Deploying Tivoli NPM/IP Java Servlets on WebSphere Application Server for OS/390 or z/OS

1. Scenario Overview

The purpose of this document is to describe the installation of the NPM/IP Java Servlets GUI code to run on WebSphere Application Server for OS/390.

Tivoli NPM/IP Java Servlets GUI code requires
- Java Servlet Development Kit 2.0 or above
- JDK 1.2.1 or above

To fulfill these requirements, you must install the NPM/IP Java Servlets GUI on WebSphere Application Server V3.5 or later.

This scenario uses the following software on an OS/390 system. NPM/IP and WebSphere Application Server may run on different OS/390 hosts.

- Tivoli NetView Performance Monitor for TCP/IP V1R3 (5698-PMI).
- PTF for APAR OW51942 - this PTF is required to fix a SNMP problem when running NPM/IP Java Servlets under the WebSphere for OS/390 system.

> NOTE: Without this PTF, you will get no response when using the SNMP function to query ASCII devices. This problem is due to an EBCDIC to ASCII conversion error.

- TME 10 NetView for OS/390 C runtime V1R2 (program number 5697-B82). Required for NPM/IP V1.3. This program is automatically shipped with the Tivoli NetView Performance Monitor for TCP/IP product.
- IBM WebSphere Application V3.5 (Program number 5655-A98) WebSphere Application Server V3.5 Standard Edition for OS/390 must be installed on OS/390 V2R8, or later, or zOS systems. We have OS/390 V2R10 running on our test system where this scenario was implemented and tested. WebSphere V3.5 Application Server Advanced Edition and WebSphere Application Server V4.0 may also be used.

2. HTTP and WebSphere Application Server Verification

Instructions for installing the WebSphere Application Server Standard Edition for OS/390 Version 3.5 can be found in the WebSphere Application Server for OS/390 Version 3.5 Program Directory. This program directory is shipped with the product and can also be obtained from the product library page at URL:

http://www.ibm.com/software/websphere/appserv/library_390.html

Once the Application Server is started, you can verify the install by entering the following URL from a browser to invoke the IBM-provided Installation Verification Program:

http://your.server.name /webapp/examples/index.html

3. Create a HFS to store NPM/IP Java code

To maintain NPM/IP Java Servlet separately from other WAS applications, you may create a HFS dataset dedicated to store Tivoli NPM/IP Java code.

```
//NPMIPHFS JOB (????,????),'MOY JOB',MSGCLASS=O,CLASS=A,
//         NOTIFY=????????,REGION=4000K TYPRUN=HOLD
//ALLOC1  EXEC PGM=IEFBR14
//SYSPRINT  DD SYSOUT=*
//*
//* NOTE: THESE HFS DATA SETS MUST BE DFSMS MANAGED .
//*
//BASE     DD DSN=NPMIP.JAVA.HFS,DISP=(NEW,CATLG),
//            VOL=SER=OE1SMS,UNIT=SYSDA,
//            SPACE=(CYL,(8,3,1)),
```

It is recommended that all NPM/IP files are stored under a single directory tree. In this document, all NPM/IP Java files are placed in a directory named /usr/lpp/npmip. After the HFS dataset is created, mount it to /usr/lpp/npmip or a directory name of your choice. Issue the df command to verify that the NPM/IP HFS is mounted successfully.

```
> mount -f  NPMIP.JAVA.HFS /usr/lpp/npmip
> df  | grep /usr/lpp/npmip
/usr/lpp/npmip (NPMIP.JAVA.HFS)      11824/12000    4294967294 Available
```

Issue the mkdir commands to create the following subdirectory structure under /usr/lpp/npmip or under the NPM/IP directory name that you have chosen for your system.

```
> cd /usr/lpp/npmip
> mkdir  Help
> mkdir  Htdocs
> mkdir  Images
> mkdir  Msbchart
> mkdir Servlets
```

4. FTP Java Servlets code from CD to the OS/390 System

Tivoli NetView Performance Monitor for TCP/IP product shipment includes a copy of the Java Servlets CD (LK3T-5785-01). Mount this CD on a workstation. Use FTP to transfer files on the CD to the OS/390 system.

