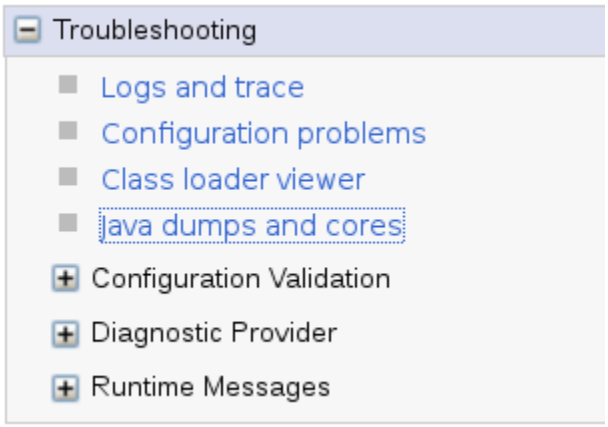
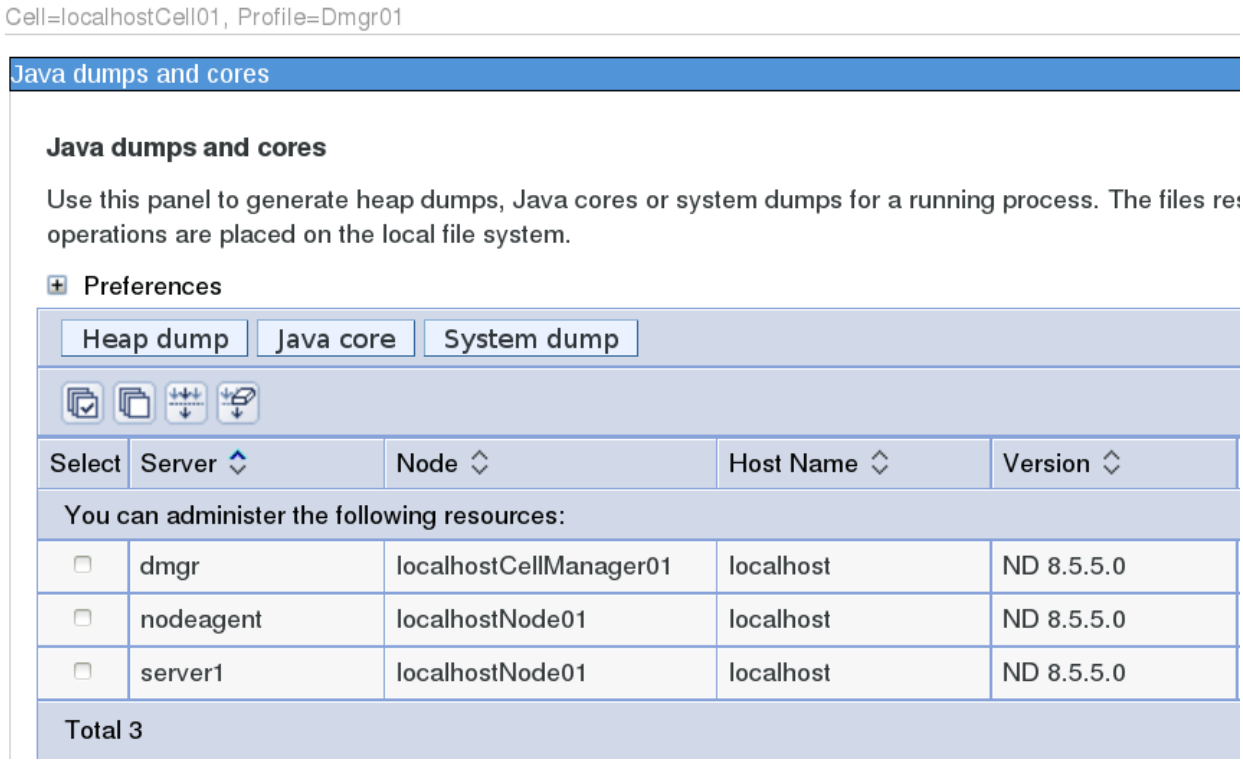
Generate Data

1. Using WebSphere Administrative Console

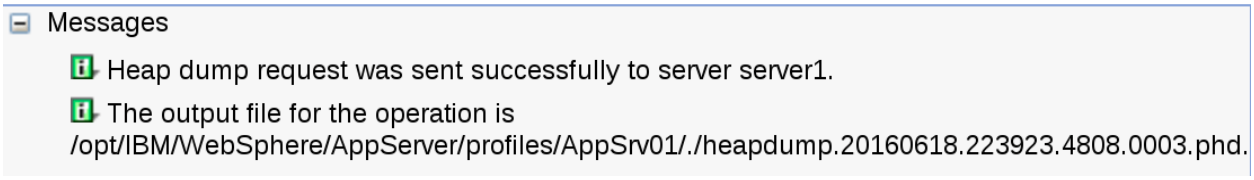
- Login into DMGR Console
- Navigate to Troubleshooting at left side
- Click on java dumps and cores



- Select the JVM from the list and click on the one you wish to generate



Dump & Core will be generated under WebSphere profile location. You will get the absolute path under the message.



2. Using Command Line

- Login into WAS Server
- Go to profile and bin folder
- Execute wsadmin.sh file

```
[root@localhost bin]# ./wsadmin.sh
WASX7209I: Connected to process "dmgr" on node
localhostCellManager01 using SOAP connector; The type of
process is: DeploymentManager
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

- Set JVM name in a variable

```
set jvm [$AdminControl completeObjectName
type=JVM,process=server1,*]
```

Note: server1 is for example. Change this to your actual JVM name.

```
wsadmin>set jvm [$AdminControl completeObjectName
type=JVM,process=server1,*]
WebSphere:name=JVM,process=server1,platform=proxy,node=localhos
tNode01,j2eeType=JVM,J2EEServer=server1,version=8.5.5.0,type=JV
M,mbeanIdentifier=JVM,cell=localhostCell01,spec=1.0
wsadmin>
```

To generate heap dump

- Execute following

```
$AdminControl invoke $jvm generateHeapDump
```

This will generate heap dump and give you the path where it's available.

```
wsadmin>$AdminControl invoke $jvm generateHeapDump
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/./heapdump.20160
618.225441.4808.0006.phd
wsadmin>
```

To generate Java Core

```
$AdminControl invoke $jvm dumpThreads
```

This will generate thread dump and will be available under profile path.

You see getting dumps are easy and analyzing them is always interesting.

Part 2

Taking Thread Dumps in WebSphere

It's recommended to take multiple thread dumps in 5-8 seconds. You can use any of the following methods to generate it.

1. Using wsadmin.sh

- Login into WAS Server
- Go to profile and bin folder
- Execute wsadmin.sh file

```
[root@localhost bin]# ./wsadmin.sh
WASX7209I: Connected to process "dmgr" on node
localhostCellManager01 using SOAP connector; The type of
process is: DeploymentManager
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

- Set JVM name in a variable

```
set jvm [$AdminControl completeObjectName
type=JVM,process=server1,*]
```

Note: server1 is, for example. Change this to your actual JVM name.

```
wsadmin>set jvm [$AdminControl completeObjectName
type=JVM,process=server1,*]
WebSphere:name=JVM,process=server1,platform=proxy,node=localhos
tNode01,j2eeType=JVM,J2EEServer=server1,version=8.5.5.0,type=JV
M,mbeanIdentifier=JVM,cell=localhostCell01,spec=1.0
wsadmin>$AdminControl invoke $jvm dumpThreads
```

This will generate a thread dump and will be available under the profile path.

2. Using kill

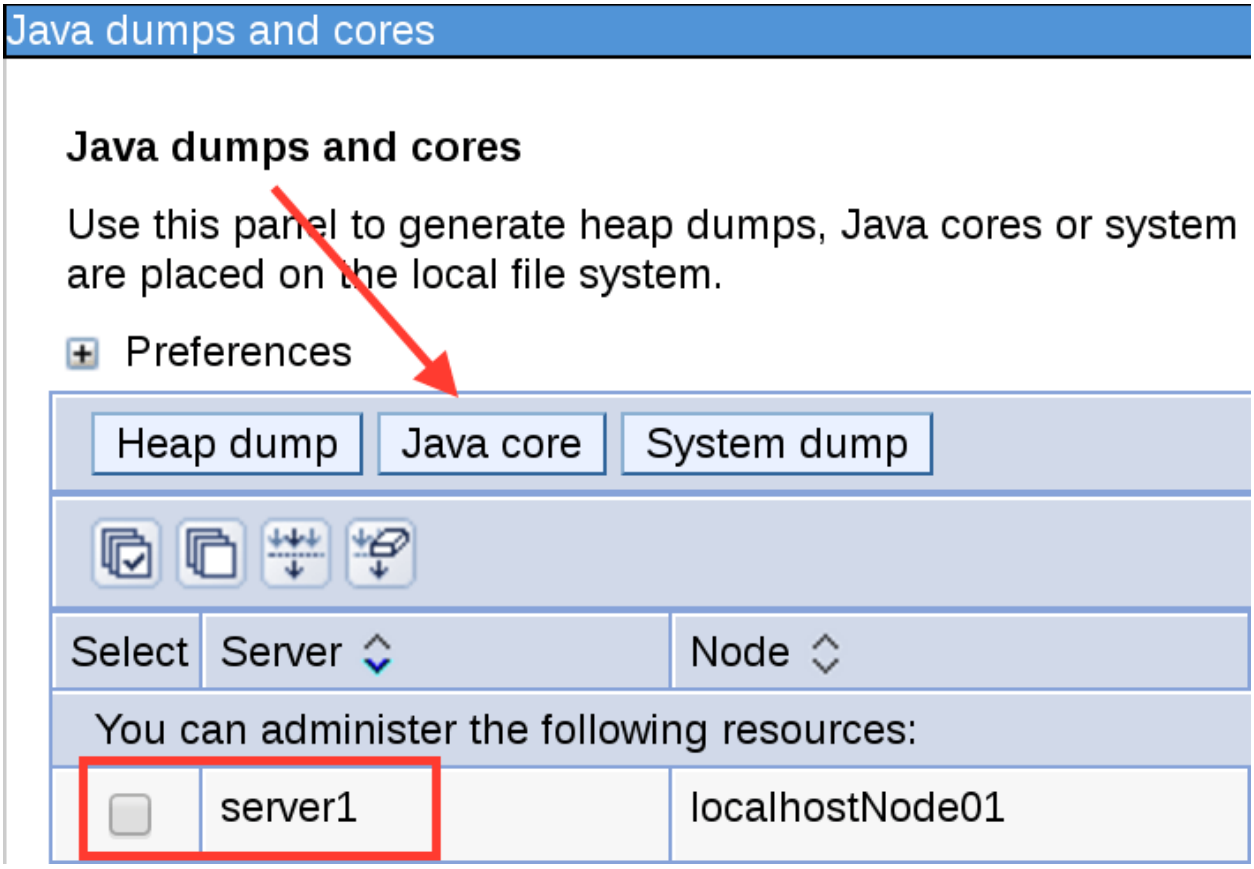
- Find the JVM process ID using ps command
- Execute kill -3 $PID

```
# kill -3 $PID
```

You will find the dump in the profile path.

