**WebSphere Application Server**

**Introduction:**

* WebSphere application server is a software framework which hosts the java based web applications.
* It is a run time environment for j2ee applications.
* Various platforms are supported by was such as linux, windows, Aix,ios,IBM etc.
* It works with various web services like apache http server, Microsoft Internet Information Services.

**Various WAS packages:**

* WAS network deployment
* WAS Express
* WAS Base

**Installation:**

WAS can be installed only through IBM installation manager

For installation we need to the follow the below steps:

* Install IBM installation manager
* Install WAS ND through IBM IM

Based on the credentials we need to install the IBM IM through GUI mode

* if the user is admin then we need to run the install utility
* if the user is non admin the we need to run the userinst utility
* if it is a group user then we need to run groupinst

There are various websphere terminologies:

1. Application server
2. Node
3. Cell

**Application server**

* The application server is the platform on which java based applications run
* It is an environment where you host banking applications

# Node

* A node is a logical group of one or more application servers on a physical computer
* It is an administrative grouping of application servers for configuration and operational management within one operating system instance
* Application servers that run on a specific host is grouped as a logical unit for ease of administration and enabling other WAS features.

**Cell:**

* A cell is a logical group of nodes that enables common administrative activities in a WAS distributed environment
* The below illustrates the components of the application server environment. The number of nodes and servers present in application server environment varies based on the topology.

**Deployment manager**

* Application servers can be managed centrally through deployment manager
* It is a central administration point of a cell that consists of two or more nodes (stand-alone application server) and node groups in a distributed server configuration

**Node agent**

* While adding a stand-alone application server to deployment manager cell, an operating system process will be created in the machine wherein the stand-alone application server resides. This is called node agent

There are various WAS topologies:

* Standalone Environment
  + It is used for testing the applications where clustered environment is not supported.
* Distributed server environment
  + Multiple servers are managed from a single deployment manager cell.
* Administrative agent environment
  + An administrative agent can **manage single server remotely** without any additional connection to the physical machine
* Job manager environment
  + Job manager **manages multiples topologies** through a single entry point

**Usage of WebSphere tools**:

**Profile management tool(PMT) :**

* PMT is used to create application servers on which you host banking application
* Application server's run time environment is called as profile
* Below are the profile types in which various profiles are created :
* **Cell** (Deployment manager and a federated application server)
* **Application server**(Stand-alone application server profile)
* Management (**Deployment manager**, job manager and administrative agent profile)
* Managed (Federated node to DMGR without server)
* Secure Proxy (Secures the WAS topology).

**Creation of an application server in linux environment:**

* It can be created through manageprofiles.sh or through gui mode by selecting the typical profile creation by running the profile management tool(pmt.sh)
* Select the typical profile creation option
* Provide the WAS credentials