- Transfer all files in the **Help** directory to **/usr/lpp/npmip/Help**.
- Transfer all files in the **Htdocs** directory to **/usr/lpp/npmip/Htdocs.**
- Transfer all files in the **Images** directory to **/usr/lpp/npmip/Images.**
- Transfer all files in the **Servlets** directory to **/usr/lpp/npmip/Servlets.**
- Transfer all files in the **Chart** directory to **/usr/lpp/npmip/Msbchart**

NOTE: All of the files and directory names are case sensitive**. All files must be transferred in binary mode.** When you ftp HTML files to the OS/390 system in binary mode, the files are stored in the ASCII and are not converted to EBCDIC. Therefore, the html files will be illegible if you view them using OMVS oedit or TSO ishell utilities.

Following are sample FTP commands to transfer the NPM/IP files to an OS/390 system.
- lcd command is used to define the local current directory. In the following example, E: represent the CDROM drive letter where the NPM/IP CD is mounted.
- cd is used to change directory name at the FTP server (OS/390 in this case).
- prompt command is used to turn off interactive mode
- bin command is used to change file representation to binary

```
ftp os390host
bin
lcd E:\Help
cd /usr/lpp/npmip/Help
prompt
mput *
lcd E:\Htdocs
cd /usr/lpp/npmip/Htdocs
mput *
lcd E:\Images
cd /usr/lpp/npmip/Images
mput *
lcd E:\Servlets
cd /usr/lpp/npmip/Servlets
mput *
lcd E:\Chart
cd /usr/lpp/npmip/Msbchart
mput *
quit
```

5. Customize HTTP Server Configuration File (httpd.conf)

IBM HTTP Server powered by Apache is a server based on the Apache Web server developed by the Apache Group (www.apache.org). IBM HTTP Server is included with the WebSphere Application Server V3.5. HTTP Server must be running to process all URL requests. Based on the configuration statements in the httpd.conf file, HTTP server can pass URL requests to be serviced by the WebSphere Application Server. Verify or add the following statements to support NPM/IP in the httpd.conf file.

- Verify the port statement used to define the port number assigned to the HTTP server. You may bring up a test HTTP server to test out the NPM/IP code. Each HTTP server must use a unique port number.

  Port 8973

- Add the following two Web server directives to the httpd.conf configuration file to provide the HTTP server with the entry point to Application Server's initialization, request processing and exit routines. Verify that the WebSphere Application Server configuration file (was.conf) specified on the ServerInit statement contains the parameters needed for the NPM/IP. Step 6) in this document describes the necessary modifications for the was.conf file. If WAS V4.0 is used, you will need to specify was400plugin.so instead of was350plugin.so on these statements.

  ServerInit /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:init_exit /usr/lpp/WebSphere,/usr/lpp/npmip/was.conf

  ServerTerm /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:term_exit

  Notes: In this example, the ServerInit directive is split for printing purposes. In the actual httpd.conf file, each directive must be on a single line.

- Define map directives to add the NPM/IP directory prefix, /usr/lpp/npmip, to the user's URL requests. These mapping statements redirect the URL requests to the actual physical locations of the NPM/IP Java component on the OS/390 system. For example, when a user enters "http://os_390_web_server/logon.htm", the HTTP server on the OS/390 will map that URL to /usr/lpp/npmip/Htdocs/logon.htm, which is where the actual logon.htm resides on the OS/390 system in this example.

  These statements are case sensitive and should be placed before any other more general map or pass statements in the httpd.conf file. HTTP server scans through all directives in the httpd.conf file sequentially and takes the appropriate action whenever a match is made.

  ```
  Map     /logon.htm      /usr/lpp/npmip/Htdocs/logon.htm
  Map     /SNMP.html      /usr/lpp/npmip/Htdocs/snmp.html
  Map     /query.html     /usr/lpp/npmip/Htdocs/query.html
  ```

```
Map     /mib.html       /usr/lpp/npmip/Htdocs/mib.html
Map     /htdocs/*       /usr/lpp/npmip/Htdocs/*
Map     /images/*       /usr/lpp/npmip/Images/*
Map     /Chart/*        /usr/lpp/npmip/Msbchart/*
Map     /Help/*         /usr/lpp/npmip/Help/*
Map     /help/*         /usr/lpp/npmip/Help/*
Map     /servlets/*     /usr/lpp/npmip/Servlets/*
```

- Add Service statement to pass all NPM/IP URL requests to WebSphere Application Server. If you are using WAS 4.0, you will need to specify was400plugin.so instead of was350plugin.so on this statement.