3. Using the WebSphere Administrative Console

- Login into DMGR Console
- Navigate to Troubleshooting at the left side
- Select the JVM and click on “Java core.”



Once you have the thread dump, you can use the following tools to analyze them.

Analyzing WebSphere Thread Dumps

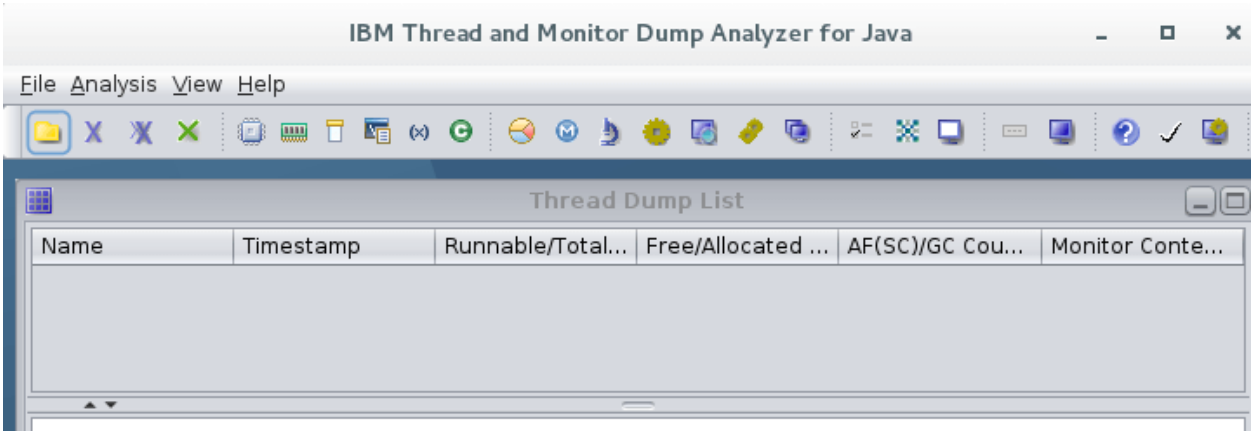
Analyzing dumps are always challenging, and following FREE tools will help you.

IBM Thread Dump Analyzer (TDA)

Using IBM TDA, you can identify hangs, deadlocks, and bottlenecks in Java threads.

You can download it from here (https://www.ibm.com/support/pages/ibm-thread-and-monitor-dump-analyzer-java-tmda) It's just the jar file, and once downloaded, you can execute the following to start the GUI.

```
java -jar jca457.jar
```

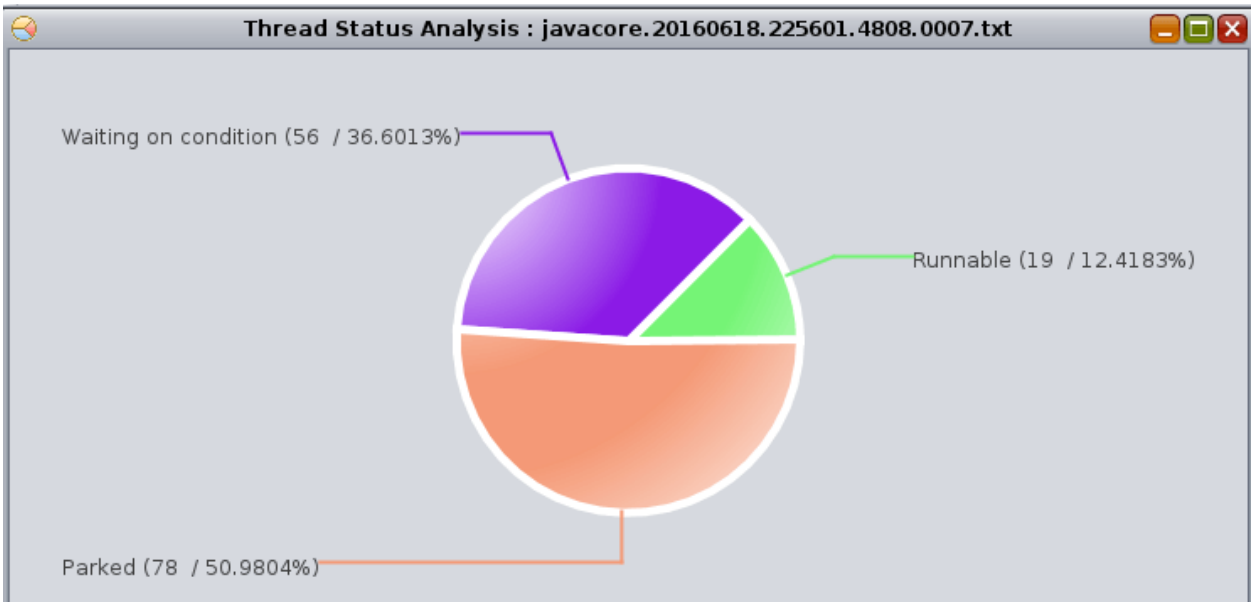


- Click on File >> Open Thread Dumps
- Browse the folder and select the dump you wish to analyze

It will take a few seconds, and you are all set to do the following analysis

- CPU usage
- Memory segment
- User process resources limit
- Command-line argument
- Environment variable
- Shared class cache
- Native memory
- Thread status
- Method

All are available under the "**Analysis**" menu. A quick look at thread status analysis



So go through the analysis you are looking for have fun with IBM TDA.

So now, you should be able to generate and analyze the thread dumps for application troubleshooting. There is more analyzer explained here.

Part 3

Taking Thread Dumps Automatically When JVM Stops

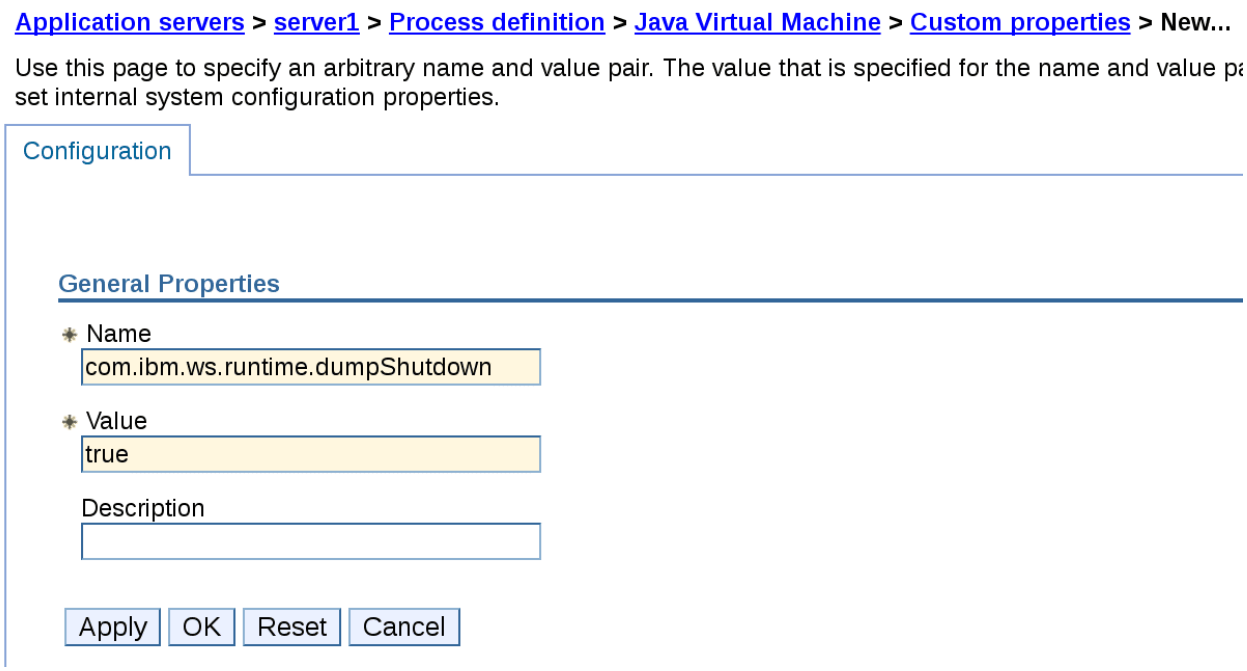
There might be a scenario where you would like to take the thread dump whenever JVM stop. This will be useful when you are not sure whom or how JVM is getting stopped.

To configure this;

- Login to WebSphere Administrative Console
- Navigate to Servers >> Server Types >> WebSphere application servers
- Go inside the JVM and expand “Java and Process Management” under Server Infrastructure
- Click Process definition
- Click “Java Virtual Machine” under Additional Properties >> Custom properties
- Click New and enter Name as

```
com.ibm.ws.runtime.dumpShutdown
```

- Value as **true**
- Click OK



- Review and Save the configuration

Now, heap dump will be generated whenever JVM stop.

I stopped the JVM and could see the following javacore file on WAS server under profile path.

```
-rw-r--r--. 1 root root 3179348 Nov 28 22:49
javacore.20161128.224916.4926.0001.txt
-rw-r--r--. 1 root root 3124208 Nov 28 22:49
javacore.20161128.224924.4926.0002.txt
```

Taking Thread Dump When Hung

If you are having hung thread issue, and currently you take the dump manually, then you can automate this.

