What is a Web Server

- A Web Server is a program running on a system
  - This system is referred to as a Server
- It sends files to web users
- It routes requests from Web users to other system resources
- It calls programs on the server
  - Using COMMON GATEWAY INTERFACE (CGI) protocol
- It routes output from CGI programs to web users
- It processes SSI (Server-side include) directives
- It offers secured and unsecured data transfer
Available Web Servers

- Original web server is CERN
  - The original HTTP Web Server
  - Most widely used web server on AS/400 and iSeries
  - V3.0A is the final version. It was generated July 15 1996
- Apache – www.Apache.org (free/open source)
  - Most widely used web server in the world (well over 70% of market)
- Microsoft’s Internet Information Server (IIS)
- Novell Web Server for NetWare
- Domino Web Server (uses the “Go” Web server)
- WebSphere Web Server (it is Apache)
- Jigsaw – W3C’s open source web server written in Java
  - Version 2.2.4 available as of Feb 18, 2004

- Other commercial (3rd party) web servers also available

iSeries Web Server

- Web Servers on iSeries are batch jobs running in a subsystem
  - Subsystem name: QHTTPSVR
  - The Web Server program runs in this subsystem
- You create a configuration for the web server
- You create an instance of the web server
  - Associate the configuration with the instance
- Multiple copies of the instance may run simultaneously
  - Each copy of the web server instance is a separate job running in the QHTTPSVR subsystem
Start/Stop the Web Server

- Web Servers use TCP/IP communications
- Starting a web server is starting a TCP/IP server
- Use the STRTCPSVR command
  - Specify the SERVER(*HTTP) parameter
  - Specify HTTPSVR(inst-name) or HTTPSVR(*ALL)
- End the server by using ENDTCPWSVR command
  - Specify SERVER(*HTTP) for the server to end
  - Specify the instance to end on the HTTPSVR parameter

Selection or command:

===> STRTCPSVR SERVER(*HTTP)
**Work Active Jobs**

Type options, press Enter.
2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display msg 
8=Work with spooled files   13=Disconnect ...

<table>
<thead>
<tr>
<th>Opt</th>
<th>Subsystem/Job</th>
<th>User</th>
<th>Type</th>
<th>CPU %</th>
<th>Function</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>QBATCH</td>
<td>QSYS</td>
<td>SBS</td>
<td>.0</td>
<td>DEQW</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>QCMN</td>
<td>QSYS</td>
<td>SBS</td>
<td>.0</td>
<td>DEQW</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>QCTL</td>
<td>QSYS</td>
<td>SBS</td>
<td>.0</td>
<td>DEQW</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>QSYSSCD</td>
<td>QPGMR</td>
<td>BCH</td>
<td>.0</td>
<td>PGM-QEZSCNEP</td>
<td>EVTW</td>
</tr>
<tr>
<td>_</td>
<td>QHTTPSVR</td>
<td>QSYS</td>
<td>SBS</td>
<td>.0</td>
<td>DEQW</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>DEFAULT</td>
<td>QTMHHTTP</td>
<td>BCH</td>
<td>.1</td>
<td>PGM-QZHBHTTP</td>
<td>CNDW</td>
</tr>
<tr>
<td>_</td>
<td>DEFAULT</td>
<td>QTMHHTTP</td>
<td>BCI</td>
<td>.0</td>
<td>TIMW</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>DEFAULT</td>
<td>QTMHHTTP</td>
<td>BCI</td>
<td>.0</td>
<td>TIMW</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>DEFAULT</td>
<td>QTMHHTTP</td>
<td>BCI</td>
<td>.0</td>
<td>TIMW</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>DEFAULT</td>
<td>QTMHHTTP</td>
<td>BCI</td>
<td>.0</td>
<td>TIMW</td>
<td></td>
</tr>
</tbody>
</table>

**Ending the Web Server**

- Issue the ENDTCP SVR SERVER(*HTTP) command
  - The original “classic” HTTP server takes several minutes to end (if web was active)
  - The Apache server ends immediately in most cases or when any currently running processes finish
- All jobs and then the subsystem are ended
Selection or command:

```plaintext
===>
ENDTCPSVR SERVER(*HTTP)
```

**QHTTPSVR (subsystem)**

- HTTP Web Server Job 1
- HTTP Web Server Job 2
- HTTP Web Server Job 3
- HTTP Web Server Job 4
- HTTP Web Server Job 5