  Service /usr/lpp/npmip/*
  /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:service_exit

Note: This service statement must be placed after the map statements. In this example, the Service directive is split for printing purposes. In the actual httpd.conf file, this directive is on a single line.

The Service directives within the hosting Web server's httpd.conf file indicate which requests the WebSphere Application Server is to process. In this scenario, all NPM/IP URL requests will be added with a /usr/lpp/npmip prefix via the Map directives and routed to the WebSphere Application Server via the Service directive. WebSphere Application Server will then serve all NPM/IP URL requests.

6. Customize WebSphere Application Server Configuration File

A Web application is a grouping of Web components, such as servlets, Java Server Pages (JSPs), and static files such as HTML files and GIF files, that is defined and then deployed into a WebSphere virtual host and managed as a single unit. Each web application must be identified with a unique name. In this document, **npmip** will be used as the <webapp-name> to represent the NPM/IP Java Servlets application. There are two distinct sets of attributes that you must configure for each Web application:

a) Deployment Properties

Deployment properties define the physical residency and characteristics of the the WebSphere Application Server into which the Web application is being deployed, such as the fully qualified directories that comprise the Web application's classpath. The format for defining deployment properties in the WebSphere Application Server configuration file is:

deployedwebapp.<webapp-name>.property=value

The deployment attributes enable the ServletContext object to correctly resolve these physical residencies and characteristics at run time and efficiently deploy the Web application. Following deployment properties are coded to support the NPM/IP Java Servlet application:

```
deployedwebapp.npmip.rooturi=/usr/lpp/npmip
deployedwebapp.npmip.documentroot=/usr/lpp/npmip
deployedwebapp.npmip.classpath=/usr/lpp/npmip/Servlets/Ae
s.jar:/usr/lpp/npmip/Servlets/jSNMP.jar
deployedwebapp.npmip.autoreloadinterval=10000
deployedwebapp.npmip.host=default_host
```

Notes: In this example, some deployedwebapp statements are split for printing purposes. In the actual was.conf file, each directive must be on a single line.

- `deployedwebapp.npmip.rooturi -` provides the pattern by which a web application is known within its virtual host. In this scenario, all NPM/IP requests' URL will be prefixed with /usr/lpp/npmip when they are received by the WebSphere Application Server.

- `deployedwebapp.npmip.host -`specifies the administrative name of the virtual host in which a Web application is deployed. In this example, the default host will be used to deploy the NPM/IP application. You may change to use any host definition in your WAS configuration.

- `deployedwebapp.npmip.classpath -` specifies the classpath that the application level class loader searches for a servlet when the system class loader cannot find it. This property is REQUIRED and must specify the directory where the servlet resides. Aes.jar and jSNMP.jar are the class files used by NPM/IP.

- `deployedwebapp.npmip.documentroot -` specifies the name of the directory root that contains the files associated with the NPM/IP Web application.

- deployedwebapp.npmip.autoreloadinterval - specifies that NPM/IP Web application is to be reloaded if changes are detected. WebSphere Application Server waits 10,000 ms between checks.

b) Definitional attributes

Definitional attributes define the characteristics of a Web application such as the servlets that are part of the application, the HTML, image file, servlet, and JSP mapping information. Properties relating to the configuration of Web applications have the format

webapp.<webapp-name>.property=value.

The same name is used on both the deploywebapp and webapp properties associated with a specific Web application. **npmip** is the unique <webapp-name> name being used to identify the NPM/IP Java Servlet code.

Following webapp property definitions were added to was.conf file to specify its definitional attributes. All webapp definition has the following form:

```
webapp.npmip.filemapping=*.htm
webapp.npmip.filemapping=*.html
webapp.npmip.filemapping=*.gif
webapp.npmip.filemapping=*.Gif
webapp.npmip.filemapping=*.jpg
webapp.npmip.filemapping=*.jar
webapp.npmip.servletmapping=/Servlets/*
```

Tivoli NPM/IP Java Servlet GUI uses several types of Web components including HTML file, gif image files, jar file. **webapp.npmip.filemapping** statements provides the WebSphere Application Server with information about the types of Web components used by NPM/IP. Files ending with html, htm, jpg, gif are treated as static files. **webapp.npmip.servletmapping** is used to designate the pattern for the NPM/IP Java Servlet. NPM/IP Java Servlet URL, /servlets, is mapped to /usr/lpp/npmip/Servlets by the HTTP server. In order for WebSphere Application Server to match URL /usr/lpp/npmip/Servlets to the NPM/IP web application, /usr/lpp/npmip is assigned as the value of deployedwebapp.npmip.rooturi property and /Servlets is assigned as the value of **`webapp.npmip.servletmapping`** property. Concatenation of these two values equals /usr/lpp/npmip/Servlets which matches the NPM/IP URL /usr/lpp/npmip/Servlets routed by the HTTP server.

