

COBHTTPD

The COBOL Application Server

Users Manual

(Ver. 2.30)

INDEX

Description	3
How work this program.....	4
COBHTTPD Programs and Files	5
COBHTTPD Directories.	6
Installation Directory	6
Documents Directory.....	6
Reports Directory	7
Temprary Directory	7
XML Directory	7
Configuration.....	9
Configuration (Projects)	13
COBOL Compilers suported	16
ACUCOBOL, interphase httpacu.cbx	16
RMCOBOL, interphase httpm5.cob	16
MSCOBOL 5, interphase httpms5.exe	16
POWER COBOL 3, interphase httpwc3.exe.....	17
POWER COBOL 85, interphase httppw85.exe	17
Edit XML button	19
XML interface file format:	20
XML file Interface editor	22
Runing the the COBHTTPD server program	25
Running the COBOL programs.....	26
HTML content	26
PDF content	26
Direct access:	26
Calling since HTML files	26
Data input received in our COBOL program	28
HTTP-CONTROL	28
HTTP-DATA-INP	29
How generate the output to send to client browser	30
Work with sessions.....	32
Including external files	33
CGILIB Introduction	34

Description.

COBHTTPD is a Application Software for COBOL. With this proram you can run COBOL programs in remote way.

Contains too a Report Server, for view reports in Internet.

This Report Server work with a special version of COBVIEW for convert the reports to PDF and send to COBHTTPD.

COBHTTPD work too like http server, because use the http protocol to receive and send the data from client browser and to client browser.

COBHTTPD is not a gate program to run COBOL programs, COBHTTPD is a complete software, not need external programs to full function, and not need any DLL or external library to work.

COBHTTPD not need special installations for work, because work in Windows 98/2000/XP, and the installation is manual, because you copy the program where you want, and the directory is defined by you. From this way, you always know where put the files the program.

COBHTTPD can send HTML content and PDF reports.

How work this program.

First a little sample:

- 1) The user visit the page for view a report, and receive the html form with the data need for build the report.
- 2) The user fill the form, and post this form to the server.
- 3) The server run the apropiate COBOL program.
Convert the file to PDF.
- 4) The server return the PDF file to the user and he view in your IE explorer or Netscape.

COBHTTPD always return the reports in PDF format, reasons:

- PDF is universal
- Exist the pdf plugin for your preferal browser.
- Not need any special program for view reports.
- Can be printed in any printer.
- Can be viewed in several plataforms: Windows/Linux/Unix

COBHTTPD help you to convert your COBOL programs to internet, because all parameters that your program need are received by the LIKAGE-SECTION.

Because the http protocol can work with data, but this data is not formated, not contains any information type, and is not delimited by size, COBHTTPD need know the programs that can be published to internet, and need know the input data that every program need to work.

So, before that any program can be used, need be defined int COBHTTPD, and need be defined the input data for this program, all can be do with the Cobhttpd-configurator.

COBHTTPD Programs and Files

The programs that are part of the cobhttpd suite are:

CFGCOBHTTPD.EXE – this program is the Cobhttpd-Configurator. With this program can modify all information for cobhttpd can work. Can modify the TCP/IP port listen for server, directories for work, projects, programs, field input.

COBHTTPD.EXE – This is the program that need be run for serve the client petitions. When this program run, the server is ready for work.

COBHTTPD_SVC.EXE – This program is the special version for Windows 2000/XP that run as service. A service not need that your windows installation LOGON to run the program, because the services run since your PC is turned on, and not need LOGON to your windows session.

COBHTTPD_INT.EXE – This program is the interface between the COBHTTPD when work like service, because a service not have user intervention, this program help us for monitor the access that are done y that programs are executing, special for beta test and develop time.

COBVIEWHTTPD.EXE – This program convert the report generated by your COBOL programs to PDF and return to server so this can return to the Browser client.

COBHTTPD.DAT – This file contains the more important information of configuration for the cobhttpd program. Contains the general information of the program, TCP/IP port, directories configuration, and Projects information.

All this program need exist in the same directory, by example:

C:\COBHTTPD

But you can put your COBHTTPD installation where you want. For COBHTTPD is not important where you install.

COBHTTPD Directories.

The COBHTTPD program need the following directories for work.