---

**More Web Server Jobs**

- The more web server jobs the more hits your web site can handle
- The Server automatically increases number of jobs as load increases
- Default is to start 5 instances or jobs
- Minimum is 1, but that really turns into 3
  - Controlling Job, a Multi-thread, and a single-thread CGI job
- On a high-end system, with tens of thousands of hits per day, hundreds of Web server jobs are common
- For an intranet, 5 to 10 web jobs is normally enough
Starting the Administrator Instance

- IBM provides a configure “wizard”
- Configuration Wizard is available on port 2001
- To configure:
  - Start the *ADMIN instance
  - Launch a browser
    - Point it to port 2001 of your system’s IP address
  - Select: Web Server Powered by Apache

Starting the ADMIN Instance

Selection or command:

```plaintext
===> STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```
Get to the Web Server Config

  - Use this to get to the IBM-supplied webpage for configuring the web server
  - NOTE: Your IP will be different

- Access is to either server:
  - Original/Classic/Native HTTP Server
  - IBM Web Server by Apache

The Original HTTP Configuration

- Each instance of the HTTP Server requires a configuration
- Configuration editing via WRKHTTPCFG command
- Configurations are saved as separate members in a database file
  - File name is: QATMHTTPC in QUSRSYS
  - Record length is 640-bytes
    - File has one field, named: CHARFIELD Char(640)
    - Many records; several configuration directives
- Usage is similar to DOS text files
  - Except it is a “real” OS/400 DB2 Database file
Creating Configurations

- New configurations are created by the WRKHTTPPCFG command
- The CFG parameter of WRKHTTPPCFG identifies the configuration name to work with
- WRKHTTPPCFG CFG( CONFIG | config-name )
  - Specify the name of the configuration to edit or use the default name of "CONFIG"

Using WRKHTTPPCFG

WRKHTTPPCFG CFG( MYCFG )

WrkHttpCfg causes a member to be added to QATMHTTPPC.
Apache Configuration

- Use the “Wizard” from *ADMIN configuration web page

- Create the initial configuration

- Edit using the *ADMIN web pages

Apache Config Location

- Configuration file is named httpd.conf
- It is stored on the IFS
  - The base directory is the \www directory
  - The subdirectory is the same name as your web server (specified on the Admin page)
  - Within the subdirectory is another subdirectory named conf
  - The httpd.conf file is stored there

- e.g., \www\mywebsite\conf\httpd.conf
- Can use the EDTF command to edit the file
Apache Configuration

/www/mywebsite/conf/httpd.conf

/A subfolder “subdirectory” off the Root directory

/www

/mywebsite

/The name you’ve assigned to this web configuration

/conf

/The conf subdirectory contains the configuration file

/httpd.conf

/The configuration file name

Creating an Instance

- The Web Server runs an instance of a configuration
- Instances are activations of a Configuration
- The Instance name is specified on the first element of the HTTPSVR parameter of the STRTCPSVR command
- An Instance is associated with a configuration by specifying the configuration name as the second element of the HTTPSVR parameter of the STRTCPSVR command
HTTPSVR Parameter Elements

- STRTCPSVR’s HTTPSVR Parameter
- Supports two elements
  - Instance name
  - Instance settings
- Instance name is the name of the instance
  - You make up this name
- Instance settings are well known settings
  - Configuration name, min/max instances to start, etc.

Overriding Start Up Settings

```plaintext
STRTCPSVR SERVER(*HTTP) + HTTPSVR(MYWEB '-r MYCFG -p 8088')
```

Start Up Settings

- Configuration
- Port
- Instance
- Start Up Settings

This stuff is optional.
Instance Start Up Settings

- `-r config` Configuration name
  - Name of the configuration to associate with this instance
- `-p nnnn` Port to use for the instance
  - Same as the PORT directive in the config file
- `-minat nn` Minimum instance jobs to start
- `-maxat nn` Maximum instance jobs to start

Why Instance AND Config?

- A configuration contains the settings for an instance of the web
- An instance allows you start multiple copies of the configuration with different names
- Example configuration with multiple instances
  - `WRKHTTPCFG CFG(LIVE)`
  - `STRTCPSVR SERVER(*HTTP) HTTPSVR(LIVE ‘-r LIVE’)`
  - `STRTCPSVR SERVER(*HTTP) HTTPSVR(DEMO ‘-r LIVE’)`
  - Each uses the LIVE configuration file
Config, Instance, Job

- Config File1
  - Instance1
    - Web Job
  - Instance2
    - Web Job

- Config File2
  - Instance
    - Web Job

Associating an Instance with a Configuration Member

- QATMHTTPC
  - Contains Configuration directives
  - Member name is same as configuration name

- QATMHINSTC
  - Contains Instance Information
  - Member name is same as instance
  - Contains one record
  - Record points back to QATMHTTPC member containing configuration for the instance
How to Associate an Instance with a Configuration

**STRTCPSVR *HTTP HTTPSVR(MYWEB ‘-r MYCFG’)**

The first time, or whenever start-up settings are specified, the start-up settings are saved in the QATMHINSTC file. They are automatically retrieved the next time STRTCPSVR is issued without the start-up settings portion of HTTPSVR being specified.
Previous Association