NOTE: This document is written with the assumption that all NPM/IP Java Servlets files are placed in a file system and mounted to /usr/lpp/npmip. Your installation may choose a different directory to house the NPM/IP Java Servlets code. In that case, you will need to modify various statements (e.g. Map, Service, webapp ..etc.) in the httpd.conf and was.conf accordingly.

7. Start Web Server

After completing the above customization, you can start the WebSphere Application Server by creating and starting a start procedure similar to the following:

```
//IMWPROC  PROC P1='-B',
// P2='-r /usr/lpp/npmip/web/httpd.conf',
// P3='-v',
// LEPARM='ENVAR("_CEE_ENVFILE=/usr/lpp/npmip/httpd.envvars")'
//*************************************************************
//WEBSRV   EXEC PGM=IMWHTTPD,REGION=0K,TIME=NOLIMIT,
//   PARM=('&LEPARM/&P1 &P2 &P3')
//STEPLIB  DD DSN=SYS1.LEMVS.SCEERUN,DISP=SHR
//         DD DSN=SYS1.CRYPTO.SGSKLOAD,DISP=SHR
//         DD DSN=SYS1.CPP.SCLBDLL,DISP=SHR
//*******************************************************
//SYSIN    DD DUMMY
//OUTDSC   DD SYSOUT=*,DEST=HOLD
//SYSPRINT DD SYSOUT=*,DEST=HOLD
//SYSERR   DD SYSOUT=*,DEST=HOLD
//STDOUT   DD SYSOUT=*,DEST=HOLD
//STDERR   DD SYSOUT=*,DEST=HOLD
//SYSOUT   DD SYSOUT=*,DEST=HOLD
//CEEDUMP  DD SYSOUT=*,DEST=HOLD
```

8. Test Accessing the NPM/IP Java GUI From a Web Browser

After successfully initializing WebSphere Application Server, you can enter the following URL from your browser to invoke the NPM/IP Java GUI:

http://your.server.name:8973/logon.htm

Note: In this example, we are using 8973 as the WebSphere Application Server port number instead of the default port number. Therefore, we needed to specify the port number 8973 on this URL.

When a user enters the logon.htm request, HTTP server on the OS/390 will receive the request and the following processing will take place.

