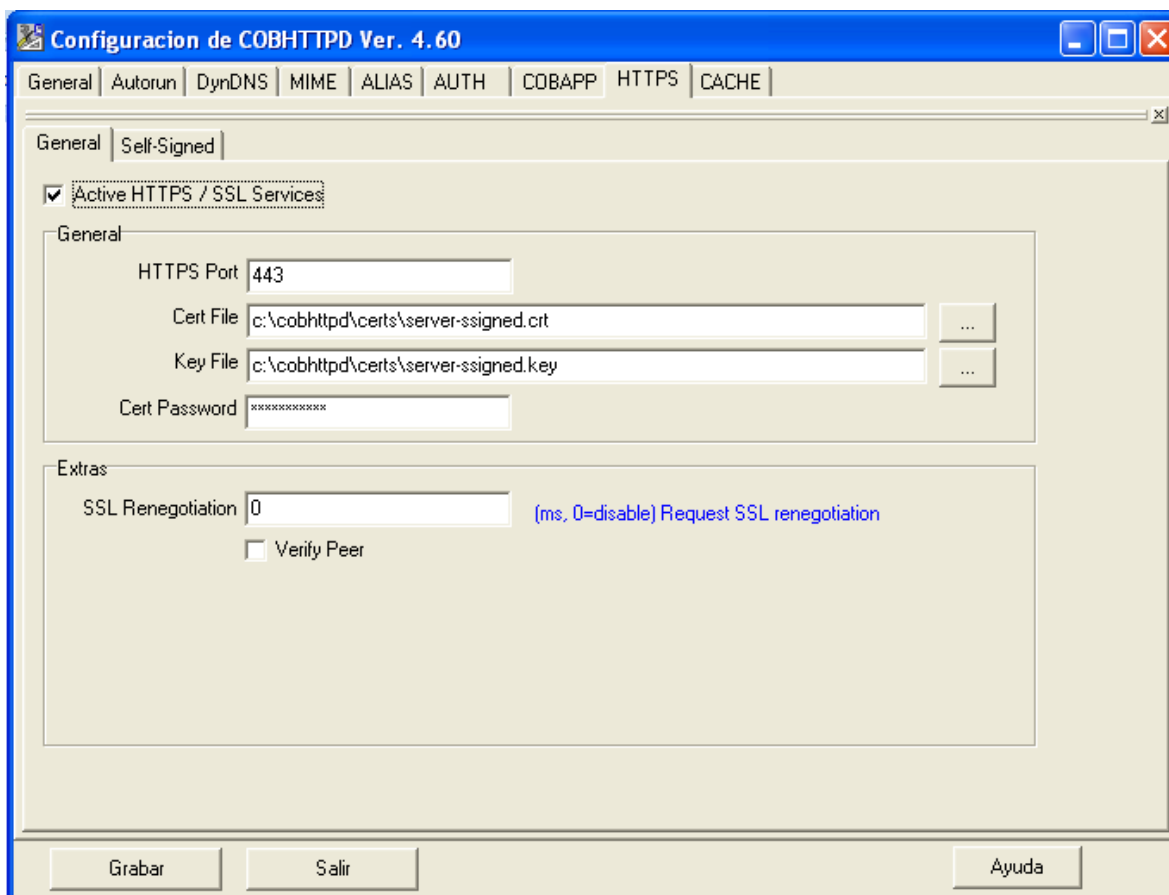
Explicit (STARTTLS)

El tipo depende de cual conexion SSL maneje el servidor, en caso de no saber, habría que probar y enviar un email con cada uno de los dos, tanto Implicit como Explicit para probar y ver si se logra hacer el envio.

HTTPS (nuevo a partir de versión 4)

En esta pestaña podemos configurar la seguridad SSL para nuestro servidor web. Con esto podemos aceptar peticiones del tipo <https://www.midominio.com>.

Estas conexiones seguras se manejan comúnmente usando el puerto 443, que es el puerto designado para conexiones http seguro.