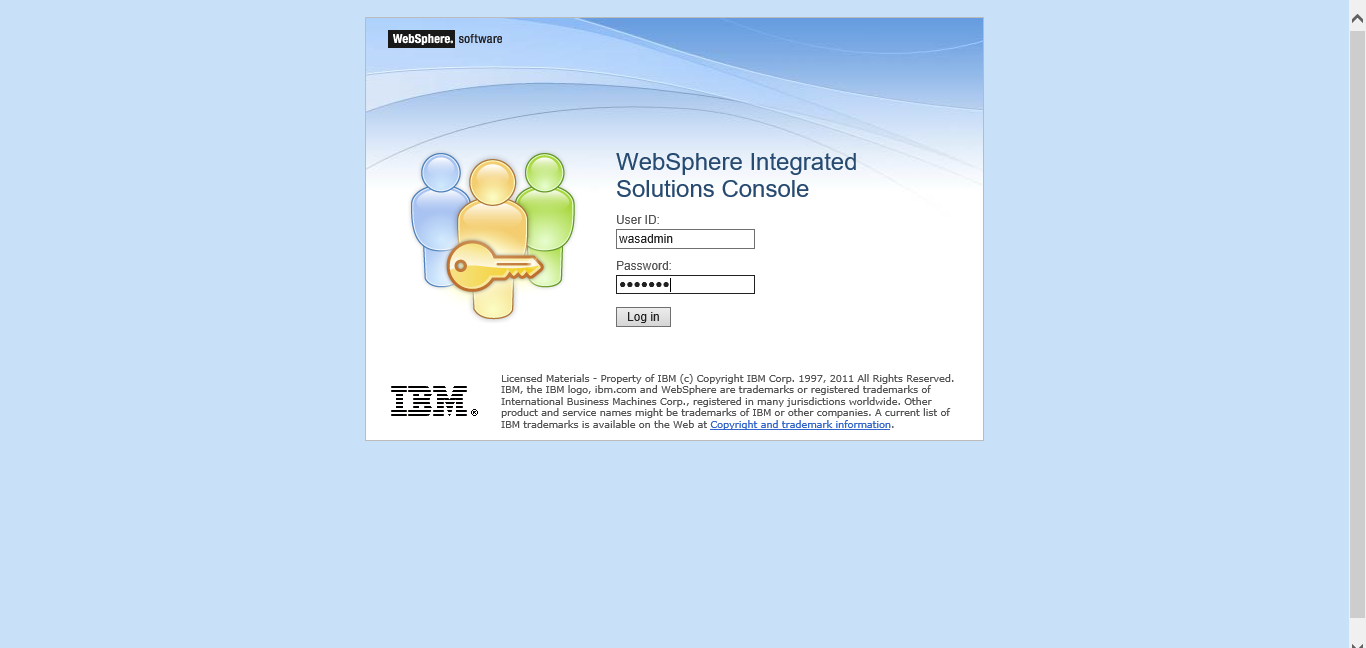
By then it can be created with the values as below:

* Name of the server: server1 (auto-generated name)
* Profile Name: AppSrv01
* Profile Template: Application server
* Host: Local host (mysimstlin-0174)
* Cell Name: mysimstlin-0174Node01Cell (auto-generated name)
* Node name: mysimstlin-0174Node01 (auto-generated name)
* Port number: 9043 (auto-generated value)

The above process explains the server creation through GUI mode, there is a second method which we can create profiles, i.e. By running the command manageProfiles.sh, through this we can create the application server and once the server is created to check the server’s status below is the command:

./startServer.sh

Once the server has started the administrative console would look like the below:



**Central Server management**:

The application servers which were created through **PMT/manageProfiles.sh** methods need to be managed by adding these servers to deployment manager’s cell through a node which is known as **federation**.

Deployment Manager:

* Each node contains an node agent through which the all the servers and services are configured, updated and monitored.
* The node agent updates all the changes made in node to deployment manager
* The deployment manager contains the configuration of each server which is federated to the cell
* The process of updating the server’s configuration to match the configuration of deployment manager is known as **synchronization**.
* After federating an application server to deployment manager the application server’s console will be no longer exists.

Distributed server environment setup can be done using three different methods based on the the phase where it will be used such as development/testing/production phase.

**Development/Test phase process**:

* Create a cell profile
* It creates both a deployment manager and an application server profile. The application server node is federated to the cell
* Both are on the same system
* This method can be used in a **development/test environment**
* It is a simple distributed system on a single server
* Banking applications can be deployed on this environment during its development and testing phase

Below is the command for federation:

create

profileName=AppSrv01  
profilePath=/usr/IBM/WebSphere/AppServer/profiles/AppSrv01  
templatePath=/usr/IBM/WebSphere/AppServer/profileTemplates/cell/default  
nodeName=<DmgrNodeName>  
cellName=<CellName>  
hostName=<HostName>  
appServerNodeName=<AppServerNodeName>  
dmgrProfilePath=/usr/IBM/WebSphere/AppServer/profiles/Dmgr01  
nodePortsFile=/usr/IBM/WebSphere/AppServer/profiles/Dmgr01/properties/nodeportdef.props

portsFile=/usr/IBM/WebSphere/AppServer/profiles/Dmgr01/properties/portdef.props

In the profile management tool, we have to select the cell (deployment manager and federated application server) to be created.

**Production Environment:**

* Firstly, we need to create a deployment manager profile
* We need to create one or more custom node profiles
* These nodes can be federated during profile creation, or manually later
* These nodes may exist in the deployment manager system, or in another OS instance
* Application servers can be created by using wsadmin scripts or administrative console based on the requirement
* This method can be used when multiple nodes/application servers on a node, or clusters creation in **production environment**
* This is the best suited production environment to host banking applications.