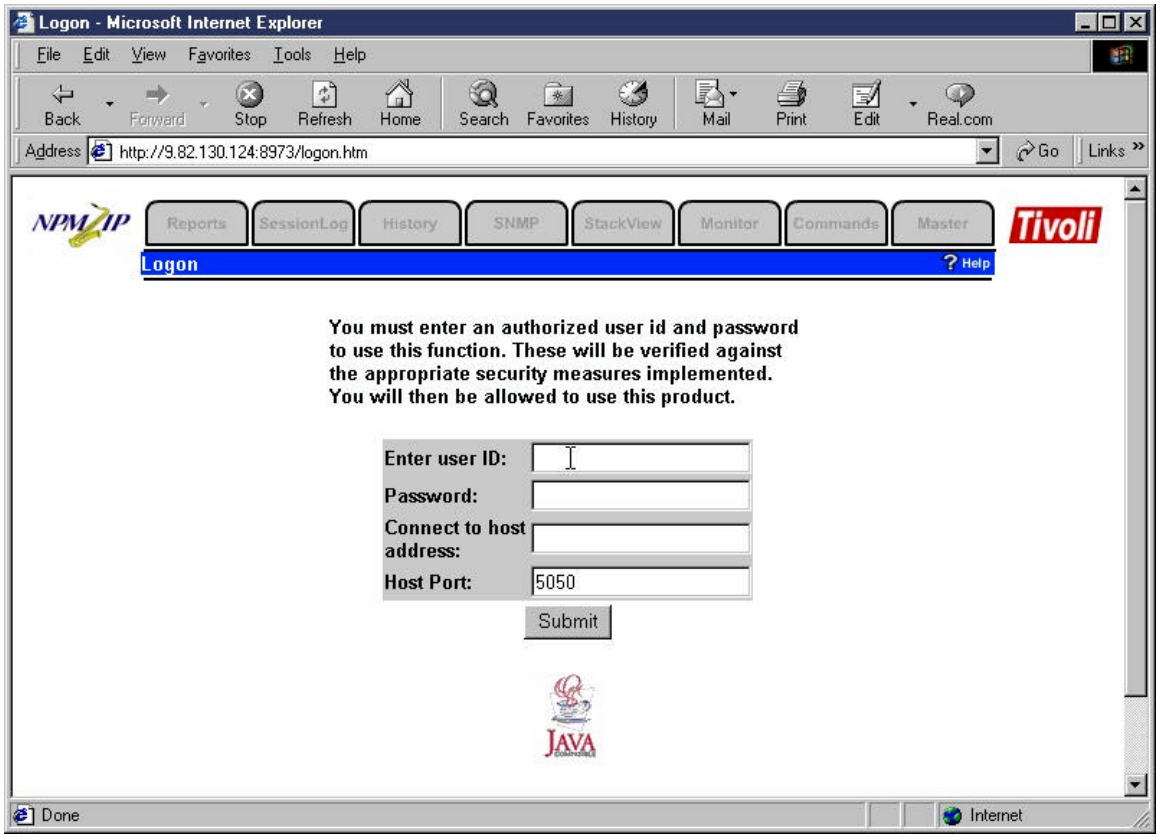
1. HTTP Server will find a match for the logon.htm in the httpd.conf and map the logon.htm to /usr/lpp/npmip/Htdocs/logon.htm. HTTP Server will continue processing the request with translated name of /usr/lpp/npmip/Htdocs/logon.htm through the rest of the statements in the httpd.conf file.

2. HTTP Server will find a match of /usr/lpp/npmip/Htdocs/logon.htm with the service statement of /usr/lpp/npmip/* in the httpd.conf and will pass the request to the WebSphere Application Server.

3. WebSphere Application Server will match /usr/lpp/npmip/Htdocs/logon.htm as a component of the npmip web application since its prefix matches the rooturi value of the npmip application as defined in the `deployedwebapp.npmip.rooturi=/usr/lpp/npmip statement.`

4. WebSphere Application Server will serve the file, /usr/lpp/npmip/Htdocs/logon.htm, because the webapp.npmip.filemapping=*.htm statement indicates that the npmip web application includes static HTML files. In this sample scenario, the actual logon.htm must reside at /usr/lpp/npmip/Htdocs/logon.htm on the OS/390 system. Otherwise, the user will get an error 404 - file not found error.

Following is a sample logon.htm file that you will receive if WebSphere Application Server is successful in locating and serving the logon.htm file to the user's web browser.



From the NPM/IP logon.htm, you can specify the user id, password and the IP address of the host where the NPM/IP host component is running to access the IP performance and monitoring functions provided by NPM/IP.

9. Sample HTTPD.CONF file (partial listing)

A partial listing of the HTTPD.CONF file is included in this and the next page. All statements referred in this document are highlighted and in italic.

```
InstallPath     /usr/lpp/internet
BindSpecific  off
DNS-Lookup     off
ServerRoot     /usr/lpp/npmip
Port     8973
imbeds on SSIOnly
UserId     %%SERVER%%
PidFile  /usr/lpp/npmip/httpd-pid
AccessLog     /usr/lpp/npmip/logs/httpd-log
AgentLog      /usr/lpp/npmip/logs/agent-log
RefererLog    /usr/lpp/npmip/logs/referer-log
ErrorLog      /usr/lpp/npmip/logs/httpd-errors
CgiErrorLog   /usr/lpp/npmip/logs/cgi-error
LogFormat     Common
LogTime       LocalTime
LogToSyslog on
AccessLogArchive purge
ErrorLogArchive  purge
AccessLogExpire 10
ErrorLogExpire  10
AccessLogSizeLimit 0
ErrorLogSizeLimit  0
AccessLogExcludeUserAgent ICS-ProxyAgent/4.2
AccessReportDoDnsLookup off
Enable     GET
Enable     HEAD
Enable     POST
Enable     TRACE
Enable     OPTIONS
Disable    PUT
Disable    DELETE
Disable    CONNECT
Welcome     Welcome.html
Welcome     index.html
Welcome     Frntpage.html
AlwaysWelcome     on
DirAccess              off
DirReadme              top
DirShowIcons           on
DirShowDate            on
DirShowSize            on
DirShowDescription     on
DirShowBrackets        on
DirShowCase            on
DirShowHidden          off
DirShowBytes           off
DirShowMaxDescrLength  25
DirShowMaxLength       25
DirShowMinLength       15
UseMetaFiles off
MetaDir      .web
MetaSuffix   .meta
#
ServerInit /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:init_exit /usr/lpp/WebSphere,/usr/lpp/npmip/was.conf
ServerTerm /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:term_exit
```