Configuración General:

Los datos a configurar son muy sencillos:

Active HTTPS / SSL Services

Activa la configuración de los servicios http seguros. Una vez que esta marcada esta casilla, se activa toda la configuración del certificado de seguridad.

HTTPS Port

Es el puerto a utilizar para aceptar las peticiones web seguras. Por default el puerto es el 443.

Cert File

Se refiere a la ubicación en donde esta el archivo de certificado a usar para manejar las peticiones web seguras.

Generalmente tiene la extensión .crt o .cer

Este archivo de certificado normalmente va asignado a un dominio o un host en particular del tipo:

www.midominio.com

shop.midominio.com

tienda.midominio.com

Y lo pueden adquirir a través de una empresa certificadora, por ejemplo:

<https://www.ssls.com/>

<https://comodossllstore.com/>

y otras mas, pero aquí es donde pueden adquirir un certificado valido y económico.

Key File

Es la ubicación del archivo .key

Es la otra parte del certificado, y es donde se encuentra la llave privada del dominio.

Cert Password

Es la contraseña que le pusieron al certificado cuando lo crearon o cuando lo compraron en la página de su elección.

SSL Renegotiation

Se refiere a un tiempo en ms, que se puede definir para solicitar una renegociación de la sesión ssl. Por default va en 0, indicando que no es necesaria una renegociación.

Solo mover si se sabe exactamente que efectos tendrá en la transmisión de datos.

