

About HTTP directives for CGIDEV2

by Giovanni B. Perotti
gb_perotti@easy400.net
August 2008

This document is for people new to CGIDEV2 and needing some introduction to the System i HTTP server.

1. Original and Apache HTTP directives

In 1996, when the AS/400 started supporting the HTTP server, the directives of an HTTP instance used to be members of physical file `QUSRSYS/QATMHTTFC`. The command for maintaining such directives used to be `WRKHTTFCFG`.

The language of such directives was unnamed at that time.

Later on, when IBM decided to adopt the most widespread language for HTTP directives – Apache – those initial language for directives was named Original, just to have a name that would distinguish it from Apache. The first AS/400 (perhaps iSeries at that time) release supporting Apache was V5R1. In my opinion, that initial support was not much, so I started using it with V5R2.

The Original HTTP server was dropped off with V5R3.

Therefore:

- If your OS/400 release is before V5R1, you must work with the Original HTTP directives
- If your OS/400 release is V5R1, you may choose between Original and Apache, though I would still recommend Original
- If your OS/400 release is V5R2, you can still choose between Original and Apache, but I now would recommend Apache
- If your OS/400 release is V5R3 or a further one, you just have Apache.

2. About the *ADMIN wizard

Since the beginning in IBM was born the idea to develop, under the internal HTTP instance *ADMIN (running on port 2001), a wizard to maintain HTTP directives. The objective was obviously that of easing out such a job for non experienced people. In my opinion that objective was very far from being met, for the following reasons:

- It takes an HTTP expert to understand the rules of the wizard game
- If your national language is other than English, the help text is hopeless, as the translation to your national language is usually not understandable or misleading
- You are then never aware of the correspondences between the real HTTP directives and the wizard screens
- You may hit cases where you completely waste your HTTP directives and you do not know how to recover

For the above reasons, I would recommend to write your own HTTP directives, hopefully starting from working examples. At least you know what you did.

3. The HTTP control file QUSRSYS/QATMHINSTC

This is a multimember physical file, containing a member for each HTTP instance.

The name of a member is the name of the HTTP instance.

You may use command

```
WRKMBRPDM FILE(QUSRSYS/QATMHINSTC)
```

to list and display its members.

When command `STRTCPSVR *HTTP` is entered, all the members are processed in order to start the appropriate HTTP instances.

Each member is made of a single record.

The record contains information about the related HTTP instance:

- a. the type of this HTTP instance : Original or Apache
- b. the qualified name of the stream file containing the directives of this HTTP instance
- c. whether this instance should be auto-started when command `STRTCPSRV *HTTP` is run.

Example 1:

```
-d /qsys.lib -f qusrsys.lib/qatmhttpc.file/default.mbr -AutoStartY
```

- a. as nothing is said about the type of this instance, that is an Original HTTP instance
- b. the HTTP directives of this instance are in the physical file QUSRSYS/QATMHTTPC member

DEFAULT

- c. this HTTP instance is autostarted when command STRTCPSRV *HTTP is run.

Example 2:

```
-apache -d /cgidev -f conf/httpd.conf -AutoStartN
```

- a. -apache means that the type of this HTTP instance is Apache
- b. the qualified name of the stream file containing the directives of this HTTP instance is /cgidev/conf/httpd.conf
- c. this instance should not be auto-started when command STRTCPSRV *HTTP is run.

4. Creating your own HTTP directives file

In the rest of this document we shall assume that you want to write directives for an HTTP instance supporting CGIDEV2 (CGIDEV2 is made of library CGIDEV2, IFS directories /cgidev and /cgidevExtHtml).

A. Original HTTP directives

Add the following directives to member DEFAULT in file QUSRSYS/QATMHTTPC

```
#==== START of CGIDEV2 HTTP DIRECTIVES =====
Map /cgidev2/start /QSYS.LIB/CGIDEV2.LIB/DEMOHTML.FILE/DEMOFRAME.MBR
Map /cgidev2/setup /QSYS.LIB/CGIDEV2.LIB/DEMOHTML.FILE/SETUPCSA.MBR
Map /cgidev2h/*.htm /cgidev2h/*.mbr
Map /cgidev2o/*.htm /cgidev2o/*.mbr
Map /cgidev2oit/*.htm /cgidev2oit/*.mbr
Map /cgidev2o/* /QSYS.LIB/CGIDEV2.LIB/DEMOHTML.FILE/*
Map /cgidev2oit/* /QSYS.LIB/CGIDEV2.LIB/DEMOHTMLIT.FILE/*
Map /cgidev2h/* /QSYS.LIB/CGIDEV2.LIB/HTMLSRC.FILE/*
Map /cgidev2r/* /QSYS.LIB/CGIDEV2.LIB/QRPGLESRC.FILE/*
Map /cgidev2l/* /QSYS.LIB/CGIDEV2.LIB/QCLSRC.FILE/*
Map /cgidev2d/* /QSYS.LIB/CGIDEV2.LIB/QDSSRC.FILE/*
Map /cgidev2f/* /cgidev/*
Pass /QSYS.LIB/CGIDEV2.LIB/*
Pass /cgidev/*
Pass /cgidevexthtml/*
Exec /cgidev2p/* /QSYS.LIB/CGIDEV2.LIB/* %%EBCDIC%%
AddType .js application/x-javascript 8bit
Enable Get
Enable Post
#===== END OF CGIDEV2 DIRECTIVES =====
```

To restart the DEFAULT HTTP instance

```
STRTCPSVR SERVER(*HTTP) RESTART(*HTTP) HTTPSVR(DEFAULT)
```

To run CGIDEV2, from your browser

```
http://as400IPaddress/cgidev2/start
```

where *as400IPaddress* is the IP address of your AS/400 (iSeries/System i, ...what_you_call_it)

B. Apache HTTP directives

Apache HTTP directives are already available in the IFS configuration file
/cgidev/conf/httpd.conf

You just need to add an HTTP instance member (pointing to this IFS configuration file) to file
QUSRSYS/QATMHINSTC .

Do this by running the following command:

```
CPYF FROMFILE(CGIDEV2/QATMHINSTC) TOFILE(QUSRSYS/QATMHINSTC)
FROMMBR(CGIDEV2APA) TOMBR(CGIDEV2APA) MBROPT(*REPLACE) CRTFILE(*YES)
```

To start the HTTP instance CGIDEV2APA, enter the following command:
`srtcpsvr *http httpsvr(cgidev2apa)`

If you look at the CGIDEV2APA configuration directives in stream file
`/cgidev/conf/httpd.conf`

you will realize that this instance works on port 8014.

Therefore, to run CGIDEV2, in your browser enter

`http://as400IPaddress:8014/cgidev2/start`

where *as400IPaddress* is the IP address of your AS/400 (iSeries/System i/ ...what_you_call_it)