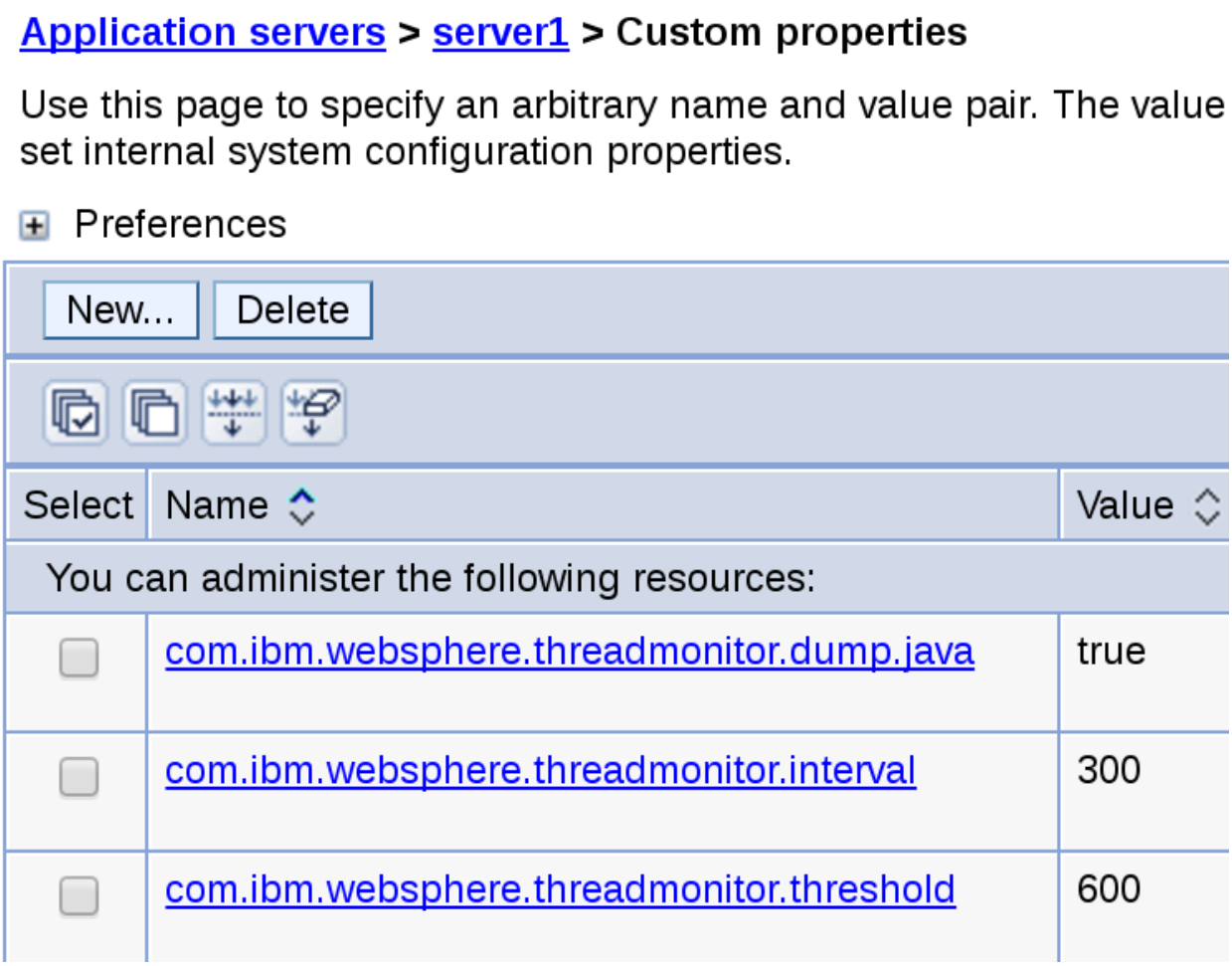
By doing basic configuration, WebSphere will generate the dump whenever a thread is hung for configured timed.

- Login to WAS Console and go inside respective JVM
- Expand “Administration” under Server Infrastructure and click Custom Properties
- Add the following three properties

com.ibm.websphere.threadmonitor.threshold – Generate dump when a thread is hung for configured time. Value is in seconds.

com.ibm.websphere.threadmonitor.interval – How frequently thread monitor should check for the hung threads. The value in seconds.

com.ibm.websphere.threadmonitor.dump.java – Generate dump when detected. Value is true.



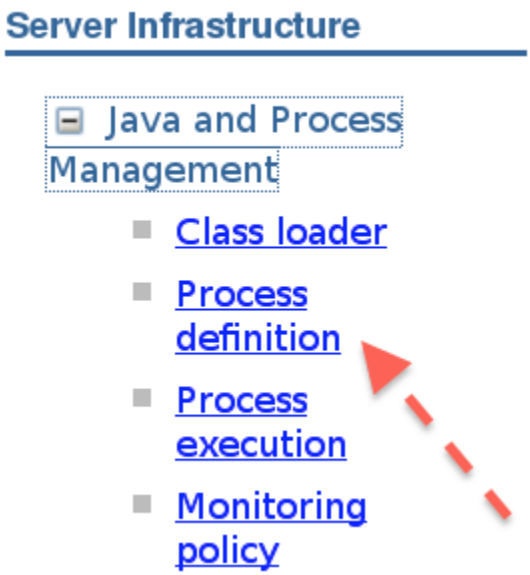
Above configuration will generate dump whenever a thread is hung for **600 seconds**, and WebSphere will monitor the thread every **300 seconds**.

Once you have the dump, you can analyze and fix the application issue.

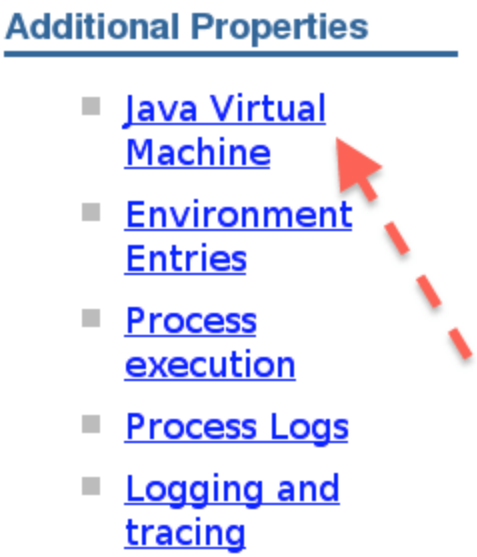
Part 4

Modify Memory Heap Size in JVM

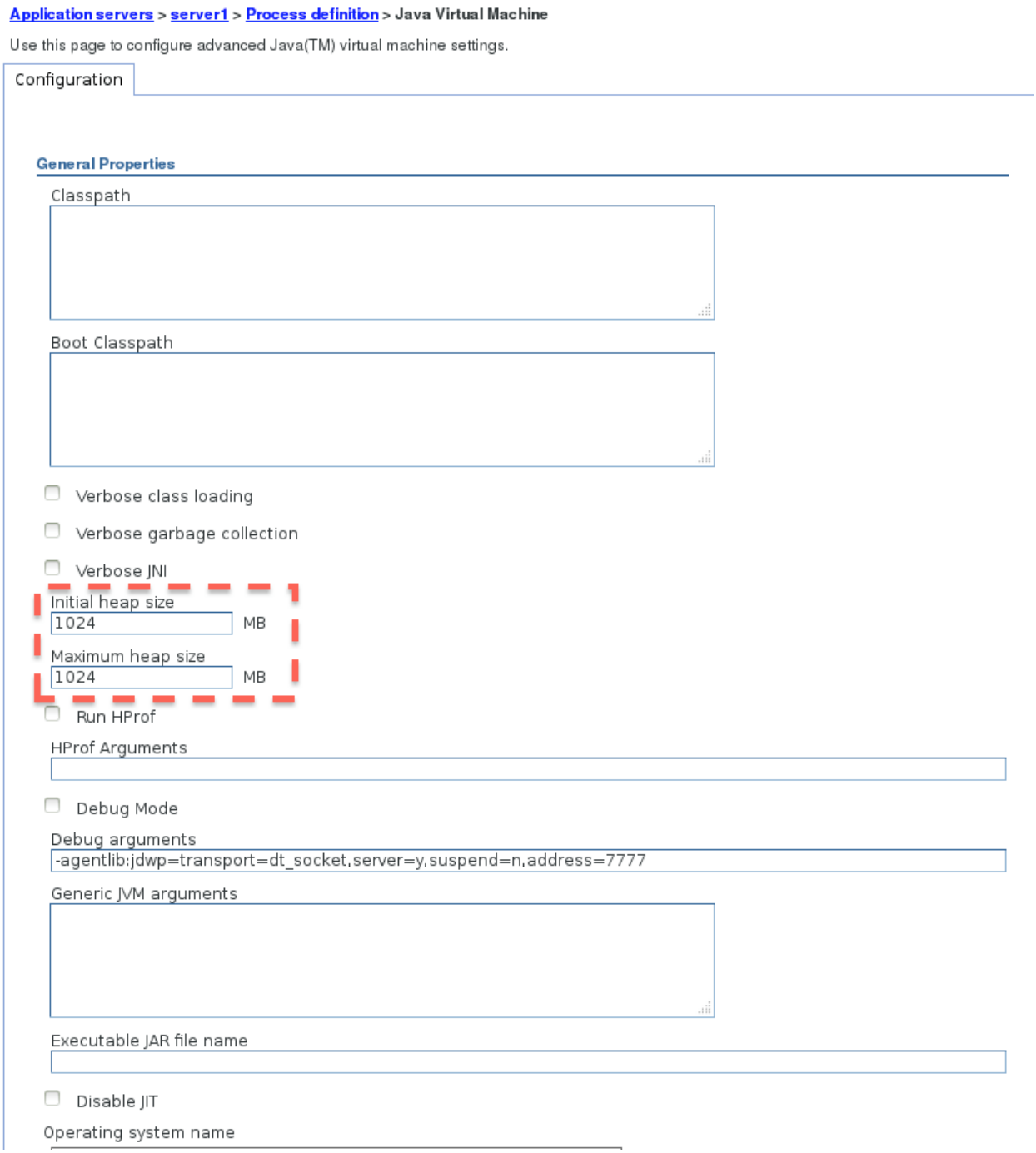
- Login into DMGR
- Go to Server Types >> WebSphere application servers
- Click on desired JVM
- Click on Process definition under “Java and Process Management.”