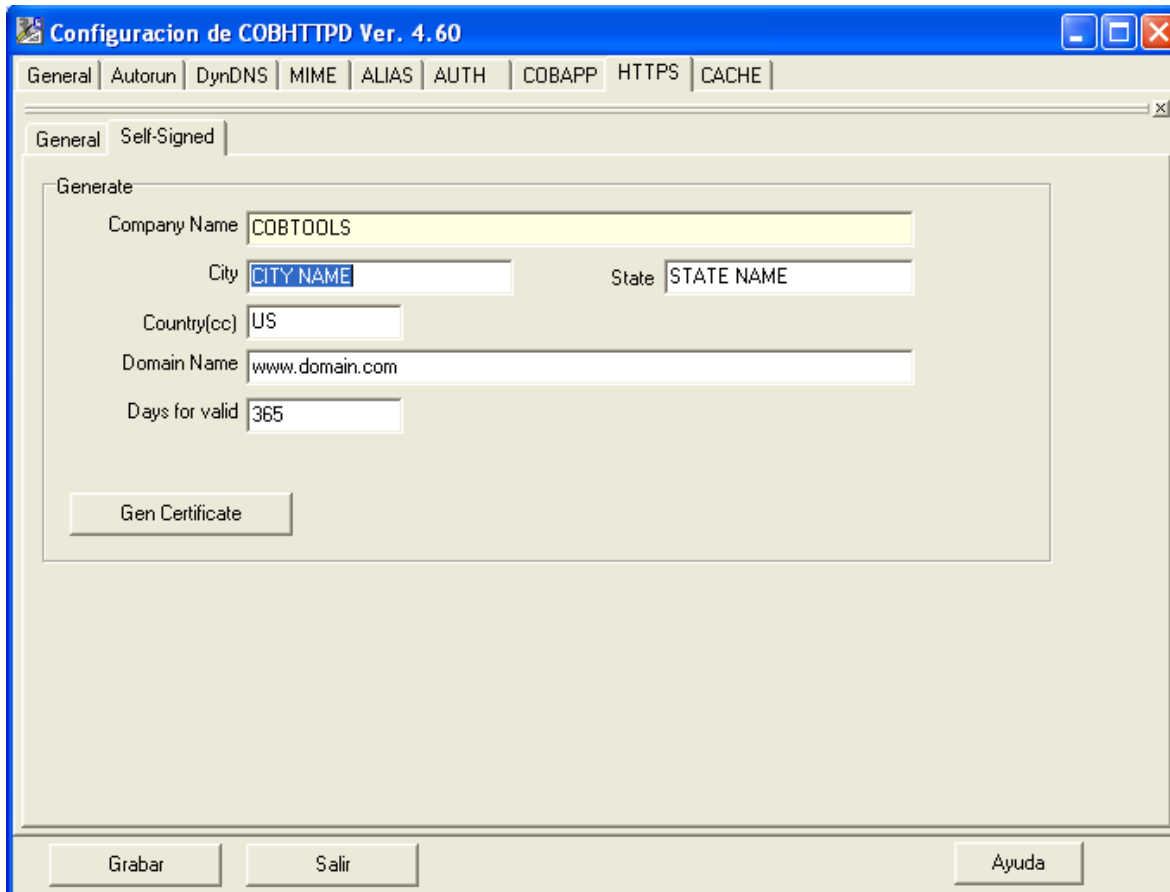
Verify Peer

Sirve para verificar que realmente la conexión que se esta estableciendo entre el cliente y el servidor sea usando el certificado configurado.

Si se marca con un certificado tipo SELF SIGNED, no podrá realizarse la conexión.

Un certificado SELF SIGNED es un certificado hecho por nosotros mismos y no usando una agencia certificadora.

Pestaña: Self Signed



En esta pestaña podremos configurar los datos para un certificado de pruebas. Es decir nosotros mismos crearemos un certificado temporal y no valido para hacer pruebas de conexión hacia nuestro puerto 443 seguro.

Los navegadores no reconocen como validos estos certificados, ya que no hay una agencia reconocida a nivel mundial que los este soportando, por esa razón van a mandar un mensaje de advertencia cuando nos conectemos a nuestra pagina usando el protocolo https.

Si el navegador lo permite, podremos crear una excepción para usar este certificado, pero en algunos casos no se crea el canal seguro usando este tipo de certificados.

Considera siempre que son solo para hacer pruebas, ya que la manera segura siempre será comprando un certificado valido en una agencia reconocida.

Los datos que solicita para crear un certificado de pruebas son:

Company name – se refiere a el nombre de compañía que llevara el certificado.

City – ciudad de la compañía

State – estado del domicilio de la compañía

Country – las siglas internacionales del país ej:

US – estados unidos

MX - mexico

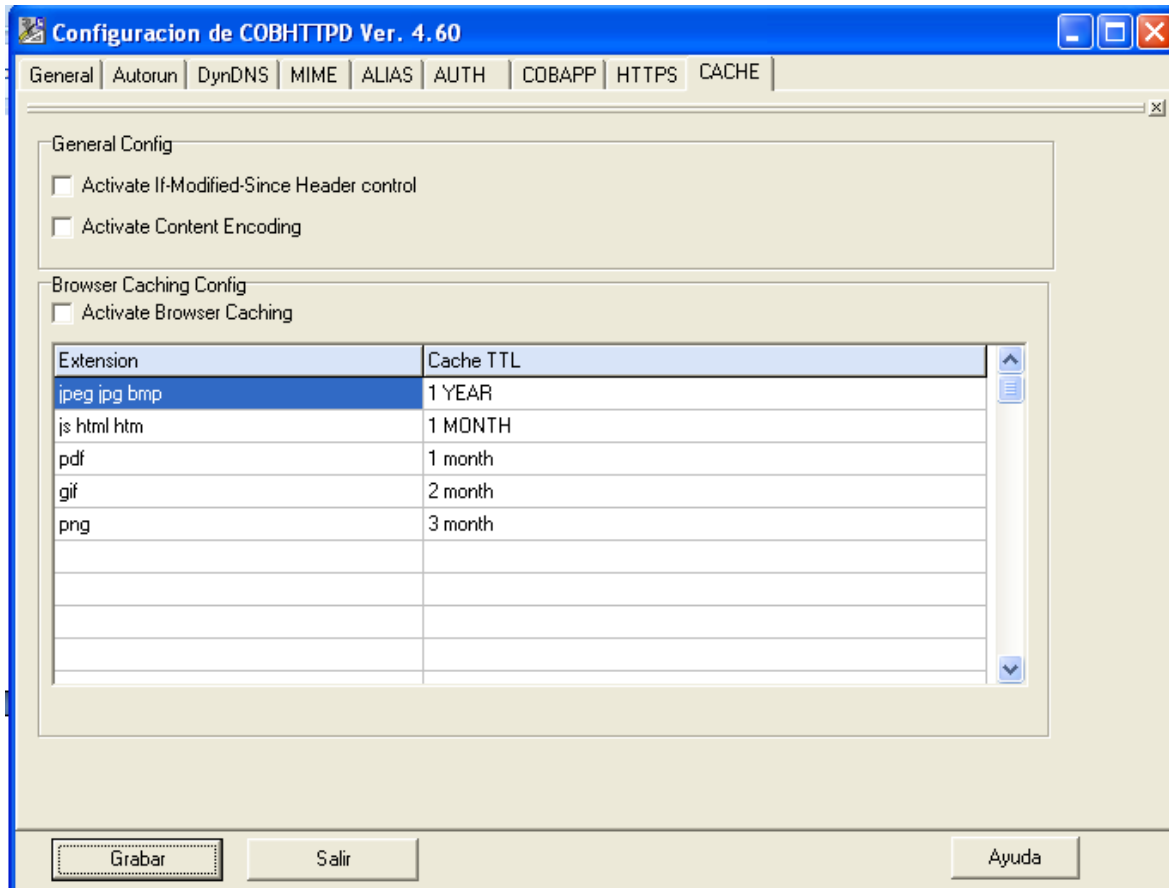
Domain Name – nombre del dominio con host, para el certificado actual.

