

**IBM ITSO Poughkeepsie
OS/390 in an e-business environment**

IBM WebSphere Application Server V1 for OS/390 Customization Hints



Roland Trauner
trauner@us.ibm.com

- ▶ To assemble these hints and tips we downloaded a version of WebAS V1 at the end of January 99.
- ▶ We successfully used JDK 1.1.6 - we also checked with 1.1.4
- ▶ Our system level was OS/390 R5 .Web server was DGW 5.0
- ▶
- ▶ For the web server part we used redbook SG24-2074-01
- ▶
- ▶ As of May 21, 1995 we did the installation again on OS/390 R7 using WebAS 1.1 and the IBM HTTP Server 5.1. Java Version was 1.1.6 (performance Version).

WebSphere Strategy



● OS/390 WebSphere Application Server V1

- ▶ replaced ServletExpress
- ▶ exception: Domino Go CA (DGW 5.0) still requires ServletExpress
- ▶ HTTP Server CA (HTTP Server 5.1) doesn't need ServletExpress anymore

● Where to get it from?

- http://www.software.ibm.com/webserver/appserv/download_v11.html#os390

● Was Version is it now - 1.0 or 1.1

- ▶ Good question
 - Web Page says it's 1.1 and has some 1.1 documents
 - Program directory says it's 1.0
 - jvm.properties says it's 1.0
 - download file is ejsw10.bin
- ▶ Whatever it is
 - It is included in OS/390 R7 (GA 3/99)



- ▶ WebAS doesn't include the Domino Go CA function. This function will be in the next release of the web server (IBM HTTP Server 5.1 for OS/390) which will be delivered with OS/390 R7.
- ▶ The function is then called HTTP Server CA.
- ▶ ServletExpress is no longer needed in R7

WebSphere Application Server V1



● WebAS Customization

- ▶ The following WebAS customization tips are intended to set up WebAS for multiple instances --- to be able to run multiple web servers with WebAS on the same system.
- ▶ The problem is that the current version of WebAS (V1) predicts it's configuration files on several "hard coded" locations within the WebAS directory structure.
- ▶ Solution:
 - Install WebAS to it's default directory /usr/lpp/WebSphere/AppServer
 - Run the Setup
 - Save the structure into a tar archive
 - Define a directory tree for WebAS servers
 - Restore the archive into one WebAS working directory
 - Customize the configuration files

- ▶ Another solution could be that you run the complete WebSphere installation in the different WebAS working directories

WebSphere Application Server V1



- **Create a WebAS Working Directory Structure**

- ▶ We suggest you to create a /was directory structure similar to our suggested web server directory structure:

```
/was/server1
    /server2
    /server3
    ...
```

- Using a directory close to root also helps to reduce the long pathnames by about 20 characters

- So instead of

```
/usr/lpp/WebSphere/AppServer/properties/server
/servlet/servletservice/jvm.properties
```

- it will be

```
/was/apple/properties/server/servletservice
/jvm.properties
```

- ▶ The was structure should correspond to the web server structure in /web/server1... to be easily able to match the WebSphere Application Server with the Web Server
- ▶ We suggested / WebAS instead of /was earlier. We changed to /was now - not for any real reason other than that is more similar to /web.

WebSphere Application Server V1



● Prepare the environment

- ▶ copy a new httpd.conf file out of:
 /usr/lpp/internet/samples/config/httpd.conf
 - into
 /was/httpd.conf
- ▶ Prepare your environment variables
 - Define the Java environment - especially the JAVA_HOME variable in .profile
- ▶ Make sure you act as "superuser" to be able to run the setup.
- ▶ Make sure you have READ access to RACF profile "BPX.FILEATTR.PROGCTL" to be able to set the program control flags.

- ▶ The copy of the httpd.conf is needed to do the configuration since the utility expects a httpd.conf file.
- ▶ This httpd.conf will be used as a reference only.