Installation Directory

This directory contain all programs and the cobhttpd configuration file.
By default and in this document we call C:\COBHTTPD.

Documents Directory

This directory contain all files published in the web server. The files published can be: html files, gif files, jpg files, and all files that need for build your special html forms.

This directory can be viewed from you browser, when you visit the site, sample:

<http://www.yourdomain.com>

If your TCP/IP port is 80, then with this address you can view the files that exist in the Documents Directory.

The first file viewed is the **Default document** registered in the Cobhttpd-Configurator, by default we call to this file index.html.

If you have a sub-directory “website” inside the Documents Dir, and the Documents dir is:

“c:\cobhttpd\httpdoc”, then you need put your special files in the directory:

“c:\cobhttpd\httpdoc\website”, and you can visit this files in the following address:

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

By default, and for best practice, this directory need be inside of the **Installation Directory**.

So, if you have the Installation Directory to D:\COBHTTPD, put your Documents directory to D:\COBHTTPD\HTTPDOC. So, if you need copy the full installation only backup the D:\COBHTTPD, and all files can be restored later.

Reports Directory

This directory is for store the PDF reports that need be viewed by any browser. Because this directory need be accessed since Internet, this directory need exist inside the **Documents Directory**.

By default we call to this directory “reports”, so, if we have the **Documents Directory**: “D:\COBHTTPD\HTTPDOC”, and for our **Reports Directory**: “reports”, the full path for this directory is: D:\COBHTTPD\HTTPDOC\REPORTS

All reports can be accessed from internet from the form:

<http://www.domain.com/reports/reportname.pdf>

This directory is depured by cobhttpd, any report generated and published int this directory have only 5 hours of life before cobhttpd erase this file.

Temprary Directory

This directory is need for create files need for the proper operation of the program. This directory contains litle files that are created by cobhttpd and accessed by your COBOL programs for data conversion, and your text file reports before that cobhttpd convert this to pdf public files.

By default and good practice we use: C:\TEMP or D:\TEMP for this work, is better a directory with 8 letter for old COBOL compilers.

This directory is depured by cobhttpd, equal that Reports Directory.

XML Directory

This directory contain the XML definition for the programs that are published in the internet. Remember that every program need be defined before that this can work in internet. Every project contain a file with alls programs, this file contains too the fields definition for the program.

By default, the directory need exists inside the **Installation Directory**, and need be called “xml”.

The xml files are maintained automatically by the **cobhttpd-configurator** Program.

Installation

This program not contain some program installation, this program was made for “system programmers” or “software developers”. And like programmer you need know wich changes do the program in your OS installation. And if you need copy the installation directory, only copy the directories in the same location in another computer and run the cobhttpd program to begin work.

For the fast installation follow the next instructions:

- 1) - Unpack the cobhttpd.zip file in your preferal directory
“c:\cobhttpd” is a good choice. With this installation you obtain the next directories:
 - httpdoc – Directory for all document html files and images
 - httpdoc/reports – Directory for all reports generated in automatic way by cobhttpd.
 - Xml – Directory for all project and programs definition. One file for every project.
 - Cobol – The directory with the COBOL programs interface between COBHTTPD and your COBOL programs.
- 2) – Define your temporary directory like c:\temp.
- 3) – It’s all for installation, next do the correct configuration in the next section.

Configuration

For configure you need run the program cfcobhttpd, all configuration is do with this program.

This program not need any parameter, just run thus program.

First begin with the data need for the program, the data are:

Web server port: This data define the correct TCP/IP port for listen server. If you now have any web server running in your computer, then you need define a TCP/IP port diferent for 80. The port 80 is the default port for http servers, you can define any port that is not used by any program.

If you define the port 8080, then you access your webserver from the next sample:

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

Default document: The default document is the first file that the web server load when you not indicate any file to load since your browser.

Default: index.html

Sample: if you visit the next address:

<http://www.yourdomain.com/dirsample>

the cobhttpd first find the file index.html, if this file exists in this directory then load the file and cobhttpd return this to the browser in the client side.

Document Dir: This directory is the default allocation for your html files and reports auto-generated.

Default: c:\cobhttpd\httpdoc

Note: you can chagen your Document dir to another diferent, sample: c:\httpdoc, but remember that when you copy the current installation to another computer need copy too all directories for the proper work.