**Development Environment**:

* Deployment manager profile to be created
* one or more application server profiles to be created
* Federate these profiles into deployment manager
* It adds nodes and application servers into the deployment manager cell
* These nodes may exist in the deployment manager system, or in another OS instance
* This method can be used in **development/small configurations**.

By the above said phases we can create a distributed environment.

Federation of an application server to DMGR can be done in two ways:

* Through addNode command
* Through administrative console page of deployment manager

**Process of federation through addNode command**.

We have to start the application server and the deployment manager profiles by using serverStatus.sh command in order to federate the application server to DMGR (deployment manager’ cell)

Then we need to running the addNode.sh command with soap default port 7014, by this the server is federated to dmgr’s cell node demo node with the below command:

addNode.sh localhost 7014 -includeapps

**Federation of an application server node to deployment manager cell through the administrative console**:

In this we can directly federate the node through deployment manager console to add the node below is the process:

**System Administration->Nodes->add node->managed node**

below are the properties in adding a node to deployment manager

* Name of the application server profile and server: AppSrv01 and server1
* Name of the AppSrv01 node: mysimstlin-0174Node02
* Name of the DMgr cell: mysimstlin-0174Cell01
* Name of the deployment manager profile: Dmgr01
* SOAP connector port number of Dmgr01: 8883

we can view the nodes added by clicking on the **available nodes** once the node is created in the console. And we can control the server from deployment manager console itself.

**Synchronization**:

In this feature for suppose if we have done any changes to the node federated to deployment manager the changes need to updated to deployment manager which can be done through synchronization.

During a synchronization operation, a node agent checks with the deployment manager to see if any configuration changes that apply to the node were updated.

2. New or updated changes are copiedto the node repository, and deleted documents are removed from the node repository.

3. This can be done automatic or manual settings

Automatic - the default file synchronization interval is 60 sec, and starts when the application server starts.

Manual - it can be done by administrative console page/wsadmin tool/sync Node command

Below are the commands used for synchronization:

* startNode - to start the node agent process of application server
* stopNode - to stop the node agent process of of application server
* syncNode - to synchronize the node with deployment manager cell.

Below is the process of synchronization to update the changes of the node to the deployment manager cell:

* Stop the application server node agent

./stopeNode.sh

* Synchronize the application server to dmgr’s cell

./syncNode.sh localhost 7014

* Start the application server node agent process

./startNode.sh

**Application deployment**:

The application deployment can be done in two ways:

**Administrative console wizard**:

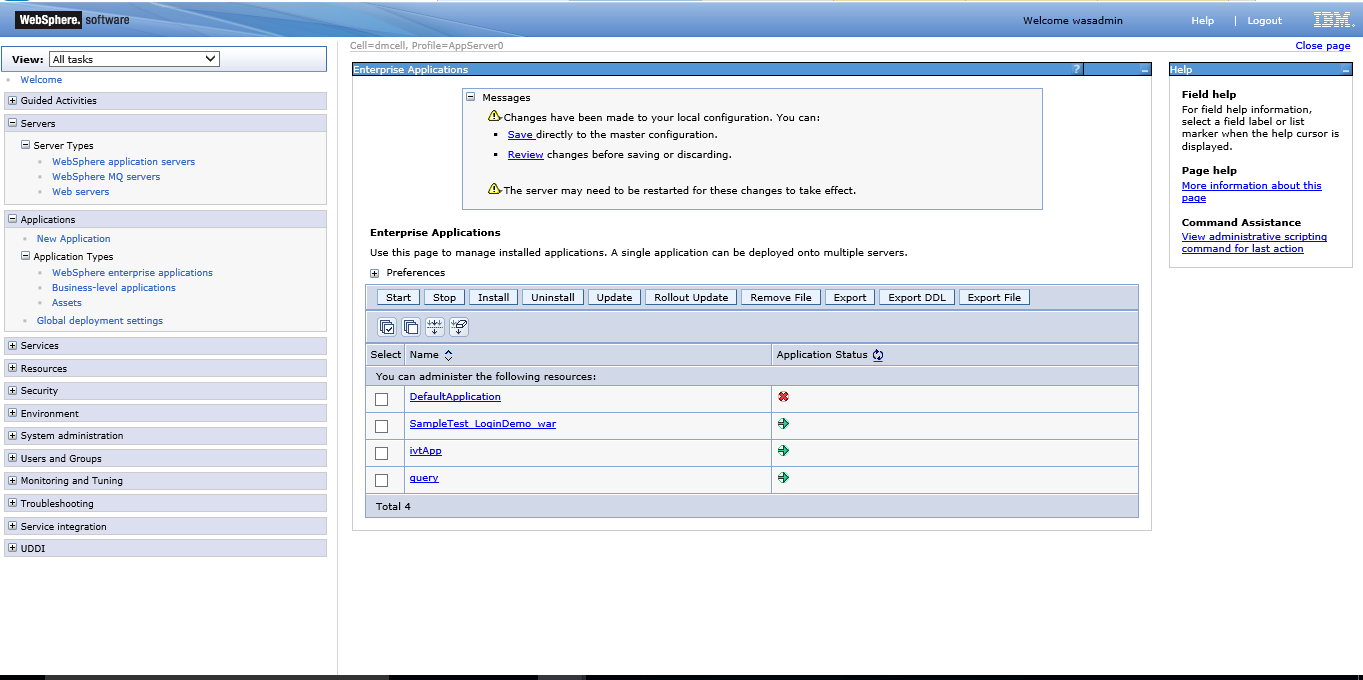
This is a GUI mode in which we can be deploy the banking applications on application server which supports WAR, EAR, JAR and SAR modules. Which is suitable for production environment

**Wsadmin scripts**:

This can be done using AdminApp object in command mode, we can automate the deployment activity through the script file.

Below is the process to deploy the application on application server:

* In administrative console page navigate to applications->WebSphere enterprise applications ->install
* Choose the file to be deployed ->. ear file
* Select the type of deployment based on the requirement.
* We need to the select the servers which we want to deploy the application
* Once installed successfully we can start the application
* We can get the port of the server by navigating to application servers->demoserver->ports
* Now we would be able to access the application through the browser as <https://localhost:portnumber/appname>



**User Management**:

As the no of application of the bank would be huge,the middle ware administrator will decide and authorize, authenticate the administrators to secure the application with the responsibilities provided. Below are the users to be created and appropriate permissions to be granted

bankingapp\_Guest1 - Secondary administrator  
bankingapp \_Guest2 - To control the services (Start/Stop)  
bankingapp \_Guest3 - To view the configuration and the current state of the banking applications  
bankingapp \_Guest4 - To change the WAS configuration  
bankingapp \_Guest5 - To perform run time operations on banking applications

**User Registries:**

* WAS provides security infrastructure and mechanisms to protect sensitive resources
* Authentication is the process of confirming a user or WAS system identity
* Authorization is the process of verifying that the user has access to WAS resources or not.

User registries store user id and user related information which can be used at the time of authentication, also it store the information to do security related administrative functions.

Below are the type of registries:

**Federated Repository**:

* If we enable security when you create a profile for distributed systems, a file-based registry is automatically created and populated with one administrative user ID
* IBM WebSphere uses federated repository by default, though it is not recommended to be used in production environment.

**Local Operating System:**

* WAS authentication mechanism uses the user accounts database of the local operating system, it is used in stand-alone server environment.

**Standalone Light directory Access protocol(LDAP)**:

* WAS security provides and supports the implementation of LDAP directory servers, which is used as a repository for user and group information
* IBM Tivoli directory server is an example for LDAP server, we can configure LDAP through IBM Tivoli directory server
* This can be used, if you have huge number of users accounts who perform various activities of WAS environment. Banking applications user accounts can be stored in LDAP server as it is an enterprise application.

**Custom Registry**:

This is used when the user information is stored in different formats for example is a database.