Sample HTTPD.CONF file continues on the next page

```
#        The following rules will allow anyone that knows your WEBADM
#        password to use the Web Server remote configuration application.
#
Protection IMW_Admin {
        ServerId        IMWEBSRV_Administration
        AuthType        Basic
        PasswdFile      %%SAF%%
        Mask            WEBADM,webadm
}
Protect /admin-bin/* IMW_Admin
Protect /Usage*      IMW_Admin
# ================================================================== #
#        Service directives for examples
# ================================================================== #
Service   /webapp/examples/* /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:service_exit
#
service /cgi-bin/htimage*        INTERNAL:HTImage*
service /cgi-bin/imagemap*       INTERNAL:HTImage*
service /Usage*                  INTERNAL:UsageFn
service /admin-bin/trace*        INTERNAL:TraceFn
#
#   MAP Directives for NPM/IP
#
Map             /logon.htm      /usr/lpp/npmip/Htdocs/logon.htm
Map             /SNMP.html      /usr/lpp/npmip/Htdocs/snmp.html
Map             /query.html     /usr/lpp/npmip/Htdocs/query.html
Map             /mib.html       /usr/lpp/npmip/Htdocs/mib.html
Map             /htdocs/*       /usr/lpp/npmip/Htdocs/*
Map             /images/*       /usr/lpp/npmip/Images/*
Map             /Chart/*        /usr/lpp/npmip/Msbchart/*
Map             /Help/*         /usr/lpp/npmip/Help/*
Map             /help/*         /usr/lpp/npmip/Help/*
Map             /servlets/*     /usr/lpp/npmip/Servlets/*
Service /usr/lpp/npmip/* /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:service_exit
#  End NPM/IP addition
#
Pass            /admin-bin/webexec/*  /usr/lpp/internet/server_root/admin-bin/webexec/*
Exec            /cgi-bin/*            /usr/lpp/internet/server_root/cgi-bin/*
Exec            /admin-bin/*          /usr/lpp/internet/server_root/admin-bin/*
Exec            /Docs/admin-bin/*     /usr/lpp/internet/server_root/admin-bin/*
Pass            /icons/*            /usr/lpp/internet/server_root/icons/*
Pass            /Admin/*.jpg        /usr/lpp/internet/server_root/Admin/*.jpg
Pass            /Admin/*.gif        /usr/lpp/internet/server_root/Admin/*.gif
Pass            /Admin/*.html       /usr/lpp/internet/server_root/Admin/*.html
Pass            /Docs/*             /usr/lpp/internet/server_root/Docs/*
Pass            /img-bin/*          /usr/lpp/internet/server_root/img-bin/*
Pass          /*                  /usr/lpp/internet/server_root/pub/*
# ================================================================== #
#        Performance directives.
# ================================================================== #
MaxActiveThreads  40
MaxPersistRequest 5
ServerPriority -10
PersistTimeout  10 secs
SSLClientAuth   off
UseACLs protectonly
.
.
.
```

NOTE: THIS IS NOT A COMPLETE HTTPD.CONF.. Other directives such as Add Type, Add Icon, Proxy directives are not printed here due to limited space.