TEMP Directory: Define the directory for the temporary files.

Default: c:\temp

This directory is not created by the configuration program, you need create this manually.

File type: In this moment, the cobhttpd work with 2 file types for return the reports to browsers.

- 1) PDF files, the best choice, because is universal and can be viewed in any computer with any operating system, only need the correct PDF plugin for your browser.
- 2) HTML/JPEG report – With this option can generate reports with an internal html viewer, and every page is a JPEG image, and every next or previous page is generated online. Special if you not want use the PDF plugin in your client browsers, bad if you need print reports in printers in the client side, only can access one page by time.

Show warnings in HTML mode – Show error messages in the browser if any error occurred, errors like XML file not founded, Directory project not found, program not found. If this data is not checked the error is not showed, and the program execution is stoped.

Save Activiy LOG – If check this option cobhttpd generate a log file with the events occurred like get file, run the cobol program, etc. The file is named: cobhttpd.log in the Installation Directory “c:\cobhttpd\cobhttpd.log”.

Autorun on windows start – Check this option if you want that you cobhttpd program start when windows start. This function need that the user start her session, this because all programs autorun only start when the user logon to windows.

Put in systray – This option allow that when autorun the program with the prior option, the program run minimized and allocated in the systray.

Run as NT Service – This option appear only if the computer where is installed the cobhttpd program have Windows 2000 or XP.

If you check this option then the cobhttpd program run as NT service. The cobhttpd server run when the Windows start, even if the user as not start session. When this options is checked the options: “Autorun on windows start” and “put in systray” are disables, and are enabled all the data in the block “Run as Service”.

Run as service block

Install button – Install the cobhttpd as service in this computer. Only can be actived once. For click this button need input the data username and password.

Uninstall button – With this button can uninstall a cobhttpd service, when this was installed before.

Start button – Start the service when this are installed and is not runing in this moment.

Stop button – Stop the button when this are installed and is runing in this moment.

Username – is the user for run the service. The user must exist in the current system. Is in the form: .username

Password – The password for the username.

Remember your username and password, because when you save your cobhttpd configuration never show again for security reasons. And this data is not saved in the cobhttpd.dat configuration file.

Projects button – This button allow access to projects configuration screen, explained in the next section.

Cookies Block

COBHTTPD need work with cookies for can maintain permanent variables by session. A session store a small data in the Client browser, and the Client browser need have the option for Accept Cookies, otherwise the permanent variables can't be reused, and always have a initial value of zero or space.

Expire Cookies (mm) –

This data indicate the time life that a cookie have, the time that is configured here is in minutes, is very important define a life time adapted. By sample, if an installation have a life time for 30 minutes, when a person access an application with her password, this session it would be opened for 30 minutes, is necessary also to place an option for close the session manually and assure to us that the users select this options for close the session and not wait the 30 minutes for the COBHTTPD close automatically.

Not Autoexpire cookies –

If mark this option, COBHTTPD automatically renew the life time for the session while the user use the web application options. By sample, if we put 10 minutes for life time for cookies, while the user are using the application, the 10 minutes are renewed every time. If this option is not marked, then the session only work by 10 minutes, and when the life time is over, the user need LOGIN again in the application.

Domain –

This data actually is not used, the idea is can define a domain name for identify the cookies in the Browser, but actually not need have value. Is only for a future use.