- Click on Java Virtual Machine under “Additional Properties”



- Enter minimum and maximum heap size (as shown below for configuring 1 GB as minimum and maximum size)



- Click on Apply and Ok
- Restart the JVM to get this effective

Modify Memory Heap Size in Nodeagent

- Login into DMGR
- Go to System administration >> Node agents

- Click on nodeagent
- Expand Java and Process Management and click on Process definition
- Click on Java Virtual Machine under “Additional Properties”
- Enter the minimum and maximum heap size
- Restart the Nodeagent to get this effective

Modify Memory Heap Size in DMGR

- Login into DMGR
- Go to System administration >> Deployment manager
- Expand Java and Process Management and click on Process definition
- Enter the minimum and maximum heap size
- Restart the DMGR to get this effective

Tip: Memory heap size is stored in server.xml of particular JVM/Nodeagent/DMGR. If you are not sure the location, use find command on Linux.

Part 5

What is WebSphere Performance Tuning Toolkit?

It's important to understand what is PTT and how it can help you.

WAS PTT is an Eclipse-based tool which collects the data from WebSphere and provides you nice statistics/graph to find the bottlenecks so you can tune the application for the optimal performance.

WebSphere PTT collect the data from PMI (performance monitoring interface) through JMX. To **summarize** it does four main things.

- Collect the data
- Transform the data
- Intelligent alarm
- Let you configure auto actions

PTT is real-time monitoring and analysis platform and got easy to use interface. This is a standalone tool so you may not need to install anything additionally on WebSphere Application Server. I hope now you have an idea how it can fit in your WebSphere environment.

PTT has some **predefined rules** to alert or take actions when threshold reach the configured level.

For ex:

- CPU usage
- Heap memory usage
- Thread pool
- Connection timeout
- Thread waiting
- Hung thread
- Transaction timeout/rolled back
- Generate thread/heap dump

The good thing is you can always create your own rule so **you control** how the tool should behave.

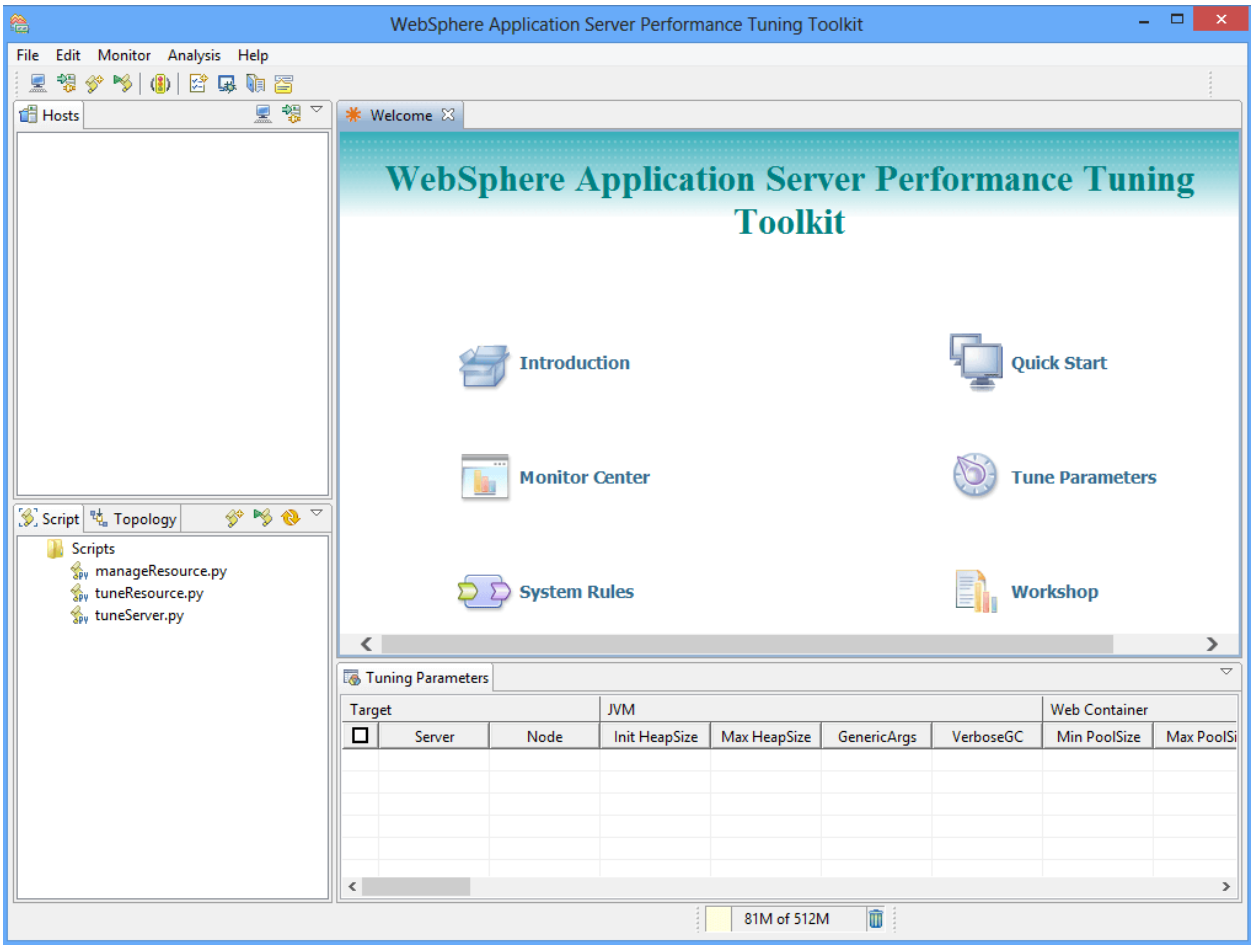
Download WebSphere PTT

PTT is available for Windows & Linux so download the one you need from the following URL

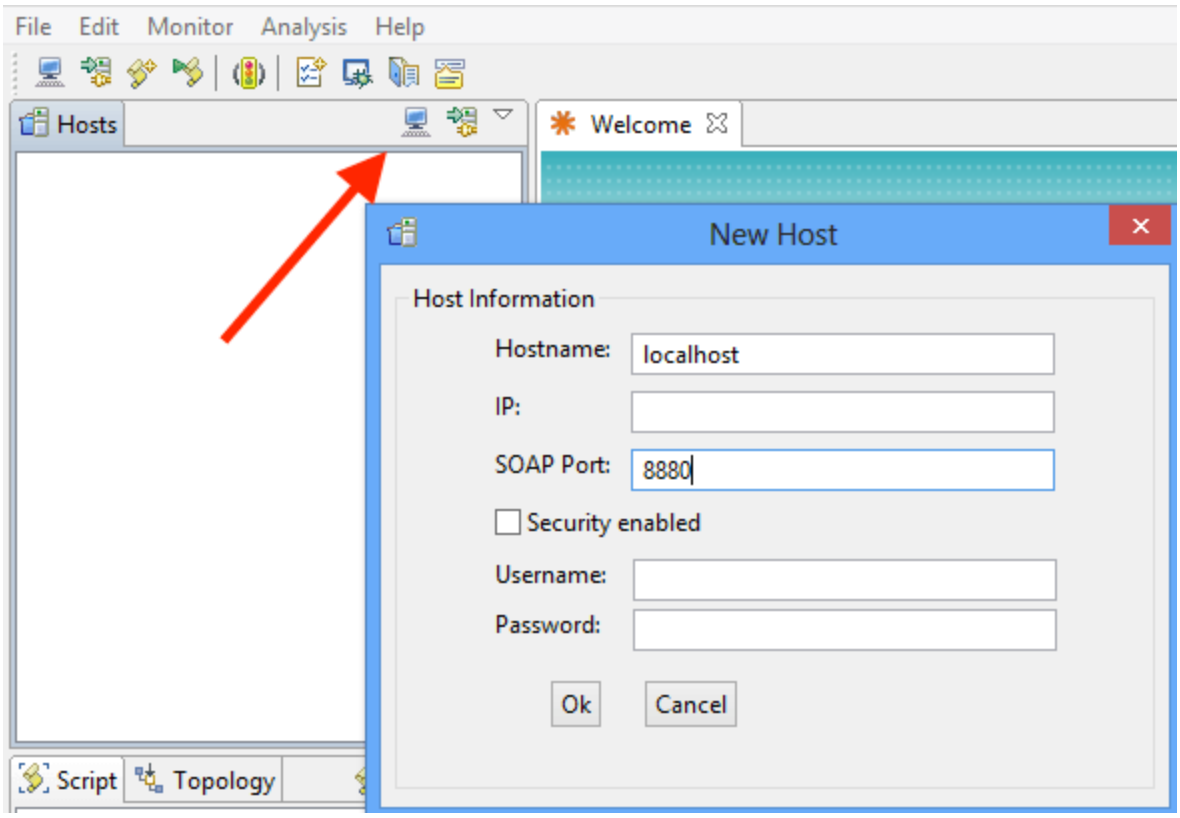
http://www.ibm.com/developerworks/websphere/downloads/performtuning.html#download

It's more than 200 MB file so may take few minutes based on your Internet speed. Once downloaded, extract the zip file.

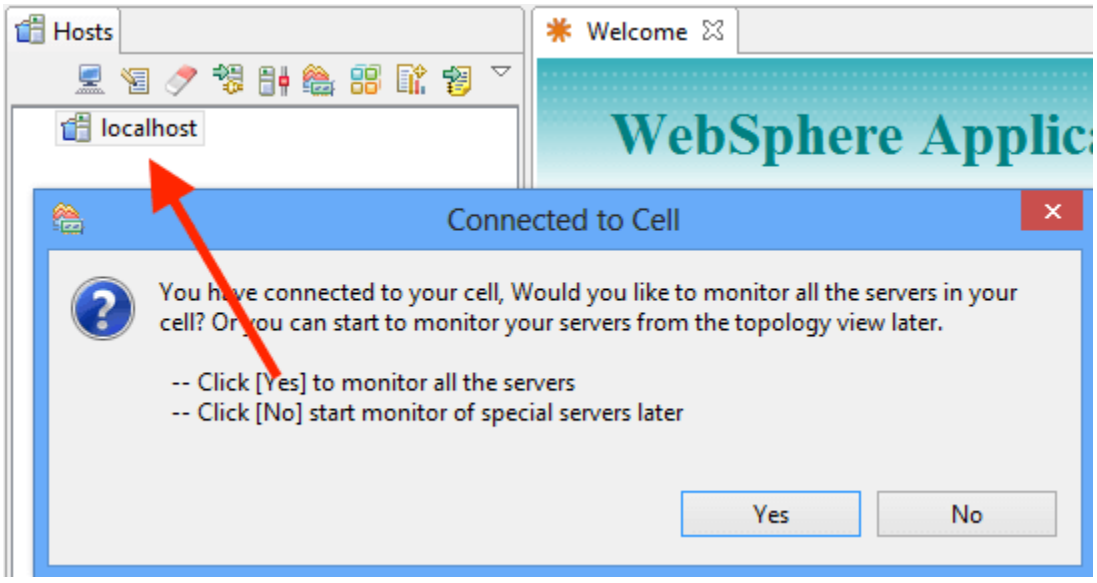
- It will create a new folder – PerformanceTuningToolkit2_linux32 and inside you will have executable file as "PerfTuningToolkit"
- Double click to open the WAS Performance Tuning Toolkit



- Let's add one JVM now. Click on "add new host" icon and enter the DMGR host/IP and SOAP port number.