**Note**: Only single registry can be configured in a certain scope.

**Administrative Security Roles**:

Below are the users:

**Users**:

bankingapp\_Guest1 - Secondary administrator  
bankingapp \_Guest2 - To control the services (Start/Stop)  
bankingapp \_Guest3 - To view the configuration and the current state of the banking applications  
bankingapp \_Guest4 - To change the WAS configuration  
bankingapp \_Guest5 - To perform run time operations on banking applications

Below are the roles for the above said users which can be configured based on their responsibilities:

**Roles**:

* Admin Security Manager
* Administrator
* Auditor
* Configurator
* Deployer
* ISC admins
* Monitor
* Operator

**Configurator:**

* It has monitor permissions and can changethe WebSphere Application Server configuration.

Eg: bankingapp\_Guest4

**Operator:**

* It has monitor permissions and can changethe run time state (start/stop services).

Eg: bankingapp\_Guest2

**Monitor:**

* It has the least permissions. This role primarily confines the user to viewingthe WebSphere Application Server configuration and current state.

Eg: bankingapp\_Guest3

**Deployer:**

* It has permission to perform both configurationactions and run time operationson applications.

Eg: bankingapp\_Guest5

**Administrator:**

* **The administrator has the permissions for operator, configurator, accessing sensitive data like password etc.**

**Eg: bankingapp\_Guest1**

**Admin Security Manager:**

* **The manager maps other users to security roles.**

**ISC Admin:**

* The ISC admin has administrator privileges for managing users and groups from within the administrative console only.

**Auditor**

* The auditor has permission to view and changethe configuration settings for the security auditing subsystem.

The above said are the various security roles in WAS.

Process to enable and disable the security in WAS:

The security can be enabled in two methods:

* One before the profile is created
* In the Console page post profile creation

If the security is not enabled, if the administrator user is not enabled any one can access the url in DMGR.also users cannot be added under manage users console in WAS if the security feature is not enabled

To enable security through console page:

* Administrative console->Security->global security->security configuration wizard
* We need to choose federated repository ->Enter the credentials ->save it

Username as ‘**sampleuser**’

* Click on finish->restart the deployment manager

The administrative user **sampleuser** has been enabled

For example, if the admin user password is lost and the admin user can be disabled and can be reset by the following methods:

* Using wsadmin.sh file
* By modifying security.xml file

Using wsadmin:

* Run wsadmin.sh file from WAS installation directory bin

Command: ./wsadmin.sh –conntype NONE

Run the command as below:

* Wsadmin>security off
* Restart the application server

By using the command ./../profiles/demoserver/bin/stopServer.sh to first stop the server

Start the server using the command ./../profiles/demoserver/bin/startServer.sh

Once the security feature is disabled the admin user can login without a password and reset the password as said above in enabling the security process.

By modifying security.xml file:

By making the property enabled as false we can disable the feature of security through settings.xml

Steps to disable security through settings.xml:

* We need to take a backup of the settings.xml file which is present under the location of profile/profile name/config/cells/CellName
* Change the property of enabled as false
* Restart the application server (demo server)

Now the admin user can enter into the console of application server profile without a password and follow the above said process for enabling security.

**Creation of additional Users**:

Deployment manager console page->users and groups->Manage users->create

Create a user with the name ‘sampleuser’,enter the username

Navigate to->administrative user roles->add

Login with user named sample user whose role is assigned to operator so that the user can start/stop the servers and applications.

**Database configuration**:

**JDBC Resources**:

# Global J2C (Java 2 Connector) authentication alias security domains

* + These data entries are used by JDBC data sources and resource adapters
  + It is the **alias**for the **resources** which you configure
  + For eg we can create an alias name for Oracle credential. We can provide the alias name while testing the connectivity of database instead of actual username to ensure the security.

# JDBC

* + WAS used the driver that are encapsulated with JDBC to connect to various databases such as Oracle, Apache Derby, DB2, Microsoft SQL Server, Sybase and Informix
  + Database specific driver file is required to configure the database with WAS.

# Data source

* Application will connect to database through this resource.

**Configuring jdbc resources**:

Configuring J2c authentication for oracle database credentials

* Create an alias name for oracle credential as username as guest and password as guest1
* Place the jar in opt/IBM folder ojdbc6.jar file
* Define a datasource for oracle database.