Days for valid – numero de días en los que será válido el certificado. Un certificado aun cuando lo compramos es válido por años, regularmente 1, 2 o 3, dependiendo de la pagina donde lo adquiramos.

Una vez que le demos Gen Certificate, se generara un certificado de pruebas en el directorio C:\COBHHTTPD\CERTS

Si no existe este directorio, hay que crearlo manualmente, ya que por default ahí se crea el certificado, y es recomendable también ahí mismo meter los certificados adquiridos, solo por cuestión de tener un almacen de certificados fuera de las areas de datos públicos.

CACHE (nuevo a partir de versión 4)



Esta carpeta sirve para configurar la parte de cache, que se puede manejar entre el servidor y el cliente.

Activate If-Modified-Since Header control

Al activar esta casilla se active el uso de manejo de las fechas de los archivos físicos existentes, como los .js, .css, .jpeg, .jpg, etc.

Si un navegador ya tiene en su cache un archivo, envía en un header la fecha del archivo actual, y en el servidor, si dicha fecha sigue siendo la misma del archivo, entonces no envía el archivo completo, solo un http status como 304 not modified.

Esto hace un manejo óptimo del ancho de banda.

Activate Content Encoding

Esta casilla activada lo que provoca es que el contenido que regresa el navegador se pueda enviar empacado, ahorrando ancho de banda de transmisión de resultados.

No todo el contenido regresado por el servidor es comprimido, mucho depende de la petición del cliente, si el cliente no maneja Content Encoding, el servidor lo enviara tal cual está el archivo, sin codificar.

Activate Browser caching

Esta casilla sirve para activar el Browser Caching, que es otro tipo de cache para el cliente.

Cuando el cliente descarga un archivo de imagen por ejemplo, en los headers puede llevar por que el servidor lo envía, el tiempo que puede almacenar dicho archivo, sin volver a solicitarlo. Ahorrando ancho de banda enviando solo lo necesario.

Se debe de tener cuidado con esta configuración, sobre todo con los archivos .JS ya que si se llegase a cambiar una librería de java script, y la versión anterior la tenían configurada con un Browser caching de 1 o 2 YEAR, no podrá actualizar el archivo a menos que se fuerce la solicitud del lado del cliente, borrando el cache del navegador.