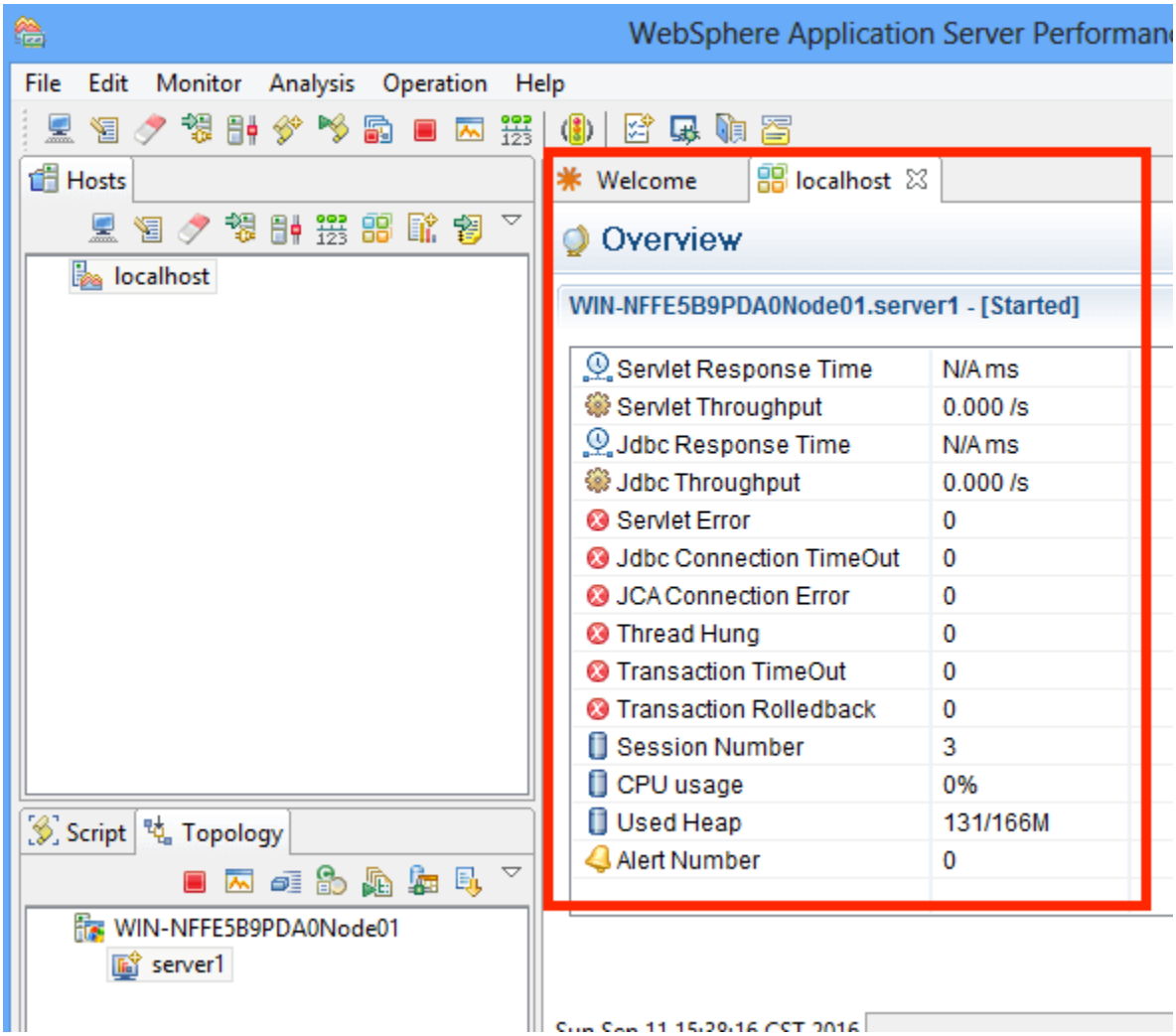


- Double click on newly added “host” to connect. It will prompt if you want to monitor all servers or a specific one.



Note: if you just want to monitor just JVM then provide JVM host/port instead of DMGR.

- Once connected, you should see the overview windows with some of the high-level metrics.



- CPU Usage
- Thread hung
- Transaction TimeOut
- JDBC response time
- Servlet response time
- Used heap

So now you have successfully connected your WebSphere environment to Performance Tuning Toolkit and its up-to-you to monitor and analyze the various WAS metrics.

Let's see some of the useful metrics

Generate overall performance report

Would be helpful if you would like to have overall performance report of WebSphere. You can generate report by going to Monitor >> Generate Report

Performance Report

1. Proxy Server

Node	Server	Cache Hit	Cache Hit (Validated)	Failed Request	Thread Hang	Avg Cpu Usage(%)	Avg Used Heap(M)	ReponseTim (ms)	Throughput
Average									

2. Application Server

Node	Server	Servlet Error	Transaction Rolledback	Connection Timeout	Thread Hang	Avg Cpu Usage(%)	Avg Used Heap(M)	ReponseTim (ms)	Throughput
WIN-NFFE5B9PD	server1	0	0	0	0	9	122	NaN	0
Average		0.000	0.000	0.000	0.000	9.000	122.000	NaN	0.000

3. Transaction

Node	Server	Transaction Rolledback	Transaction Rollback (%)	Local Transaction Timeout	Global Transaction Timeout	ReponseTime (ms)	Throughput
WIN-NFFE5B9PDA0I	server1	0	NaN	0	0	NaN	0.000
Average		0.000	NaN	0.000	0.000	NaN	0.000

4. Session

Node	Server	Application	Affinity Break Count	Timeout Invalidation Count	Live Count	Avg Life Time
WIN-NFFE5B9PDA0N	server1	isclite#wasportlet.	0	1	0	1858000
	server1	isclite#adminredir	0	0	0	0
	server1	ibmasyncrsp#ibm	0	0	0	0
	server1	ivtApp#ivt_app.wa	0	0	0	0
	server1	DefaultApplicatior	0	0	0	0
	server1	isclite#iehs.war	0	0	0	0
	server1	isclite#ISCAdminF	0	1	0	1852178
	server1	isclite#isclite.war	0	1	0	3471865
	server1	isclite#WIMPortle	0	0	0	0
	server1	filetransferSecure	0	0	0	0

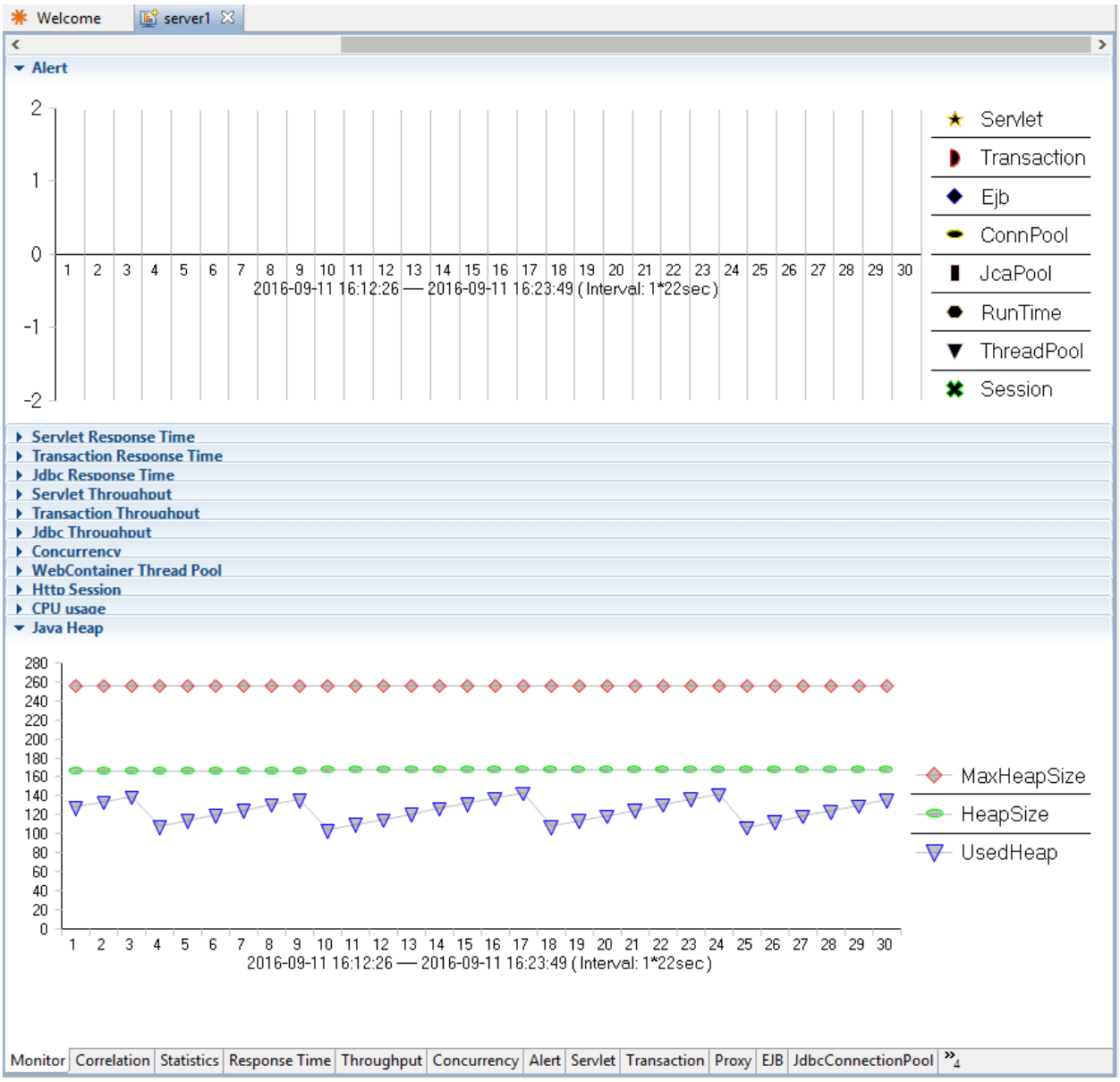
5. ThreadPool

Node	Server	Thread Pool	Declared Hang Thread	Clared Hang Thread	Max Pool Size	Min Pool Size	Pool Size	Used Pool
WIN-NFFE5B9PDA	server1	HAManager.t	0	0	2	2	2	0
		SIBFAPThrea	0	0	50	4	0	0
		WebContaine	0	0	50	50	7	1
		MessageListe	0	0	50	10	0	0
		SoapConnect	0	0	5	3	3	1

You have the option to save the report in PDF or HTML format.