WebSphere Application Server V1



● Run the WebAS Initialization

```
cd /usr/lpp/WebSphere/AppServer/config  
postinstall.sh
```

- ▶ Several prompts will appear
 - JAVA_ROOT: enter your JAVA root directory
/usr/lpp/java16/J1.1
 - Httpd.conf: reply with the "new" httpd.conf in /was
/was/httpd.conf
 - SE_root: the WebSphere Application Server Installation dir.
/usr/lpp/WebSphere/AppServer
- ▶ Now it does several links and copies several files
- ▶ Updates config files like jvm.properties
- ▶ Changes the ownership of the files to webadm:imweb
 - If you like another user ID / group --- change the shell script
- ▶ Executes WASconfig to update httpd.conf

WebSphere Application Server V1



- **Save the directory structure to an archive**

- ▶ Stay on the ../AppServer directory
- ▶ Keep superuser authority
- ▶ Store the directory and subdirectories into an archive file

```
pax -wv * > /was/WebAS11.pax
```

WebSphere Application Server V1



- **Restore the directory structure to an archive**

- ▶ Switch to the WebAS working directory

```
cd /was/apple
```

- ▶ keep superuser authority

- ▶ restore the data

```
pax -rv -pe < /was/WebAS11.pax
```

- ▶ Make sure that the Program Control Flags for the two WebSphere Application Server Modules are set

```
/was/apple/lib/libadpater.so
```

```
/was/apple/lib/libicsnativ.so
```

- requires RACF access READ to BPX.FILEATTR.PROGCTL in RACF FACILITY class

- use the ISPF - Shell

```
Option "a" --> Edit --> Extended attributes
```

```
--> Program Control Flag to 1
```

- or the OMVS - Shell

```
extattr +p *.so
```


WebSphere Application Server V1



● Customize the configuration files

▶ httpd.conf

```
#      Service directives
#
#  =====
#  *** OS/390 WebSphere Application Server 1.0      ***
#  =====

ServerInit /was/apple/lib/libadppter.so:AdapterInit /was/apple/properties
-- cont --> /server/servlet/servletservice/jvm.properties
ServerTerm /was/apple/lib/libadppter.so:AdapterExit

Service /*.jhtml      /was/apple/lib/libadppter.so:AdapterService
Service /*.shtml      /was/apple/lib/libadppter.so:AdapterService
Service /*.jsp         /was/apple/lib/libadppter.so:AdapterService
Service /servlet/*     /was/apple/lib/libadppter.so:AdapterService
```

© Copyright IBM Corporation, 1999

Roland Trauner trauner@us.ibm.com

- ▶ I would suggest to put these service directives after the Web Traffic Express (WTE) service directives (for the proxy server).
- ▶
- ▶ -- cont --> means that the statement should be in one line.
- ▶ It was needed to display the line content on this foil.

WebSphere Application Server V1



● Customize the configuration files

▶ httpd.conf --- cont.

```
#Pass /reports/*           /usr/lpp/internet/server_root/pub/reports/*
Pass /reports/*           /web/apple/reports/*
Pass /img-bin/*           /usr/lpp/internet/server_root/img-bin/*
# *** ADD NEW PASS RULES HERE ***

# *** WebSphere Application Server Samples PASS directives
Pass /IBMWebAS/samples/*   /was/apple/samples/*
Pass /IBMWebAS/Docs/*      /was/apple/system/admin/*
Pass /IBMWebAS/doc/*       /was/apple/doc/*
Pass /IBMWebAS/system/admin/* /was/apple/system/admin/*
Pass /IBMWebAS/*           /was/apple/web/*

# Pass /*                 /usr/lpp/internet/server_root/pub/*
Pass /Server/*            /usr/lpp/internet/server_root/pub/*
Pass /*                   /web/apple/pub/*
```

- ▶ These pass statements are defined in the way we suggest in the "Web Server Setup - Quick and Proper" presentation.

▶

WebSphere Application Server V1



● Customize the configuration files

▶ jvm.properties

- /WebAS/apple/properties/server/servlet/servletservice/

```
# @(#)jvm.properties.1.81 97/12/02
#
# Configuration properties for JVM and plugin dll start-up
#

# System Properties
IBMWebASVersion=1.0.0
# server.root=/usr/lpp/WebSphere/AppServer
server.root=/was/apple
server.name=servlet
server.description=IBMWebAS
java.compiler=

# NCF Properties
ncf.service.name=servletservice
ncf.service.class=com.ibm.servlet.service.SEServlet
ncf.plugin.classname=com.ibm.servlet.ServletSystem
```

WebSphere Application Server V1



● Customize the configuration files

▶ jvm.properties --- cont

```
#
# Enable native DLL plugin logging by setting 'ncf.native.logison'
# to 'true'. Change 'ncf.native.logfile' to the <fully-qualified >
# path of an alternate file location if desired.
#
# ncf.native.logison=false
ncf.native.logison=true
# ncf.native.logfile=/usr/lpp/WebSphere/AppServer/logs/native.log
ncf.native.logfile=/web/apple/logs/ncf_native.log
```