Columna extensión.

Aquí va la extensión del archivo al que se le quiere configurar un Browser caching indicando tiempo de almacenamiento.

Ej:

pdf

gif

png

jpg jpeg bmp

la extensión no lleva el punto, no es necesario.

Se pueden registrar varias extensiones que usen el mismo TTL de caching, separándolas con un espacio.

Columna Cache TTL

Aquí va el tiempo que va a tener de vida en el cache del lado del cliente.

Los valores pueden ser:

1 MIN

1 HOUR

1 DAY

1 MONTH

1 YEAR

El valor puede variar de 1 en adelante, ej:

2 MONTH

5 MONTH

Manejo de Black List de IPs sospechosas

En la nueva versión 4.62 de COBHTTPD, se incorporo un componente para bloquear intentos de ataque, y crear una lista negra de IPS.

Primero, un intento de ataque se vería en el log como algo así:

```
[20/jul/20 21:30:02:854] 10.0.0.1 6697: 404 : GET /system_api.php
[20/jul/20 21:30:05:513] 10.0.0.1 6698: 404 : GET /streaming/clients_live.php
[20/jul/20 21:30:08:177] 10.0.0.1 6699: 404 : GET /stalker_portal/c/
[20/jul/20 21:30:10:841] 10.0.0.1 6700: 404 : GET /api.php
[20/jul/20 21:30:13:501] 10.0.0.1 6701: 404 : GET /login.php
[20/jul/20 21:30:16:458] 10.0.0.1 6702: 404 : GET /streaming
```

Aquí, lo extraño a revisar es que, en nuestro servidor COBHTPD, no existen archivos con extensión .PHP, con esto determinamos que es un extraño el que está intentando buscar algún hoyo de seguridad en nuestro server.

Y aunque, realmente, al no existir el lenguaje PHP en nuestro server, y no ejecutar ningún programa inseguro, la verdad es que si toma tiempo del servidor y ancho de banda en regresar el error 404 de no encontrado.

Por esta razón se hizo el control de lista negra, para bloquear estos intentos de ataque, con peticiones desconocidas.

Configuración de la lista negra:

HACKFILTERLIST.TXT

Podemos determinar entonces, que un intento de ataque, es cuando en la petición solicitan un archivo que pertenece a otro lenguaje que no es COBOL, como ejemplo .php, .pl, o .cgi. Estas extensiones o partes del nombre, se pueden configurar en el archivo HACKFILTERLIST.TXT

Ejemplo:

path=.php	*** remover si se va a usar soporte para PHP, etc
path=wp-login.php	*** aqui no usamos WordPress!!
path=.pl	*** no usamos Perl

path= sirve para especificar parte de la ruta, y si cumple parte con el valor de path=, se bloquea el acceso y también se bloquea la ip, para que cualquier petición de la misma ip se bloquee.

Modo de bloqueo:

El bloqueo consiste en regresar un error, pero retardando la respuesta en 60 segundos, esto provoca que el cliente remoto tenga que esperar por la respuesta.

El bloqueo es por cierto tiempo, por default son 4 horas.

Esto quiere decir, que cualquier petición que venga de esta ip, sera bloqueada durante 4 horas, después de 4 horas se limpia la lista de IP, y puede volver a intentar accesár, pero si sigue tratando de hacer peticiones invalidas, será bloqueada nuevamente.

remaddr=10.0.0.1, con esta variable se especifica una ip manualmente, que automáticamente va a quedar bloqueada y en la primer petición se procede a regresar el error, retardando la respuesta en 60 segundos.

Esta ip, al ser puesta manualmente por configuración directa al archivo, jamás será limpiada en 4 horas ya que queda fija en la lista negra.

Modo de bloqueo 2:

Otra forma de bloquear una IP, es cuando un atacante remoto, intenta hacer demasiadas peticiones en poco tiempo, por default son marcadas las IPS como negras, cuando hacen más de 1,000 peticiones en 15 minutos.