10. Sample WAS.CONF

A sample WAS.CONF is listed in this and next page. All statements referred in this document are highlighted and in italic.

```
appserver.version=3.50
appserver.usesystemclasspath=false
appserver.libpath=
appserver.classpath=
appserver.name=defaultServletEngine
appserver.jvmpropertiesfile=
appserver.loglevel=INFO
appserver.logdirectory=/usr/lpp/npmip/logs
appserver.workingdirectory=
appserver.configviewer=
appserver.jspbasehrefadd=true
objectleveltrace.enabled=false
session.enable=true
session.urlrewriting.enable=false
session.cookies.enable=true
session.protocolswitchrewriting.enable=false
session.cookie.name=sesessionid
session.cookie.comment=servlet Session Support
session.cookie.maxage=-1
session.cookie.path=
session.cookie.secure=false
session.invalidationtime=180000
session.tablesize=1000
session.tableoverflowenable=false
session.dbenable=false
session.dbjdbcpoolname=SessionJDBCConnectionPool
session.dbtablename=
session.domain=
jdbcconnpool.default_jdbcpool.waitforconnectiontimeoutmilliseconds=30000
jdbcconnpool.default_jdbcpool.idleconnectiontimeoutmilliseconds=120000
jdbcconnpool.default_jdbcpool.inuseconnectiontimeoutmilliseconds=120000
jdbcconnpool.default_jdbcpool.jdbcdriver=ibm.sql.DB2Driver
jdbcconnpool.default_jdbcpool.databaseurl=
jdbcconnpool.default_jdbcpool.datasourcename=default_pool_datasource
#
jdbcconnpool.SessionJDBCConnectionPool.minconnections=10
jdbcconnpool.SessionJDBCConnectionPool.maxconnections=40
jdbcconnpool.SessionJDBCConnectionPool.waitforconnectiontimeoutmilliseconds=30000
jdbcconnpool.SessionJDBCConnectionPool.idleconnectiontimeoutmilliseconds=120000
jdbcconnpool.SessionJDBCConnectionPool.inuseconnectiontimeoutmilliseconds=-1
jdbcconnpool.SessionJDBCConnectionPool.jdbcdriver=ibm.sql.DB2Driver
jdbcconnpool.SessionJDBCConnectionPool.databaseurl=
# ================================================================ #
#   Virtual Host settings
# ================================================================ #
#      host.<virtual_hostname>.alias=<hostname>|localhost

host.default_host.alias=localhost
#
#-----------------------------------------------------------------#
#      host.<virtual_hostname>.mimetypefile
#
host.default_host.mimetypefile=
```

Sample WAS.CONF file continues on the next page.

```
#
#
# ================================================================ #
#
#   Deployed Web Application Settings
#
# ================================================================ #
#
# ================================================================ #
#
#   The following defines the examples web application used for
#   installation verification.
#
# ================================================================ #
deployedwebapp.examples.host=default_host
deployedwebapp.examples.rooturi=/webapp/examples
deployedwebapp.examples.classpath=$was_install_root$/AppServer/hosts/default_host/examples/servlets
deployedwebapp.examples.documentroot=$was_install_root$/AppServer/hosts/default_host/examples/web
webapp.examples.jspmapping=*.jsp
webapp.examples.jsplevel=1.1
webapp.examples.filemapping=/
webapp.examples.servlet.simpleJSP.code=SimpleJSPServlet
webapp.examples.servlet.simpleJSP.servletmapping=/simpleJSP
webapp.examples.servlet.simpleJSP.servletmapping=/simpleJSP.servlet
webapp.examples.servlet.showCfg.code=com.ibm.servlet.engine.config.ServletEngineConfigServlet
webapp.examples.servlet.showCfg.servletmapping=/showCfg
# ================================================================ #
#
#   The following defines by Wen for NPM/IP Java code
#
# ================================================================ #
deployedwebapp.npmip.rooturi=/usr/lpp/npmip
deployedwebapp.npmip.documentroot=/usr/lpp/npmip
deployedwebapp.npmip.classpath=/usr/lpp/npmip/Servlets/Aes.jar:/usr/lpp/npmip/Servlets/jSNMP.jar
deployedwebapp.npmip.autoreloadinterval=10000
deployedwebapp.npmip.host=default_host
webapp.npmip.filemapping=*.htm
webapp.npmip.filemapping=*.html
webapp.npmip.filemapping=*.gif
webapp.npmip.filemapping=*.Gif
webapp.npmip.filemapping=*.jpg
webapp.npmip.filemapping=*.jar
webapp.npmip.servletmapping=/Servlets/*
```