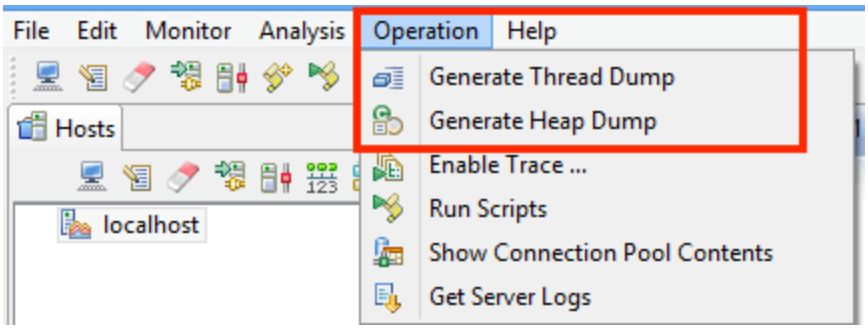
Monitor WebSphere Environment

Monitor various metrics under single window by navigating to Monitor >> Open monitor page



Generate Thread and Heap Dump

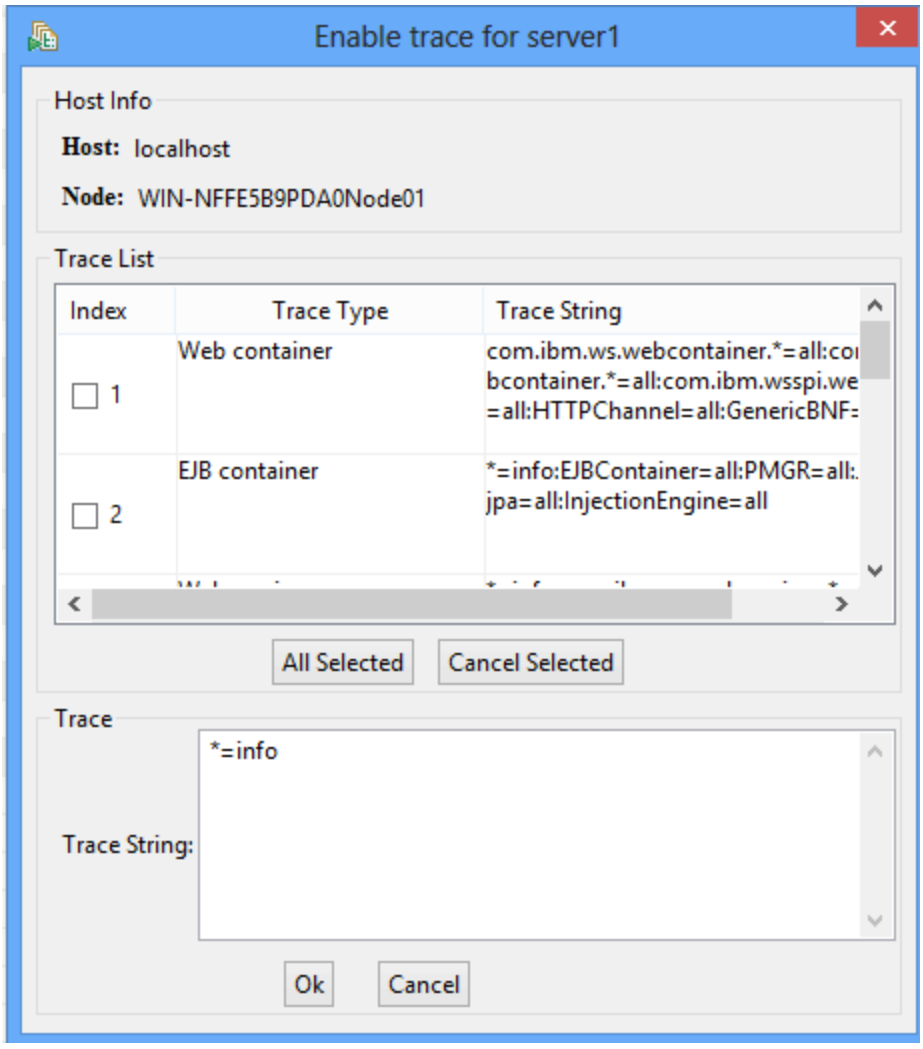
You don’t need to get into WAS server to generate one instead, navigate to Operation >> Generate Thread/heap Dump



Enable Trace

You can enable trace for the following by going to Operation >> Enable Trace

- Web/EJB container
- Web services
- Security
- Database connection pool
- Connection leak
- Classloader
- Messaging



WebSphere PPT can really be helpful to **tune the performance related issue** and especially when your application is in project phase and want to play around with the configuration and see what works the best for you.

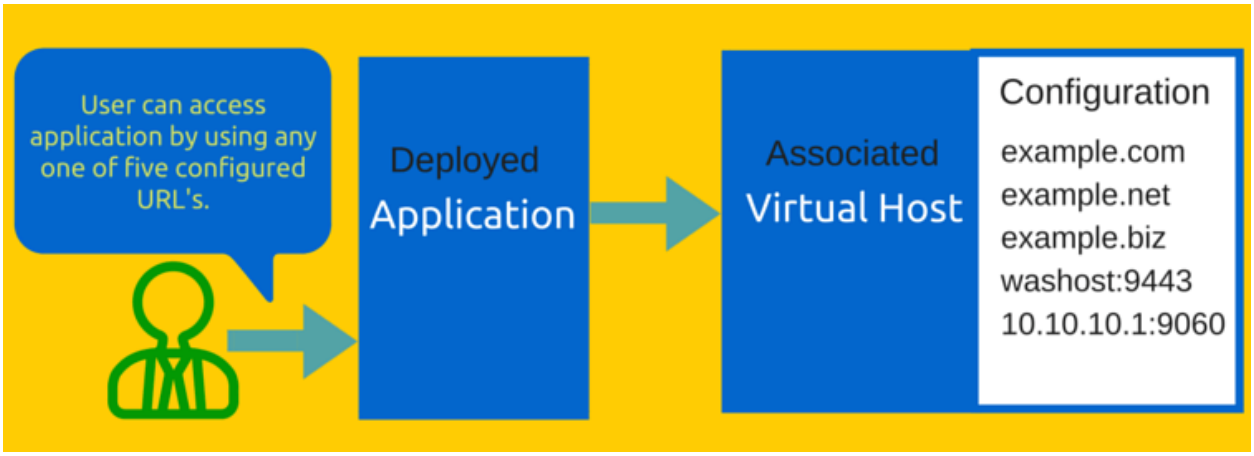
I hope this gives you an idea how to use **WAS PTT** and what you can do with it.

Part 6

Virtual host

Virtual host means allowing multiple URL's (ex – example.com, example.net, example.biz, washost:9443, 10.10.10.1:9060) on a single application through an associated virtual host. This can be either IP-based or name-based. A virtual host configuration is done through WAS administrative console.

Using a virtual host, you can also define the MIME types. In default WAS installation, you will have two virtual host **admin_host** & **default_host**, which you may use or can create the new one for your application. Let's take a look at below illustration of a virtual host.

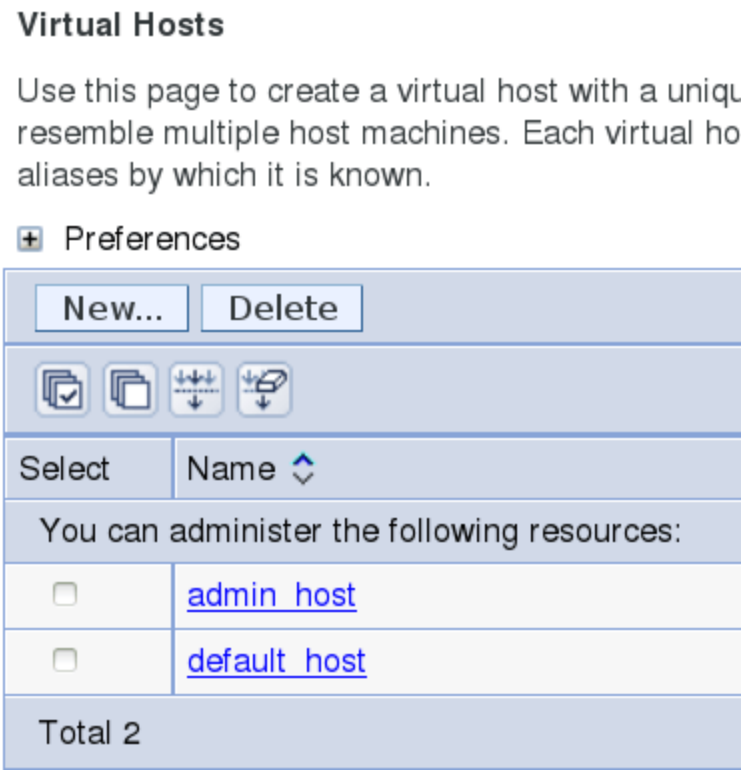


So now you know about the virtual host and next is to **create & configure it**.