```
STRTCPSVR *HTTP HTTPSVR(MYWEB)

QATMHTTPC
MBR: MYCFG
```

```
QATMHINSTC
MBR: MYWEB
```

```
-r /QSYS.LIB/QUSSYS.LIB/QATMHTTPC.FILE/MYCFG.MBR
```

One Record per member containing expanded HTTPSVR settings

---

Classic Web Server Directives

- Configuration Directives for HTTP Server
- Many, many directives available
- Not the purpose of this presentation to define all of them
- The main directives to learn include:
  - Map, Pass, Exec, Fail
  - Redirect
  - AddType
PASS Configuration Directive

- Controls which files can be accessed and on which path
- Allows translation of a template to a path
  - *Path* is the PC-DOS path/directory term
- Access to the directories and files specified with PASS is restricted to those authorized to QTHMHTTP user profile

Example PASS Directives

```
PASS /files/* /QSYS.LIB/COZZILIB.LIB/QHTMLSRC.FILE/*.MBR
PASS /html/* */myweb/docs/*
```

```
FROM: http://www.rpgiv.com/files/index
TO: /QSYS.LIB/COZZILIB.LIB/QHTMLSRC.FILE/INDEX.MBR
```

```
TO: /myweb/docs/index.html
```
MAP Configuration Directive

- Only does the mapping, no processing is performed
- Allows pattern translation from HTML forms and URL-encoded strings to “real” locations on your system
- Allows translation of a template to a path
  - Path is the PC-DOS path/directory term
- Access to the directories and files specified with PASS is restricted to those authorized to QTMHHTTP user profile

Example MAP Directives

MAP from-pattern to-pattern

MAP /files/* /QSYS.LIB/COZZILIB.LIB/QHTM SRC.FILE/*.MBR
FROM: http://www.rpgiv.com/files/index
TO: /QSYS.LIB/COZZILIB.LIB/QHTM SRC.FILE/INDEX.MBR

MAP /html/* /myweb/docs/*
TO: /myweb/docs/index.html
FAIL - Security Directive

- Controls access to directories or folders
- If accessibility is to a directory identified by the FAIL directive
  - Access is denied

FAIL /myweb/docs/private/*
PASS   /myweb/docs/*

Any files in the /mydocs IFS directory may be accessed, except for files located in the /myweb/docs/private subdirectory. Files in that directory may not be accessed.

EXEC Configuration Directive

- Enables execution (runtime access)
- Only those programs or library with an EXEC access can be run
- Contains an integrated “MAP” capability

EXEC  from-pattern  enabled-item
Example EXEC Directives

MAP /cgi-bin/* /cgi-bin/*.pgm
EXEC /cgi-bin/* /qsys.lib/cgilib.lib/*

You’d specify this syntax in your HTML Form (more on this later).

Apache Web Server Directives


- Map equivalent
  - Classic: Map /cgi-bin/* /cgi-bin/*.pgm
  - Apache: MapMatch ^/cgi-bin/(.*) /cgi-bin/$1.pgm

- Exec equivalent
  - Classic: Exec /cgi-bin/* /qsys.lib/cgilib.lib/*
  - Apache: ScriptAliasMatch ^/cgi-bin/(.*) /qsys.lib/cgilib.lib/$1
More on Apache CGI Directives

- Enable CGI Program Calls
  - Options +ExecCGI

- Add a .pgm to the CGI program name
  - MapMatch ^/cgi-bin/(.*) /cgi-bin/$1.pgm

- Translate the /cgi-bin/ prefix into the QSYS name
  - ScriptAliasMatch ^/cgi-bin/(.*) /qsys.lib/cgilib.lib/$1

Summary

- Original Native HTTP Server is widely used on iSeries (currently)
- Apache is becoming the standard for OS/400
- Domino is also popular for web serving on iSeries Intranets
  - But Domino’s popularity and usefulness as a web server is dwindling