**Configuring an oracle database**:

Configuring J2C authentication data:

We need to create a J2C alias name for oracle database with the username as guest and the password as guest1. we can connect to the oracle database with the password as guest1.

Login into administrative console ->security->global security->JAAS ->J2C authentication data->new

enter the alias name for oracle credentials under the properties fields.

**Enabling single sign-on In WAS:**

The single sign on is enabled through LTPA keys(Lightweight third party Authentication) keys.We export the LTPA keys into one instance of websphere application server and then import the same key into another instance of websphere application server

To enable the single sign on feature in WAS please follow the below process:

Log in to webpshere application server admisnitration console

Navigate to security->global security

In Authentication cache settings ->under web and SIP security->Single sign on(SSO)

In general properties tab->we need to specify the configuration values of SSO as below:

**Properties**:

* **Enabled** property by default it is selected
* **Requires SSL property**-here we need to specify the domain name which is being used for the servers
* **Interoperability Mode:** needs to be selected if not selected by default
* **Web inbound security attribute propagation:** it is enables by default

Once all the properties are entered click on OK to save the configuration to master configuration

## **Exporting the LTPA key:**

The LTPA key can be imported into on server instance. Below is the process:

Log in into administrative server console page

Navigate **to security->Global security->authentication->LTPA**

In the **cross cell single sign on** section we need to specify the password for LTPA key.

Now enter the LTPA key name and the directory to which the key should be exported the fully qualified key name as in windows **C:\key\_name** and in Linux environment it is **/opt/key\_name**

Click on **export keys**

Click on **ok** to save it to master configuration

Importing the LTPA key into the another server instance where we host the application

**Process for importing LTPA key**:

Navigate to the directory where the LTPA key needs to be imported specify the filesystem directory location

* Log in to the WebSphere Application Server administration console.
* Navigate to **Security** > **Global security** > **Authentication** > **LTPA**.
* In the **Cross-cell single sign-on** section, specify the password for the LTPA key.
* Enter the directory on your file system where you copied the LTPA key in the **Fully qualified key name** field. Click **Import keys**.
* Click **OK** and save to the master configuration.
* Restart both the server you exported the LTPA key from and the server into which you imported the LTPA key. Restart the servers only after you have imported the LTPA key into all the servers for which you plan to establish SSO.

**Note**: restart the server in which the key is imported

**Verifying single sign-on:**

Log in to the WebSphere application server where the LTPA key is exported

In the browser’s address bar enter the url for the WebSphere application server administration console where the key is imported.

If the WebSphere application server administration console opens without requiring a log in, then the SSO is successfully set up for the application

**Adding certificates in WebSphere application server:**

The certificate from HTTP server can be added to WebSphere trust store as in the below process:

* Log into the IBM WebSphere Application Server Integrated Solutions Console and select Security > SSL Certificate and key management > Key stores and certificates.
* Click CellDefaultTrustStore.
* Click Signer Certificates.
* Click Retrieve from port.
* Enter the Host name, SSL Port, and Alias of the web server. The Alias is typically an arbitrary string that will become the name of the credentials.
* Click Retrieve Signer Information and then click OK. The root certificate is added to the list of signer certificates.

**Exporting and importing a configuration archive:**

Wsadmin is the tool which is used for jython scripting**:**

the wsadmin tools is divided into 4 groups as AdminApp,AdminConfig,AdminTask,AdminControl

AdminApp:It is used to install or remove the applications in a an application server

,AdminConfig:to create or modify the object attributes of configuration

AdminTask:it is used for export and import the profiles

AdminControl:To start or stop the servers and applications installed

We can use the ConfigArchiveOperations command group to export and import server

configurations or entire cell configurations as a compressed archive (CAR) file. You can use

this capability to replicate server or profile configuration.

The general procedure to use an archived file is:

* Export a WebSphere Application Server configuration into a compressed archived file

containing the server configuration.

* Optionally, extract the files for browsing or updating for use on other systems, for example,

For updating resource references.

* Upload the archive file to the target system.
* Import the archive file. The import process requires that you identify the object in the

configuration you want to import and the target object in the existing configuration. The

target can be the same object type as the archive or its parent. Consider the following information as below:

* If we want to import the application server configuration to another application server then the configurations would be merged.
* If we want to import a server archive to a node, then the server is added to the node.