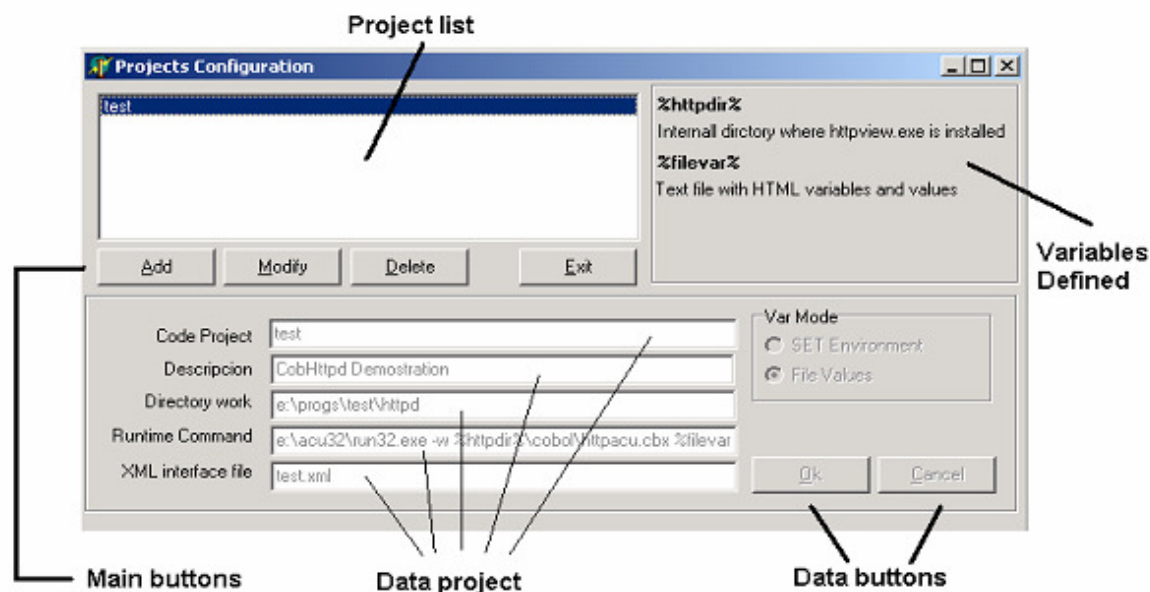
Configuration (Projects)

Projects are group of programs that need be accesed since the internet. A project contain data for the COBOL programs that are members for the same COBOL application.

To indicate that project use in the url call:

`http://www.yourdomain.com/cobpdf?proid=PROJECT`

The cobpdf function in the call for this url is explained later, the more important now is the way how indicate the project ID. This ID is unique and global, always we call our project with this ID, and is universal and unmodified because in all our calls to any program member of this project only can do with the ID. It's like your application name, always is the same and never change. When you begin to use this ID in your calls in the html files can be dificult change it later to all calls.



The project configuration screen contain the following components:

Project list

is the list with all project defined. When click in any project in the list all Data Project are showed, but is not possible modify now, because only is selected. For do any action in any project need select the appropriate button.

- Add – Allow insert a new project in our project list. When select this button the data project are enabled and put in spaces.
- Modify – Allow modify the current project select in the project list. When is selected the data are enabled for modify the data project.
- Delete – Allow erase the current project selected in the project list. When is selected the project is deleted and disappear from the project list.
- Exit – Close this window and return to main configuration screen.

Data project:

Code Project

Is the Project ID. Must be unique and can't be changed later.

Description

A short description for project.

Directory work

Is the directory where the COBOL programs reside. Always that your COBOL program is running the current directory is changed to this before to run the program.

Runtime command

Is the program to run your COBOL programs. If your COBOL compiler generate auto-executable programs is not problem, we can do a little skip to this.

Normally the runtime need pass any special parameter for the runtime and the COBOL program to run.

Exists two vars that can be used for special things in this data: %httpdir% and %filevar%
%httpdir% can tell us where is the directory for the cobhttpd installation. Of course you can do by example with "c:\cobhttpd" to refer to directory installation, but if you change the directory installation in any computer later, you can modify too the project definition. This problem is resolved with this variable.

%filevar% is the data file that contain the data input for the COBOL program. We can't know the value for this file name in the runtime, but we can indicate the position for this value in this variable.

For every COBOL compiler we have a little program for use like gate between our cobhttpd and your COBOL program. This program is different for every compiler.

Samples:

- for run a project in the directory F:\APPS\ACCOUNT
- Is developed in Acucobol, the runtime is RUNCBL.EXE and this file is allocated in C:\ACU
- Project ID: account

Code project: account

Directory work: F:\APPS\ACCOUNT

Runtime Command:

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

If we have instaled cobhttpd in c:\cobhttpd

Is tha same if we put:

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

But remember, not always the directory can be c:\cobhttpd

If in any other installation we put cobhttpd in d:\cobhttpd, then we need change our projects the directory for cobhttpd.

The subdirectory cobol is where by default we put the Internal cobol programs for every compiler.

We develop a program for every compiler.

In this case the data: %filevar% the cobol program receive this like parameter, and make with this any things.