- ▶ point the log directives to your web server log structures

WebSphere Application Server V1



● Customize the configuration files

▶ jvm.properties --- cont

```
#
# Enable JVM logging by setting 'ncf.jvm.stdoutlog.enabled'
# to true. Change 'ncf.jvm.stdoutlog.file' to 'false' to write
# to a Java debugging console or 'true' for output to a log file.
# Uncomment the line for 'ncf.jvm.stdoutlog.popup', thus setting it
# to '2' to display the combined ResourceUsage/EnableTrace/Console
# popup. Otherwise, just the console popup is displayed.
# Change 'ncf.jvm.stdoutlog.filename' to the <fully-qualified>
# path of an alternative file location if desired.
#
# ncf.jvm.stdoutlog.enabled=false
# #ncf.jvm.stdoutlog.popup=2
# ncf.jvm.stdoutlog.file=true
# ncf.jvm.stdoutlog.filename=/usr/lpp/WebSphere/AppServer/logs/ncf.log
#
ncf.jvm.stdoutlog.enabled=true
ncf.jvm.stdoutlog.popup=2
ncf.jvm.stdoutlog.file=true
ncf.jvm.stdoutlog.filename=/web/apple/logs/ncf_jvm.log
```

WebSphere Application Server V1



● Customize the configuration files

▶ jvm.properties --- cont

```
# NCF - Admin Service Properties for BasicNCFConfig Applet
# ncf.jvm.classpath=/usr/lpp/WebSphere/AppServer/lib/ibmwebas.jar:
-- cont --> /usr/lpp/WebSphere/AppServer/lib/jst.jar:
-- cont --> /usr/lpp/WebSphere/AppServer/lib/jsdk.jar:
-- cont --> /usr/lpp/WebSphere/AppServer/lib/x509v1.jar:
-- cont --> /usr/lpp/WebSphere/AppServer/lib:
-- cont --> /usr/lpp/WebSphere/AppServer/web/admin/classes/seadmin.jar:
-- cont --> /usr/lpp/WebSphere/AppServer/web/classes:
-- cont --> /usr/lpp/java16/J1.1/lib/classes.zip
# ncf.jvm.libpath=/usr/lpp/java16/J1.1/lib:
-- cont --> /usr/lpp/java16/J1.1/lib/mvs/native_threads:
-- cont --> /usr/lpp/WebSphere/AppServer/lib:/usr/lib
ncf.jvm.classpath=/was/apple/lib/ibmwebas.jar:
-- cont --> /was/apple/lib/jst.jar:/was/apple/lib/jsdk.jar:
-- cont --> /was/apple/lib/x509v1.jar:/was/apple/lib:
-- cont --> /was/apple/web/admin/classes/seadmin.jar:
-- cont --> /was/apple/web/classes:/usr/lpp/java16/J1.1/lib/classes.zip
ncf.jvm.libpath=/usr/lpp/java16/J1.1/lib:
-- cont --> /usr/lpp/java16/J1.1/lib/mvs/native_threads:
-- cont --> /was/apple/lib:/usr/lib
ncf.jvm.path=/usr/lpp/java16/J1.1/bin
ncf.jvm.use.system.classpath=false
```

© Copyright IBM Corporation, 1999

Roland Trauner trauner@us.ibm.com

- ▶ Make sure that the java classes.zip is the last entry in the classpath - else you have severe performance problems.
- ▶ Check the PSPBucket as well.
- ▶
- ▶ ncf.jvm.classpath can become a very long statement. There is no way to concatenate it.

WebSphere Application Server V1



● Customize the configuration files

▶ jvm.properties --- cont

```
# Max Java Heap Size
ncf.jvm.mx=67108864

# Properties for Domino Go
# ncf.native.httpd.cnf.path=/WebAS/httpd.conf
ncf.native.httpd.cnf.path=/web/apple/httpd.conf

# OS/390 WebSphereAS 1.0 Only!
#
# The ncf.jvm.threads.max property is used to increase the number of threads
# that the JVM is allowed to create for threaded servlets, chaining servlets,
# or filtering servlets. If any of these types of servlets are being executed,
# this property will need to be set to accomodate the threading needs.
#ncf.jvm.threads.max=5
ncf.jvm.threads.max=160
```

