*XML Configuration Files*  
*Property files*  
*Log Files*

WebSphere stores its configuration to set of XML files. When we use the Admin console to configure WebSphere, certain XML files are updated internally.

**CELL-scope**

• admin-authz.xml  
Contains the roles set for administration of the Admin console.  
<profile\_root>/appsrv01/config/cells/<cell\_name>/

• profileRegistry.xml  
Contains a list of profiles and profile configuration data

• resources.xml  
Defines operating cell scope environmental resources, including JDBC, JMS, JavaMail, URL end point configuration, and so on.

• security.xml  
Contains security data , including all user ID and password information.

• virtualhosts.xml  
Contains virtual host and Multipurpose Internet Mail Extensions (MIME)-type configurations.

• variables.xml  
Contains cell level WebSphere variables

• wimconfig.xml  
Contains the federated repository configurations for global security  
<profile\_root>/config/cells/<cell\_name>/wim/config/

**NODE-scope**

• namestore.xml  
Provides persistent JNDI namespace binding data

• resources.xml  
Defines node scope environmental resources, including JDBC, JMS, JavaMail, URL end point configuration, and so on

• serverindex.xml  
Specifies all the ports used by servers on this node

• variables.xml  
Contains node level WebSphere variables

**SERVER-scope**

**${PROFILE\_HOME}/config/cells/${CELL}/nodes/${NODE}/servers/${SERVER}/server.xml**

• resources.xml  
Contains the configuration of resources, such as, JDBC, JMS, JavaMail, and URL end points at server scope

• server.xml  
Contains application server configuration data

• variables.xml  
Contains server level variables

**properties files**

If the global security is enabled WebSphere Application Server cell, you need to manually enter the username and password every time you run the wsadmin tool. By editing the sas.client.props and the soap.client.props files, you can specify the username and password you have configured for global security so you are not prompted to enter the username and password every time you run administrative scripts.

***soap.client.props file***  
com.ibm.SOAP.securityEnabled=<true  
com.ibm.SOAP.loginUserid=<username  
com.ibm.SOAP.loginPassword=<password  
Optionally, set the following property:  
com.ibm.SOAP.loginSource=none

***sas.client.props file***  
com.ibm.CORBA.loginUserid=  
com.ibm.CORBA.loginPassword=  
Also, set the following property:  
com.ibm.CORBA.loginSource=properties

**Log Files**

**JVM logs**  
***native\_stderr.log,native\_stdout.log***  
stdout and stderr streams are redirected to log files at application server startup, which contain text written to the stdout and stderr streams by native modules, that is, Linux Modules, and so on. In normal error-free operations, these logs files are typically empty.

***startServer.log***   
It is created in your logs directory when the server starts up. This log is very useful to determine JVM parameters used in the start-up process, the server’s process id, and also the date and time in which the server was started. If there are errors experienced during the start-up (for example, security configuration errors where the application server cannot start), then log information will exist for problem determination.

***stopServer.log***   
when server was stopped via a command line, the log will be written to this. If the server has trouble stopping, then Java stack traces will be written to the log which can be used in determining why a given application server failed to stop.

***SystemErr.log***   
contains Java exceptions and stack traces. An empty SystemErr.log file does not necessarily indicate a successfully running application server JVM. You may need to consult the other logs in this directory.

***SystemOut.log***   
This log file contains messages as generated by the JVM during runtime. Some messages are informational, some are warnings or status updates. Applications can be configured to write to the log and so it is very common for the SystemOut.log to be your first port of call in application debugging.

***<server\_name>.pid***   
contains the process id of the server. In Linux, this is the actual process id assigned to the JVM process.

**FFDC logs**  
FFDC directory contains detailed logs of exceptions found during the runtime of the WebSphere Application Server. Can be found at WAS\_ROOT/profiles/logs/ffdc