Sample 2:

The same sample but using RMCOBOL in C:\RM

Runtime command:

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

COBOL Compilers supported

Next will be detailed the different programs that are used with every compiler supported by COBHTTPD, in addition we put an example of how it would be left the line of **Runtime Command**.

ACUCOBOL, interphase httpacu.cbx

The interphase program to use is httpacu.cbx, and was created using the acucobol ver. 3.2, it program need the variable %filevar%

Sample of "Runtime Command":

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

Can use too the -b parameter, for indicate that we want execute the program in background, if acucobol mark any error, this is send direct to COBHTTPD console, for view in the server side.

Too can add the -e errorfile parameter, with this parameter can create a error file for view like log file. If the cobol program abort in any way, you can view the file and locate the error registered.

RMCOBOL, interphase httprm5.cob

This interphase program is for RMCOBOL version 5.36.

Sample of "Runtime Command":

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

In this compiler the %filevar% variable is passed like parameter.

MSCOBOL 5, interphase httpms5.exe

This compiler not need runtime program for execute the COBOL programs. This because the interphase program is auto-executable, and it is execute directly.

Sample of "Runtime Command":

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


POWER COBOL 3, interphase httpwc3.exe

This compiler not need runtime for execute the COBOL programs. This because the program is autoexecutable, the program interphase is executed automatically.

Simple of "Runtime Command":

```
%httpdir%\cobol\httpwc3.exe %filevar%
```

POWER COBOL 85, interphase httpw85.exe

Equal that the previous compiler, not need runtime, this program are compiled in Power COBOL 3, using the COBOL 85.

Simple of "Runtime Command":

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

This two interphases are created using POWER COBL 3, you need build your COBHTTPD programs for use with this interphases like DLL modules in Power COBOL 3.

Runtime Type

This block is for define the Runtime type that we have for run our programs, the options are:

MSDOS
WIN 16
WIN 32

Is very important select the proper option, COBHTTPD execute every runtime program in different way, and when selected a uncorrected option can crash COBHTTPD.

If our runtime is for MSDOS and work in TEXT mode, selected MSDOS.
In change if the runtime is for Windows 3.11, we selected WIN 16.
And, is the runtime is for Windows 98 or superior, we choose WIN 32.

Process Show

This block is for specify to COBHTPD how execute the runtime, in some times when execute the runtime, this program automatically show a window that is opened when the program Start. If we need do that this windows is not showed we selected the option HIDE, some runtimes need a special configuration for indicate that no show this initial Window, like RMCOBOL by sample.

On other hand, if we desired display some data type for can debug our program when we do tests, we need choose the SHOW option, only for the moment when we debug the program, and we can see the DISPLAYS that we do in our program.

It's necessary clarify that when a Server is working in production, we need choose always HIDE.

Edit XML button

With this button you can edit the XML file with the xml editor, that has cobhttpd.

When you click this button appear the screen for xml file maintainance, where you can:

- insert new programs
- modify programs
- add fields input to any program
- delete programs

The another way to modify your XML file, is editing with any text editor.

In the next two sections, I explain first how is the internal format for the xml file, later I explain in detail the full functions for the XML file editor.

XML interface file format:

The xml interface file need have first, the httpview root element.

Example:

```
<httpview>
  programs definitions
</httpview>
```

Programs definition:

Every program need have the element <program> with the following parameters:

Name="program-name"

Example:

```
<program name="prog01">
  fields definitions
</program>
```

Field parameters for program.

The field definition is with the tag <field>, format:

Parameters:

Type – field type, 9-numeric, X-alphanumeric

Size – field size

Int – Integers size

Dec – Decimals size

Sign – Indicate if the field need sign

The value in this element is the field name in the url or form data.

Example:

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

Full Example:

```
<httpview>
  <program name="prog01">
    <field type="X" size="10">name</field>
    <field type="X" size="20">lastname</field>
  </program>
  <program name="prog02">
    <field type="9" int="3">startpage</field>
    <field type="9" int="3">endpage</field>
  </program>
</httpview>
```

prog01 cobol source interface:

```
. . .
01 DATA-INPUT.
   02 INPUT-NAME      PIC X(10).
   02 INPUT-LASTNAME  PIC X(10).
. . .
```

Prog02 cobol source interface:

```
. . .
01 DATA-INPUT.
   02 STARTPAGE      PIC 9(03).
   02 ENDPAGE        PIC 9(03).
. . .
```

XML file Interface editor

This editor is part of cobhttpd, and can be accessed since the “**Edit xml**” button.