Si una ip intenta detectar una vulnerabilidad, hace demasiados accesos tratando de descubrir cómo hacer daño, pero cuando COBHHTTPD detecta que hizo más de 1,000 accesos en 15 minutos, automáticamente la IP queda bloqueada, y se manda a el control de la lista negra, bloqueando la IP por 4 horas, denegando el acceso al server.

Comentarios:

si se desea indicar un comentario en el archivo, se puede usar un asterisco, en la columna 1, ejemplo.

* Este es un comentario

También se puede poner un comentario en la misma línea de la configuración, usando 3 asteriscos, ejemplo:

path=.php *** este es un comentario, para indicar algo sobre esta configuración.

WHITEIPLIST.TXT

En este archivo podemos agregar una lista de IPS fijas, que deseamos que no se bloqueen nunca por el BLACK LIST.

Ejemplo:

172.16.

Pone en lista blanca, toda la red 172.16.0.0, no importando de que ip vengan las peticiones.

Comentarios:

Podemos poner comentarios en el archivo, usando un asterisco, en la columna 1, ejemplo:

* Este es un comentario

COBHTTPD.DEF

Este archivo contiene la configuración avanzada del control de BLACK LIST.

Básicamente son 3 valores los que podemos alterar:

BlockAttempts = 1000

Esta variable, indica la cantidad de intentos que puede hacer una IP, antes de considerarse como un riesgo.

Por default están los 1,000 accesos permitidos, pero si tenemos clientes habituales que acceden mas de esta cantidad podemos abrirla sin problema a 15,000 o incluso 20,000.

BlockAfterMins = 15

Esta variable, es el tiempo límite, en el que un cliente puede hacer menos de los accesos configurados en la variable anterior. Por default son 15 minutos.

Es decir, controla el tiempo en el que un cliente puede acceder las mil ocasiones al servidor. También, puede modificarse este tiempo, es en minutos, y puede cambiarse a 5, 10 o lo que se necesite.

BlockForMins = 240

Este es el periodo en el que se bloquea la IP que acceso más veces de las permitidas en el tiempo límite. Por default son 240 minutos o 4 horas.

Como desactivar la protección Black List

Cuando se usa un modem ADSL como router hacia nuestro servidor, puede ser que todas las peticiones lleguen con una misma IP (La del Modem).

Esto hace, que todos los accesos cuenten como si vinieran del mismo cliente.

Y si existe un intento de hackeo o ataque, se bloquee la única IP que accesa a el servidor, que es la IP del modem ADSL.

Por esta razón, es que ahora se puede desactivar la protección de ataques/hackeo.

Para esto, dentro del archivo COBHTTPD.DEF, ahora se puso una variable que se llama:

EnableProtect = true

Para desactivar toda la protección de intento de ataques y lista negra, solo hay que poner esta variable como false.

Y con esto, ya se podrá usar el acceso al servidor sin restricciones.

Hay que aclarar que esto deja sin ningún tipo de validación de accesos al server desde fuera.

Lo mejor sería configurar correctamente el modem, para poder obtener la IP remota real, y que no se enmascare con la IP del modem.

Ejecutando el programa Server COBHTTTPD

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(64000) .
```

Estos dos registros se encuentran ya definidos en el archivo **LS-HTTP.FDT** que se encuentra en el directorio COBOL dentro de COBHTTPD, 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 COBHTTPD hacia nuestro programa COBOL que será ejecutado:

HTTP-PROJECT – es el código del proyecto al que pertenece nuestro programa que se esta 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 esta ejecutando, y nos puede servir para hacer librerías externas y hacer referencia hacia el programa que se esta 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 COBHTTPD, cualquier numero 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 64,000 bytes de tamaño. En este campo nos llegaran los diferentes valores que nuestro programa COBOL necesita para funcionar.

Al ser de un máximo de 64,000 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.

COBHTTPD 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 esta 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 guardo 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 COBHTTPD.

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
- 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 COBHTTPD, ejemplo C:\COBHTTPD\CGILIB.