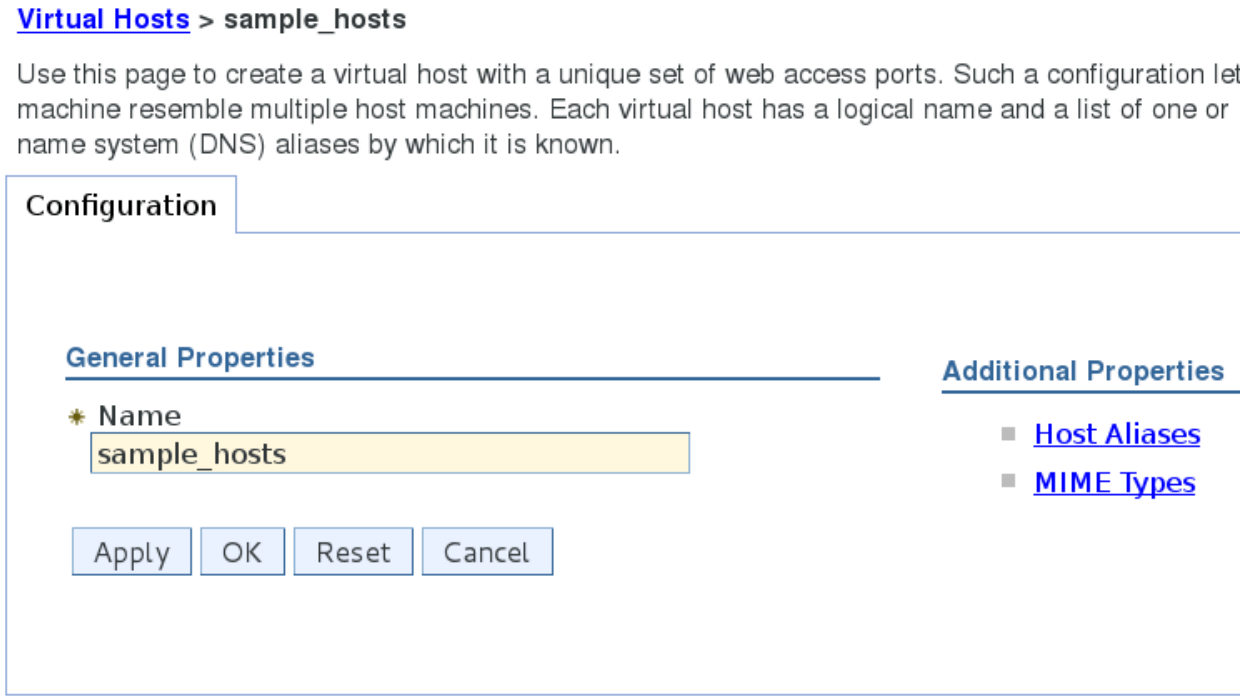
Virtual host creation is done through WAS administrative console, located under Environment >> Virtual hosts. Let's create one and name it – sample_hosts

- Login into WAS Administrative Console
- Go to Environment >> Virtual hosts
- Click on New

You can see two inbuilt virtual hosts in the above list which I mentioned above.



- Enter the name – sample_hosts



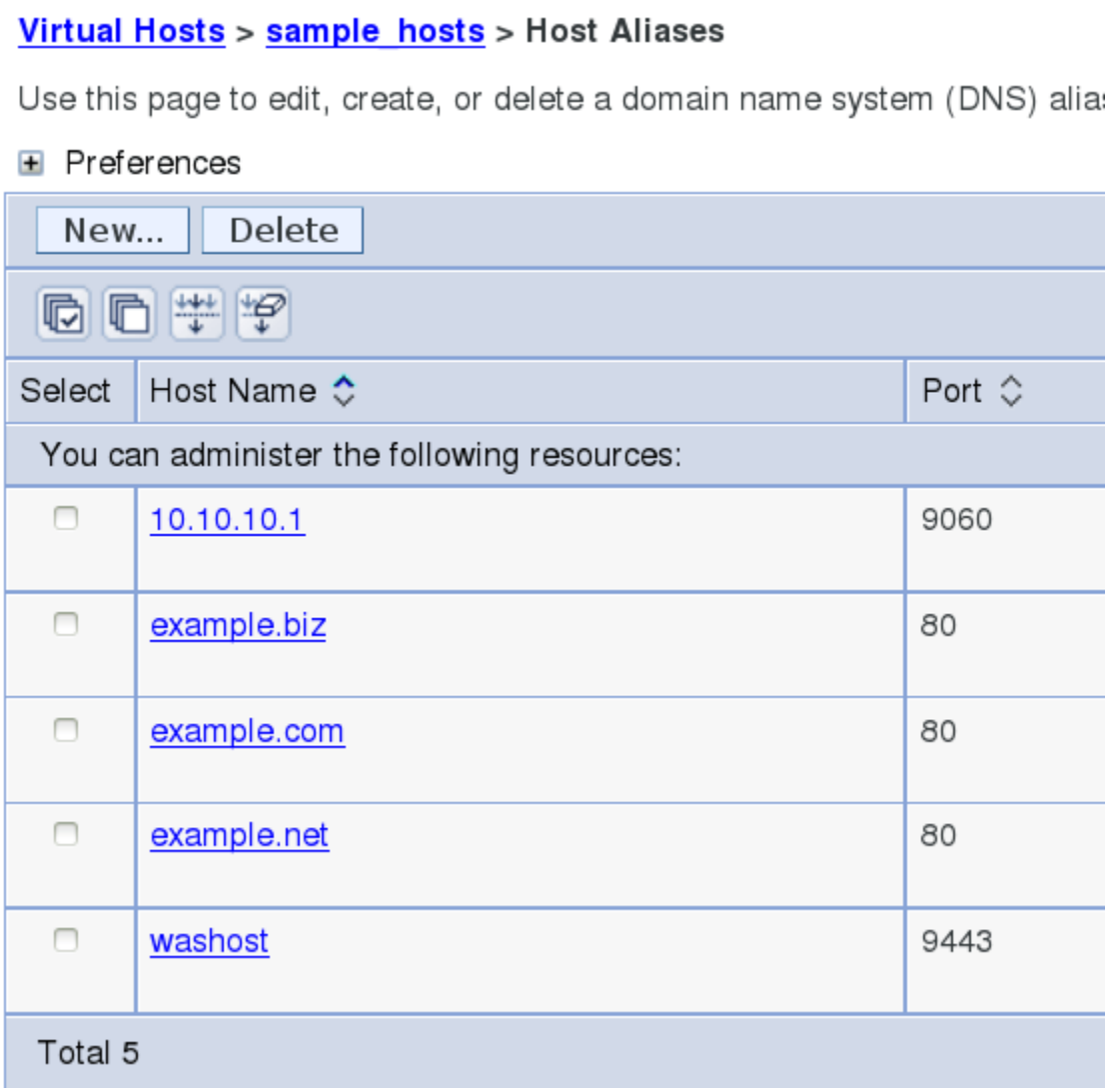
- Click on OK

So now you have a new virtual host and it's time to **configure** it.

There are two types of configuration you can do and both configurations are located inside virtual host under "Additional Properties".

Host Aliases: here you can enter DNS/Host/IP Name and Port, which will be allowed, to access the application using this virtual host.

- Go to Environment >> Virtual hosts
- Select the virtual host from the list
- Click on Host Aliases
- Click on New
- Enter Host Name & Port
- Click OK and save/review the configuration and here is what I created



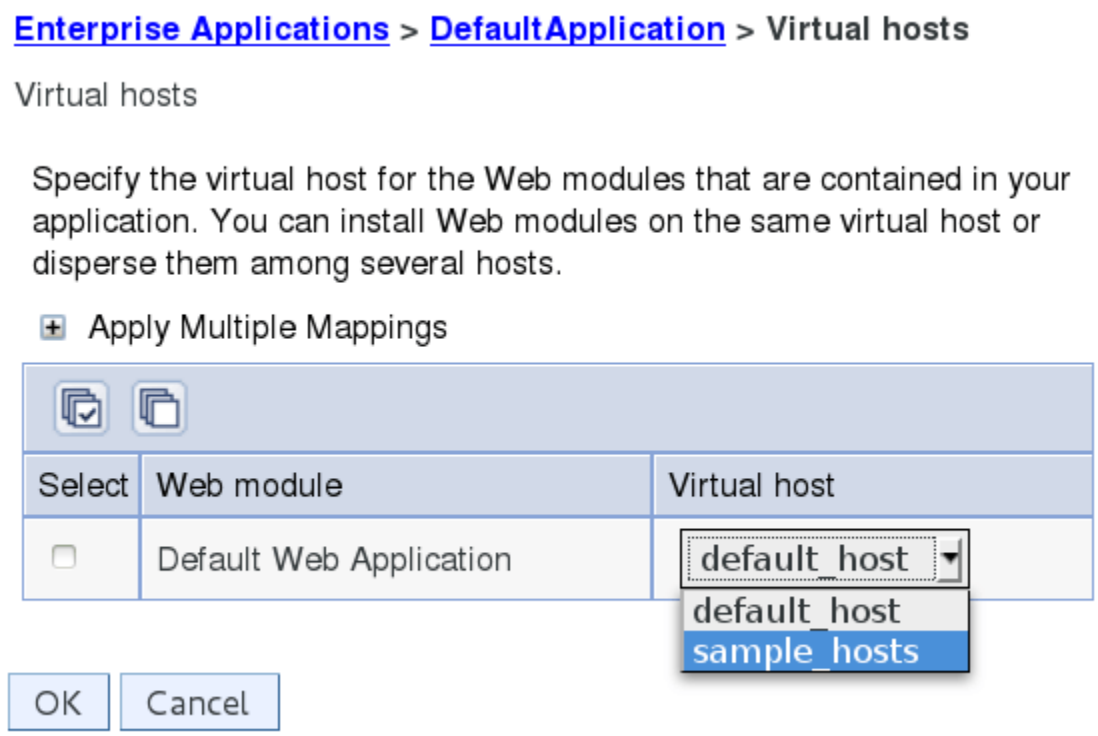
MIME Types: here you can add/create/modify MIME types for your application. However, by default WAS would have more than 640+ MIME types created which will cover almost any type of application. This default list is created when you create any virtual host.

- Go to Environment >> Virtual hosts
- Select the virtual host from the list
- Click on MIME Types
- Click on New to create one. However, as I mentioned above most likely you don't need to create, as almost all types of applications will be covered with default lists.