When you click this button, the xml file for the current project is opened, and loaded in the xml file editor.

The file is unfolded in Tree mode.

The xml file editor contain the follow menu options:

File

New - Create a new xml file, clear the current content

Open – Open a xml file if this exists

Save – Save the current content in the file text

Exit – Close the xml file

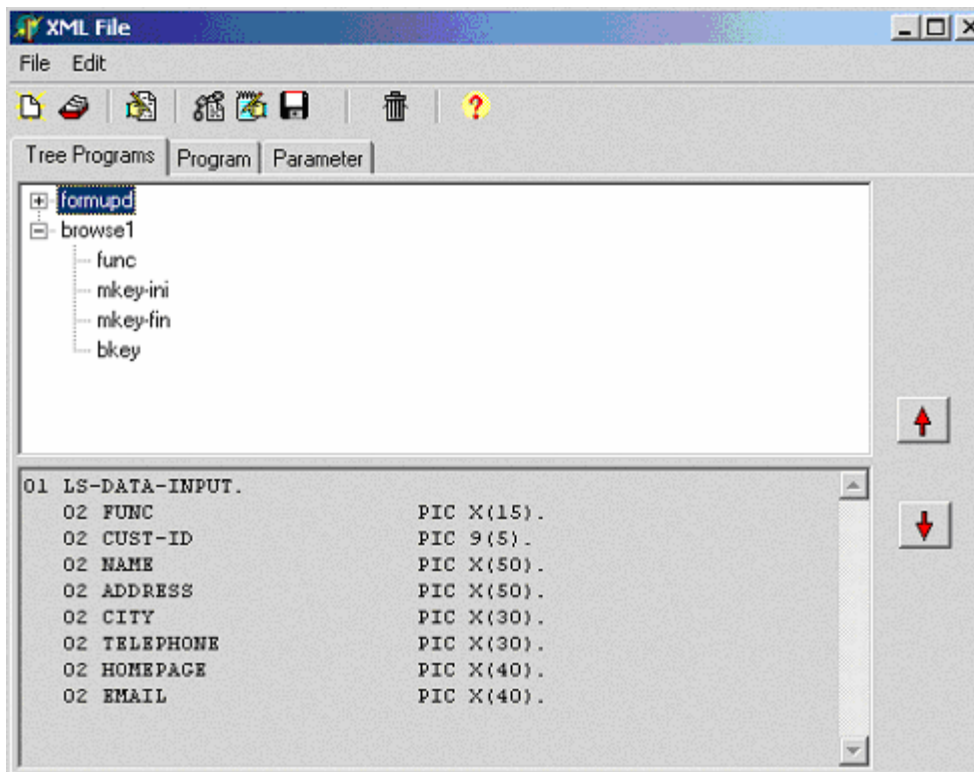
Edit

New program – Insert a new program in the current xml file content

New parameter – Insert a new Field parameter for the program current selected

Modify – Modify the program or field current selected

Current view with a xml file loaded in the editor:



As can you see, also have a buttons for any special operations:

New Program – Create a new program, equal to menu option Edit->New Program

New Parameter – Create a new parameter, equal to menu option Edit->New parameter

Modify program o parameter – Can modify the selected element (program o field).

View Linkage – With this button, we can see a LINKAGE-SECTION structure for the field definition for our program current selected.

Edit code for import fields – Allow edit the fields record for input in the program, using the COBOL syntax for easy fields edition, too is possible copy the record content directly over the COBOL source program.

Save file – Save the xml file.

Delete program – Delete the selected program.

Help – Show a little window help.

Up and Down arrows – With this change the current field position, for up or down, depending on the position desired for the current field.

When you select another program, is showed the fields record in the bottom section.

How add a new program:

For this operation, select the menu option Edit->New program, or click the button New program.

Automatically, the current tab change to Program.

In this tab only we be able to type the Program name.

After type the program name, click on the OK button to add the program to the tree xml programs definition.

Next to insert the new program, select and insert the fields for this new program.

How modify any program:

First you need select the program that you want modify.