WebSphere Application Server V1



● Customize the configuration files

▶ jvm.properties --- cont

```
# OS/390 WebSphereAS 1.0 Only!
#
# Define the OS/390 OE native log debug level. The
# following define the current debug levels:
#
#   Level 0
#   -----
#   - OS/390 OE specific tracing off.
#
#   Level 1
#   -----
#   - Trace messages in JNI wrappers.
#   - Trace messages in the ICS native library for JNI
#     calls made during servlet processing.
#   - Trace messages in OE Data Conversion Utility
#     routines.
#
#   Level 2 (Includes Level 1.)
#   -----
#   - Trace messages in the ICS native library for JNI
#     calls made to convertgetBytes_AtoE().
#
#ncf.native.os390.debug=0
ncf.native.os390.debug=0
```


WebSphere Application Server V1



- **WebAS Manager Port Number**

- ▶ WebAS provides its own application to manage WebAS and the servlets, JSPs etc.
- ▶ Per default the port is 9090 and you need a JAVA enabled browser.

- **If you use the WebAS Manager**

- ▶ to manage the WebAS configuration, you need to connect it on port 9090. Then you may change it and after restart it'll be another port.

- **Just edit admin_port.properties manually**

- ▶ It is easier to change the port number using a OS/390 editor

```
- /was/apple/properties/server/servlet/adminservice
- edit admin_port.properties
  # @(#)admin_port.properties.1.1 97/09/11
  # Administration service port for this server
  #
  # endpoint.main.port=9090
  endpoint.main.port=9011
```

- ▶ ServletExpress also used 9090
- ▶
- ▶ Changes in WebAS are not recognized by the WebServer restart command (f webserver,appl=-restart). WebServer needs to be stopped and started.

WebSphere Application Server V1



- **Configuration is ready now**
- **Another Instance of WebAS ?**
 - ▶ Simply repeat all the steps
 - ▶ Assign a different portnumber at admin_port.properties

WebSphere Application Server V1



- **Start the Web Server to start WebAS**

- ▶ During the Web Server startup now the WebAS environment will also be initialized.
- ▶ This needs some more resources now and the startup duration will be longer --- much depending on the machine model.
- ▶ Web Server is ready when issuing the Server Ready message
`IMW3536I SA 1761607710 0.0.0.0:80 * * READY`

WebSphere Application Server V1



● Verify a Successful Startup

- ▶ Check -vv trace. you will find the following entries:

```
Adding... API class "serverinit"
Adding... module "/was/apple/lib/libadppter.so" to API class "serverinit"
Adding... function "AdapterInit" to API class "serverinit"
Adding... API class "serverterm"
Adding... module "/was/apple/lib/libadppter.so" to API class "serverterm"
Adding... function "AdapterExit" to API class "serverterm"
Service.... "/*.jhtml" --> "/was/apple/lib/libadppter.so:AdapterService"
Adding... API class "service"
Adding... module "/was/apple/lib/libadppter.so" to API class "service"
Adding... function "AdapterService" to API class "service"
Service.... "/*.jsp" --> "/was/apple/lib/libadppter.so:AdapterService"
Adding... API class "service"
Adding... module "/was/apple/lib/libadppter.so" to API class "service"
Adding... function "AdapterService" to API class "service"
Service.... "/servlet/*" --> "/was/apple/lib/libadppter.so:AdapterService"
Adding... API class "service"
Adding... module "/was/apple/lib/libadppter.so" to API class "service"
Adding... function "AdapterService" to API class "service"
Service.... "/cgi-bin/htimage*" --> "INTERNAL:HTImage*"
```

WebSphere Application Server V1



● Verify a Successful Startup --- cont.

```
API... Trying to load shared library "/was/apple/lib/libadppter.so"
AppEnvFilter... Loading dll "/was/apple/lib/libadppter.so" due to default
API... Successful loading shared library "/was/apple/lib/libadppter.so"
API... Trying to get fn pointer "AdapterInit" from module "/was/apple/lib/libadppter.so"
API... Successful getting fn pointer "AdapterInit"
API... Trying to load shared library "/was/apple/lib/libadppter.so"
AppEnvFilter... Loading dll "/was/apple/lib/libadppter.so" due to default
API... Successful loading shared library "/was/apple/lib/libadppter.so"
API... Trying to get fn pointer "AdapterExit" from module "/was/apple/lib/libadppter.so"
API... Successful getting fn pointer "AdapterExit"
API... Trying to load shared library "/was/apple/lib/libadppter.so"
AppEnvFilter... Loading dll "/was/apple/lib/libadppter.so" due to default
API... Successful loading shared library "/was/apple/lib/libadppter.so"
API... Trying to get fn pointer "AdapterService" from module "/was/apple/lib/libadppter.so"
API... Successful getting fn pointer "AdapterService"
API... Trying to load shared library "/was/apple/lib/libadppter.so"
AppEnvFilter... Loading dll "/was/apple/lib/libadppter.so" due to default
API... Successful loading shared library "/was/apple/lib/libadppter.so"
API... Trying to get fn pointer "AdapterService" from module "/was/apple/lib/libadppter.so"
API... Successful getting fn pointer "AdapterService"
API... Trying to load shared library "/was/apple/lib/libadppter.so"
AppEnvFilter... Loading dll "/was/apple/lib/libadppter.so" due to default
API... Successful loading shared library "/was/apple/lib/libadppter.so"
API... Trying to get fn pointer "AdapterService" from module "/was/apple/lib/libadppter.so"
API... Successful getting fn pointer "AdapterService"
```