So next is to **change virtual host** in a deployed application

There might be a scenario where you got to segregate virtual hosts based on application. Here is how you can change the virtual host for any application.

- Login into WAS administrative console
- Go to applications>> Websphere enterprise applications
- Select your application and click on Virtual hosts located under Web Module Properties
- Select the desired virtual host and click on OK



- Click on Review/Save the configuration. You must restart the JVM in order to get the virtual host reflected.

You might be interested to know the virtual host configuration is stored in **virtualhosts.xml**. Location may differ from environment to environment so the best way to find the exact location of virtualhosts.xml is: –

On UNIX environment

```
find / -name virtualhosts.xml
```

Through WAS Console

When you make any changes in the virtual host it will ask you to review/save. Click on Review and it will show you the path of virtualhosts where it will save the configuration.

Now the time to go through world-famous **WebSphere error related to the virtual host**, which is most likely due to either not configured, or misconfiguration.

Error related to Virtual host

A WebGroup/Virtual Host to handle /test has not been defined SRVE0255E: A WebGroup/Virtual Host to handle **localhost**:**8080** has not been defined.

If you are getting the above error when you access the application then you will learn in just a few minutes how to fix it.

Let's take a look at the error carefully. There are two things you got to look which I highlighted in red color.

First – **localhost** has not been defined

The application is throwing an error because the localhost doesn't exist in Host Aliases of a respective virtual host. So quick fix is to add localhost in your virtual host.

This could be also when you have multiple virtual host and application is mapped to an incorrect virtual host. So you got to review the configuration and ensure the localhost exists in the requested virtual host.

Second – **8080** port number

Most of the time you will configure the virtual host with URL and default Http/https port, which is 80 or 443. However, if you want to access the application directly (bypassing web server) by using JVM and port you need to enter IP/Host and Port details in Host Aliases.

So to fix this – you got to ensure localhost:8080 is added to a virtual host, which is mapped to the respective applications.

Last section of this article – **Best practices of using Virtual Host** in WAS environment.

Every application is unique and one configuration may not work well in others. However, there are some rules you may follow which has helped me, and hopefully you too.

- If you are having multiple applications in a single cell, you may consider using a separate virtual host for each or at least for a group of applications.
- Take a backup of WebSphere configuration before you make any changes so if something goes wrong you may recover to the original state without damage.
- Restart mapped JVM after making changes to the virtual host so configuration is reflected.
- Propagate (if managed webserver) & Generate Web Server plugin and restart Web Server to get the configuration reflected.
- Don't put * in Host Aliases as this means any URL is allowed which may lead to a security loophole.

I hope you have a better understanding of Virtual Host in IBM WebSphere Application Server now.

Part 7

Stopping & Starting Deployment Manager, Node Agent & JVM with Scripts

stopManager.sh

You can stop the Deployment Manager using above command.

```
[root@localhost bin]# ./stopManager.sh
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/logs/dm
gr/stopServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server dmgr stop completed.
[root@localhost bin]#
```

Note: this has to be executed in DMGR profile path.

startManager.sh

You can start the Deployment Manager using above command.

```
[root@localhost bin]# ./startManager.sh
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/logs/dm
gr/startServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 9183
[root@localhost bin]#
```

Note: this has to be executed in DMGR profile path.

startServer.sh

To start the JVM, you can use startServer.sh with server name like below.

```
[root@localhost bin]# ./startServer.sh server1
ADMU0116I: Tool information is being logged in file
```

```
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
server1/startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is
10633
[root@localhost bin]#
```

Note: Node Agent must be started before starting JVM.

stopServer.sh

You can shutdown JVM by executing above command along with JVM name.

```
[root@localhost bin]# ./stopServer.sh server1
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
server1/stopServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.
[root@localhost bin]#
```

stopNode.sh

To stop the respective Node Agent, you have to go to that profile and execute stopNode.sh to stop the Node Agent.

```
[root@localhost bin]# ./stopNode.sh
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
nodeagent/stopServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server nodeagent stop completed.
[root@localhost bin]#
```

startNode.sh

Go to respective profile and execute startNode.sh to start the Node Agent.

```
[root@localhost bin]# ./startNode.sh
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
nodeagent/startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server nodeagent open for e-business; process id is
11363
[root@localhost bin]#
```

serverStatus.sh

To find out JVM's status, you can use this script with –all argument.

This must be executed in respective profile level. If you execute this in DMGR profile level, it will just show the status of DMGR.

```
[root@localhost bin]# ./serverStatus.sh -all
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
serverStatus.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU0503I: Retrieving server status for all servers
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: nodeagent
ADMU0506I: Server name: server1
ADMU0508I: The Node Agent "nodeagent" is STARTED
ADMU0508I: The Application Server "server1" is STARTED
[root@localhost bin]#
```

Backup & Restore

backupConfig.sh

One of the first things to learn while working in production support is to how to take a backup. When nothing works – backup helps.

You can use this script to take a backup of your WebSphere environment configuration. As a best practice, you can use "–nostop" argument, so it takes a backup without stopping Deployment Manager.

```
[root@localhost bin]# ./backupConfig.sh -nostop
```

```
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
backupConfig.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU5001I: Backing up config directory
          /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config
to file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin/W
ebSphereConfig_2015-04-12.zip
.....................................................................
.....................................................................
.....................................................................
.....................................................................
........
ADMU5002I: 933 files successfully backed up
[root@localhost bin]#
```

restoreConfig.sh

If you have changed configuration and things are not as expected and there is a time to restore your configuration. Well, you can use the backup file to restore the configuration.

```
[root@localhost bin]# ./restoreConfig.sh WebSphereConfig_2015-
04-12.zip -nostop
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
restoreConfig.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU5502I: The directory
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config
           already exists; renaming to
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/confi
g.old
ADMU5504I: Restore location successfully renamed
ADMU5505I: Restoring file WebSphereConfig_2015-04-12.zip to
location
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/confi
g
.....................................................................
.....................................................................
.....................................
ADMU5506I: 933 files successfully restored
```

```
ADMU6001I: Begin App Preparation -
ADMU6009I: Processing complete.
ADMU6002I: Begin Asset Preparation -
ADMU6009I: Processing complete.
[root@localhost bin]#
```

Do you like it so far? Scroll down for more fun!

Getting version & fix pack information

versionInfo.sh

To find out WAS version, build level, package, architecture & installed features installed on your server.

```
[root@localhost bin]# ./versionInfo.sh
WVER0010I: Copyright (c) IBM Corporation 2002, 2012; All rights
reserved.
WVER0012I: VersionInfo reporter version 1.15.1.48, dated
2/8/12
-------------------------------------------------------------------
-----------------
IBM WebSphere Product Installation Status Report
-------------------------------------------------------------------
-----------------
Report at date and time April 12, 2015 3:18:41 AM PDT
Installation
-------------------------------------------------------------------
-----------------
Product Directory       /opt/IBM/WebSphere/AppServer
Version
Directory       /opt/IBM/WebSphere/AppServer/properties/version
DTD
Directory           /opt/IBM/WebSphere/AppServer/properties/ver
sion/dtd
Log Directory           /var/ibm/InstallationManager/logs
Product List
-------------------------------------------------------------------
-----------------
NDTRIAL                 installed
Installed Product
```

```
------------------------------------------------------------------
-----------------
Name                  IBM WebSphere Application Server Network
Deployment
Version                8.5.5.0
ID                    NDTRIAL
Build Level            gm1319.01
Build Date            5/14/13
Package                com.ibm.websphere.NDTRIAL.v85_8.5.5000.20
130514_1044
Architecture          x86-64 (64 bit)
Installed Features    IBM 64-bit WebSphere SDK for Java
                      WebSphere Application Server Full Profile
                      EJBDeploy tool for pre-EJB 3.0 modules
                      Embeddable EJB container
                      Stand-alone thin clients and resource
adapters
------------------------------------------------------------------
-----------------
End Installation Status Report
------------------------------------------------------------------
-----------------
[root@localhost bin]#
```

Note: you may be interested in following supported arguments.

- -fixpacks: To display fix packs information
- -long: To display all fix packs and ifixes
- -ifixes: To display ifixes information

getHistoryReport.sh

If you are performing auditing or just want to list out components, fixes, refresh pack with dates, you can run this command, which will generate historyReport.html in current working directory, which is usually bin folder.

getVersionReport.sh

To display build version and build date of WebSphere installation. Often asked by IBM support guys to investigate if any suspected issue with a particular version.

Clearing the Cache

There might be various reasons to clear the cache the most obvious one would be after an upgrade. There are two caches that you should consider clearing 1) JVM 2) OSGi.

clearClassCache.sh

To clear JVM's class cache, you can execute above script.

Note: JVM's must be stopped before clearing class caches.

osgiCfgInit.sh

Execute above command to clear OSGi profile & server cache.

```
[root@localhost bin]# ./osgiCfgInit.sh
OSGi profile cache successfully cleaned for
/opt/IBM/WebSphere/AppServer/profiles/Dmgr01.
OSGi server cache successfully cleaned for
/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/servers/dmgr.
[root@localhost bin]#
```

Note: don't forget to stop the running processes before clearing caches.

Managing Profiles

managesdk.sh

You can toggle the version if you have installed multiple SDK. You can also use this script to find out available associated SDK to your profile. Below example shows a list of available SDK.

```
[root@localhost bin]# ./managesdk.sh -listAvailable
CWSDK1003I: Available SDKs :
CWSDK1005I: SDK name: 1.6_64
CWSDK1001I: Successfully performed the requested managesdk
task.
[root@localhost bin]#
```

pmt.sh

PMT (Profile Management Tool) can be used to create WebSphere profiles in GUI mode. Creating profiles using PMT is very easy – all you got to do is create the desired level of profile and follow the wizard. You got to try it!

syncNode.sh

For some reason, if you can't perform Node Sync through Administration Console, you can use syncNode.sh from profile level. Node Agent must be stopped before use this script.

You have to pass an argument for DMGR host & SOAP port number.

```
[root@localhost bin]# ./syncNode.sh localhost 8879
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/
syncNode.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU0401I: Begin syncNode operation for node localhostNode01
with Deployment
           Manager localhost: 8879
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0402I: The configuration for node localhostNode01 has been
synchronized
           with Deployment Manager localhost: 8879
[root@localhost bin]#
```

- localhost = Deployment Manager hostname
- 8879 = DMGR SOAP Port number

I hope above scripts are useful to your daily work.