Next, click in the menu option Edit->Modify program

The data that you can modify only is the program name.

How insert a new Field parameter:

First, select the program where you want insert the new field.

Later, click on the button New parameter, or in the option menu Edit->New parameter.

Fill the data for the new field:

Name, Type (X or 9), Size.

If your new field was defined like Numeric (9), then you need define the Integers size field and the Decimals size field.

If your numeric field need sign, check the data: Number with sign

Click OK button to add the new field.

How modify any Field parameter:

First, select the field that you want modify.

Click on the option menu: Edit->Modify parameter

Change the field information, in the Tab Parameter.

Click OK.

To Exit and Save the XML file definition

Select the option menu: File -> Save

And then, File -> Exit

Runing the the COBHTTPD server program

For run the cobhttpd program, and to listen request of the client browsers, run the following program.

Cobhttpd.exe, if your windows is windows 98, or NT/200/XP and you now want that your program run as Windows Service.

Cobhttpd_svc.exe – run automatically if your windows is NT/2000/XP and you configure to run as service. This program run autmatically when your windows start.

In Windows 98, you can configure your server to run minimized in the systray.

To open the main window program in this mode, you do double-click in the icon, and the main window is opened.

Running the COBOL programs

COBHTTPD have two ways to execute a COBOL program, the way for execute the program will depend of the content type to return.

HTML content

If we want return html content invoke the program in the next format:

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

PDF content

If we want return a PDF report, invoke the program in the next format:

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

Direct access:

Since your address field, in the web browser, you can run a program for test for the next way:

`http://www.yourdomain.com/cobpdf?proid=PROJ&prog=PROGNAME&F1=value`

With this data, you can run the program PROGNAME in the project PROJ.

And receive the field F1 for input parameter with: value

Remember that, if you define a diferent port that 80, then you need indicate this port for your server, sample if you register the port 85:

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

Calling since HTML files

For call since html forms, just put the address in your POST ACTION data.

```
<form method="POST" action="/cobpdf?proid=PROJ&prog=PROGNAME">  
  ... form fields definition  
</form>
```

for html novices, if you look in the action, you can see that I not write the full url address. This is not necessary, because the client browser expand this url with the current webserver address and the current TCP/IP port web server.

But, if you have the html form file in a different web server, url and ip port, and you only want run your COBOL program with the cobhttpd server, then your post can be:

```
<form method="POST"  
action="http://www.yourdomain.com/cobpdf?proid=PROJ&prog=PROGNAME">  
  ... form fields definition  
</form>
```

If you can to pass a data, but this data is not have input field in the form, you can pass this in the url address:

```
<form method="POST" action="http://www.yourdomain.com/cobpdf?  
  proid=PROJ&prog=PROGNAME&func=add">
```

In this sample we add a **func** field.

This field is passed to the program PROGNAME, and this program need be defined with this field (func) in the field definition too.

When our COBOL program run, func field also it arrives, equal that the data in the INPUT htm form fields.

Data input received in our COBOL program

From this moment, our COBOL program can receive 2 data records:

```
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) .
```

This two records will be find defined in the file **LS-HTTP.FDT** that exist in the COBOL directory, inside COBHTTPD, also it can exist in every sample because this file is need for compile the programs.

HTTP-CONTROL

This record: HTTP-CONTROL, contain some important data for our program, and are passed from COBHTTPD to our COBOL program that will executed.

HTTP-PROJECT – Is the code project that our program belongs. It value can serve to us inside our cobol program for do url links to another program inside the same project.

HTTP-PROGRAM – Is the COBOL program that be execured, and can serve to us for do external libraries and do references for the diferent program, and not have to write the program in fixed way or do multiple paragraphs for every program.

HTTP-OUTPUT – Is the filename for return the html content o pdf content that we want.

HTTP-ERROR – Is a error code that we can return to COBHTTPD, any number different of 0 will be error. (not used by the moment).

HTTP-ERROR-MSG – Is an occurs of 5 lines when we can put a error message and return to COBHTTPD. This message work with HTTP-ERROR. (not used by the moment).

HTTP-DATA-INP

This second record, contains only one field of 4 kb size. In this field arrived to us the different values that our program need to work.

Being of 4K max size, we can even receive great amounts of data to conduct operations of update files.