WebSphere Application Server V1



● Verify a Successful Startup --- cont.

```
APIClassExec Looking up API class "serverinit"
GWAPI: Create a new API data structure
APIClassExec Calling server-init function "AdapterInit"
GWAPI: HTTPD_extract() called
GWAPI: HTTPD_extract() args..... name= INIT_STRING ; name size= 11
GWAPI: HTTPD_extract() args..... buffer= 0x9b58a98 ; buffer size= 1023
GWAPI: HTTPD_extract()... Looking up server and CGI variables
GWAPI: HTTPD_extract()... successful with value=
"/was/apple/properties/server/servletservice/jvm.properties"
GWAPI: HTTPD_extract() called
GWAPI: HTTPD_extract() args..... name= SERVER_SOFTWARE ; name size= 15
GWAPI: HTTPD_extract() args..... buffer= 0xa63be50 ; buffer size= 255
GWAPI: HTTPD_extract()... Looking up server and CGI variables
GWAPI: HTTPD_extract()... successful with value= "Lotus Domino Go Webserver - North
American Edition for OS/390/V5R0M0"
GWAPI: HTTPD_extract() called
GWAPI: HTTPD_extract() args..... name= SERVER_NAME ; name size= 11
GWAPI: HTTPD_extract() args..... buffer= 0xa63be50 ; buffer size= 255
GWAPI: HTTPD_extract()... Looking up server and CGI variables
GWAPI: HTTPD_extract()... successful with value= "wtsc58oe.itso.ibm.com"
GWAPI: HTTPD_extract() called
GWAPI: HTTPD_extract() args..... name= SERVER_PORT ; name size= 11
GWAPI: HTTPD_extract() args..... buffer= 0x9b58f68 ; buffer size= 9
GWAPI: HTTPD_extract()... Looking up server and CGI variables
GWAPI: HTTPD_extract()... successful with value= "98"
GWAPI: HTTPD_log_error() called
GWAPI: HTTPD_log_error() args..... value= IBM WebSphere Application Server native plugin
initialization went OK :-) ; value size= 72
GWAPI: HTTPD_log_error()... successful
```

WebSphere Application Server V1



● Access WebAS Manager

- ▶ To access the WebAS Manager use a JAVA enabled browser and access your web server on port 9090, or the appropriate port you changed to in `admin_port.properties`
 - `http://www.the-apple.com:9090`
 - Java enabled browser means i.e. Netscape Communicator 4.03 with the JDK 1.1 patch from <http://help.netscape.com/filelib.html> or a later version
 - MS IE 4.0 or Sun HotJava 1.1 can also be used
- ▶ SE Manager prompts for user ID and password which will be `admin/admin` for the first time.
- ▶ You may change it then. Password is then stored at `/was/apple/realms/data/adminRealm/keyfile`
 - The password is encrypted in that file



More Hints

● JAVA Installation

- ▶ WebAS depends on a proper JAVA Installation
- ▶ To retain a "proper" environment for the web server, make sure that the modules on `/usr/lpp/java16/J1.1/lib/mvs/native_threads` have the extended program-control attribute set.
 - Make sure you are permitted to set the attributes

```
RDEFINE FACILITY BPX.FILEATTR.PROGCTL UACC(NONE)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY)
ID(TRAUNER) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```
 - Use the OMVS shell, switch to superuser (SU)

```
cd /usr/lpp/java16/J1.1/lib/mvs/native_threads
extattr +p *.*
```
 - Check the attributes

```
ls -E
```


More Hints



- **Check the web**

- ▶ <http://www.software.ibm.com/webserver/appserv/>



© Copyright IBM Corporation, 1999

Roland Trauner trauner@us.ibm.com