We can use this functionality between two application server profiles under the same or different product installations on the same or different host environments or to replicate the profile configuration across different platforms.

**Profile Archive**:

**exportWasprofile** command to export the entire cell configuration to a configuration archive. This archive file can be used to restore the configuration or clone the original profile on another machine or system.

**Note**: Only a base server configuration with a single node is supported by the

**exportWasprofile** command.

* **exportWasprofile** command in the AdminTask object can be used to export profile

configuration. The command needs to be executed through **wsadmin** from the *profile\_root*/bin directory

creates an archive of a profile.

Below is the command which create an archive for the profile created.

Need to navigate to application server profile directory and then run the command ./wsadmin.sh

wsadmin>$AdminTask exportWasprofile {-archive /tmp/was85\_archive/Node01\_archive\_0622.car}

here the node01 is the profile being archived. The above command wsadmin is used by jacl scripting

**Import Was profile**:

The **importWasprofile** command in the AdminTask object imports and overwrites the profile

with the archive file configuration is below:

wsadmin>$AdminTask importWasprofile {-archive

/tmp/was85\_archive/Node01\_archive\_0622.car -deleteExistingServers true}

wsadmin>$AdminConfig save

The **-deleteExistingServers** parameter is optional. It deletes existing servers from the target

profile. Consider that when importing a profile with multiple servers, the node has to contain

exactly the same number of servers. If the number of servers is not the same, the following

error message occurs:

ADMB0016E: The number of servers in the configuration archive does not match the

number of servers in the system configuration. The product only supports

importWasprofile for profiles with the same number of servers.

**Importing SSL certificates into WAS**:

Keytool is the utility to be run in order to import the certificates into WAS

We need to export the certificate to the file path where the application server resides

Below is the path:

* path\_to\_WebSphere/WebSphere/AppServer/java/jre/bin directory

import the certificate into the keystore as below:

* keytool -import -alias KeyAlias -file path\_to\_certificate\_file -keystore cacerts -storepass changeit

for windows environment below is the command:

* Windows: path\_of\_keytool -import -alias ExchangeCert -file path\_of\_cert -keystore cacerts -storepass changeit

For Linux environment:

* Linux: path\_of\_keytool -import -alias ExchangeCert -file path\_of\_cert -keystore cacerts -storepass changeit

The alias can be of any string which point to the configuration

Keytool path is the path where the appserver is resided in which folder

Changeit is the default password of cacertificates

**Logging in websphere**:

Websphere application server provides many logging options and the significant incidents are recorded in log files.

Below are the levels of logging:

**Basic Logging**:

The basic logging deals with

* java virtual machine logs
* Diagnostic trace
* Service log files

The commonly named log files are SystemOut.log, SystemErr.log, trace.log, and activity.log.

**Advanced logging:**

This deals with high performance extensible logging(HPEL) which is the feature introduced in WebSphere application server 8.5

This provides three repositories:

**Log Data repository**:

* This repository specifies the data related to the log files as SystemOut.log, SystemErr.log, or java. util. logging at higher level.

**Trace Data Repository**:

* This repository specifies the data related to the log files as java. util. logging at low level.

**Text log:**

This is the plain text with logs and traces provided

* The HPEL formats the text data into binary format
* No need to clear the logs each time when the server is restarted

To enable HPEL below is the process:

* Navigate to troubleshooting->logs and traces->servername->switch to hpel mode

To read log and trace data by hpel there is a tool provided which is known as log viewer tool. This tolls can be used to view the

Log data with the parameters considered as level, logger name, or date and time.

**For windows environment**: logViewer.bat

**For linux environment**: logViewer.sh

For suppose if we want to write the data from given start date to stop date below is the command:

* logViewer.sh -outLog /tmp/promo.logs -startDate dd/mm/yyyy -stopDate dd/mm/yyyy

**configuring the HPEL logs and traces**:

* troubleshooting->logs and trace->servername->configure hpel logging

specify the directory path where the log files needs to be written and the logdata subdirectory is created in this directory path which we mention which the log files are written to the logdata subdirectory.