In our program we can divide this data declaring a record in the WORKING-STORAGE, and only move this record to our internal record for obtain the division real, sample:

```
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.
```

In this way we use the fields of REG-INTERNO in our program.

How generate the output to send to client browser

Or generating HTML content or generating a PDF report we must record the content in a text file.

COBHTTPD will indicate to us how name this file that we need generate. This name file arrive to our program in the HTTP-OUTPUT variable. This variable contain the name for the file. If we not named the file with the same name, COBHTTPD never send the content to the client browser.

Now, since COBOL we can use the next declaration for create this file:

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

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

WORKING-STORAGE SECTION.

77 WF-CONTENT PIC X(150).
77 STAT-CONTENT PIC X.

... ..
LINKAGE SECTION.

COPY "LS-HTTP.FDT".

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

    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>First html sample</h1>".  
WRITE LINEA FROM "</center>".  
WRITE LINEA FROM "</body>".  
WRITE LINEA FROM "</html>".  
  
CLOSE CONTENT.  
STOP RUN.  
END PROGRAM.
```

In this little sample we do only a display of a message that be say:

First html simple

All tags used in this sample are html, so not be exist any problem with this part.

The interesting heres is:

- We use the LS-HTTP.FDT file to obtain all variables that give to us.
- We use the file name in the variable: HTTP-OUTPUT
- Create the file with the content desired

For create PDF content is the same way, only the tags that we use for build a PDF report need be the same tags for build a report in COBVIEW. Remember that COBVIEW is a program to view and print windows reports. And the same windows reports can be exported to PDF for internet. And COBHTTPD is the server to build and convert the COBVIEW reports online.

Work with sessions

COBHTTPD allow us work with sessions, with the aim of restore a value in any variable the next time that the user access the same program.

Like you know, the http protocol only return the content in any consult. Once that the content is returned to browser the connection between the server and client is lost. In this way the next time that the client consult another document is made another connection and the data can't be passed from any consult to another. Even though is the same document.

In order to avoid this problem COBHTTPD work sessions, wich allow save values and pass in the next consult automatically.

The sessions always are created and be updated in every consult. For save any value in any session since our COBOL program, we need create a tag with the next format:

<@ SESSION SAVE variable=valor @>

Sample:

<@ session save clave=1953 @>

Since our COBOL program we can generate the instruction in the next way:

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

For restore the value that we saved in the session, we need take the value like any input parameter. In the COBHTTPD configuration we need register the variable for the program, and declare in the record for obtain the value always in our program.

The variable always arrive to us like parameter, and inside our program we need update this value with the tag SESSION SAVE for the next time contain the last value.

Another solution for resolve this problem is that always put this variable and value inside our URL for the program, so this value too arrive like input parameter included in the same url address.

Including external files

COBHTTPD too can allow include external files. This is of great help because we can put complete libraries or routines in any file and use in different programs, and include this file in every html page that we generate.

This files can be HTML content or also JavaScript routines that we can need in our web page that we generate.

For do this, we use the next special COBHTTPD tag:

```
<@ COPY "filename.ext" @>
```

The file name need be enclosed between quotes or apostrophos.

The real ubication for the file always is in the directory where the program exits.

In another words, the directory that have registered the project for run the programs.

If you want put by example all your JavaScript files in any directory called js, this will exist inside of directory of programs.

So, for include a file called formas.js that was saved in the js directory, the line for COBHTTPD need be:

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

With this line, COBHTTPD is going to replace this text by the content in the file formas.js. This is do automatically.

CGILIB Introduction

CGILIB is a library for build programs CGI type since COBOL and was created specifically for COBHTTPD.

This library contains some files that need be included in our COBOL programs, and calling the cgilib paragraphs we can build html content in easy way.

The CGILIB capacities are:

- Build HTML content without write any html tag
- Can build tables for do format to our forms
- Can build forms for capture and show the data in HTML
- Library for build a BROWSER and navigate in the records of any file.
- Can build buttons and url links
- Creation of themes for change the presentation with only change the desired theme name.

Although this library was created thinking in COBHTTPD, is not included like part of the same program, this because the use is optional, so you can use COBHTTPD without CGILIB for build tour COBOL-HTML programs.

For use CGILIB need download in separated way and the documentation too. Nevertheless I think that this library can help to build html applications more fast.