## 停用/启用 websphere

|  |  |
| --- | --- |
| 切换到websphere目录 | cd /opt/IBM/WebSphere/AppServer/bin |
| 启用 | ./startServer.sh server1 |
| 停用 | ./stopServer.sh server1 |

## JSP缓存问题

|  |
| --- |
| 把这个jsp从项目中删除或重命名，再到浏览器里面查看那个页面，这时可能会报错，之后，再把对应的jsp添加上或名字改过来，再次到浏览器里面查看应用的时候，就发现这jsp的更新效果出来了 |

### startServer.sh

|  |
| --- |
| #!/bin/sh  # All Rights Reserved \* Licensed Materials - Property of IBM  # 5724-I63, 5724-H88, 5655-N01, 5733-W60 (C) COPYRIGHT International Business Machines Corp., 1997,2005  # US Government Users Restricted Rights - Use, duplication or disclosure  # restricted by GSA ADP Schedule Contract with IBM Corp.  # Configuration Based Server Launcher  # Launch Arguments:  #  # serverName - the name of the server process to be launched.  # -script [script\_file\_name]  # -nowait  # -quiet  # -trace  # -timeout <time>  # -statusport <port>  #  set\_script\_executable()  {  scriptfile=start\_$1.sh  while [ "$#" -gt "0" ]  do  # check for -script option  if [ "$1" = "-script" ]  then  # check the next argument for explicit script name  if [ $# -gt "1" ]  then  shift  # if the argument begins with "-", ignore it  # because it is not a script file name  if [ `echo "$1" | cut -c 1` != "-" ]  then  scriptfile="$1"  fi  fi  # make sure the file does exist before setting exec permission  if [ -f $scriptfile ]  then  chmod +x $scriptfile  fi  fi  shift  done  }  # Bootstrap values ...  SHELL=com.ibm.ws.management.tools.WsServerLauncher  binDir=`dirname $0`  . $binDir/setupCmdLine.sh  #The following is added through defect 236497.1  CMD\_NAME=`basename $0`  if [ -f ${JAVA\_HOME}/bin/java ]; then  JAVA\_EXE="${JAVA\_HOME}/bin/java"  else  JAVA\_EXE="${JAVA\_HOME}/jre/bin/java"  fi  if [ "${INV\_PRF\_SPECIFIED:=}" != "" ] && [ "$INV\_PRF\_SPECIFIED" = "true" ]  then  ${JAVA\_EXE} -cp "${WAS\_HOME}/lib/commandlineutils.jar" com.ibm.ws.install.commandline.utils.CommandLineUtils -specifiedProfileNotExists -profileName $WAS\_PROFILE\_NAME  exit 1  elif [ "${WAS\_USER\_SCRIPT\_FILE\_NOT\_EXISTS:=}" != "" ] && [ "$WAS\_USER\_SCRIPT\_FILE\_NOT\_EXISTS" = "true" ]  then  ${JAVA\_EXE} -cp "${WAS\_HOME}/lib/commandlineutils.jar" com.ibm.ws.install.commandline.utils.CommandLineUtils -wasUserScriptFileNotExists -wasUserScript $WAS\_USER\_SCRIPT  exit 1  elif [ "${NO\_DFT\_PRF\_EXISTS:=}" != "" ] && [ "$NO\_DFT\_PRF\_EXISTS" = "true" ]  then  ${JAVA\_EXE} -cp "${WAS\_HOME}/lib/commandlineutils.jar" com.ibm.ws.install.commandline.utils.CommandLineUtils -noDefaultProfile -commandName $CMD\_NAME  exit 1  fi  # For debugging the launcher itself  # WAS\_DEBUG="-Djava.compiler=NONE -Xdebug -Xnoagent -Xrunjdwp:transport=dt\_socket,server=y,suspend=y,address=7777"  #zOS  # "$JAVA\_HOME"/bin/java -Dfile.encoding=ISO8859-1 -Xnoargsconversion \  # $DEBUG \  # -Dws.ext.dirs="$WAS\_EXT\_DIRS" \  # $CONSOLE\_ENCODING \  # -Djava.ext.dirs="$JAVA\_EXT\_DIRS" \  # -classpath "$WAS\_CLASSPATH" \  # -Dwas.install.root="$WAS\_HOME" \  # -Dwas.serverstart.cell="$WAS\_CELL" \  # -Dwas.serverstart.node="$WAS\_NODE" \  # -Dwas.serverstart.server="$1" \  # $USER\_INSTALL\_PROP \  # $JVM\_EXTRA\_CMD\_ARGS \  # com.ibm.ws.bootstrap.WSLauncher \  # $SHELL "$CONFIG\_ROOT" "$WAS\_CELL" "$WAS\_NODE" "$@"  #ASV  # "$JAVA\_HOME"/bin/java \  # $DEBUG \  # -Dws.ext.dirs="$WAS\_EXT\_DIRS" \  # -classpath "$WAS\_CLASSPATH" \  # -Dwas.install.root="$WAS\_HOME" \  # $USER\_INSTALL\_PROP \  # com.ibm.ws.bootstrap.WSLauncher \  # $SHELL "$CONFIG\_ROOT" "$WAS\_CELL" "$WAS\_NODE" "$@"  # Setup the initial java invocation;  DELIM=" "  #Platform specific args...  PLATFORM=`/bin/uname`  if [ "$PLATFORM" = "OS/390" ] && [ ! -e "$WAS\_HOME/properties/service/product/WebSphere/backup/396435/defect.applied" ]  then  WAS\_EXT\_DIRS="$WAS\_EXT\_DIRS:$WAS\_SMPE\_ROOT/plugins"  fi  #Common args...  D\_ARGS="-Dws.ext.dirs="$WAS\_EXT\_DIRS" $DELIM -Dwas.install.root="$WAS\_HOME" $DELIM -Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager $DELIM -Djava.util.logging.configureByServer=true"  case $PLATFORM in  AIX)  EXTSHM=ON  LIBPATH="$WAS\_LIBPATH":$LIBPATH  export LIBPATH EXTSHM ;;  Linux)  LD\_LIBRARY\_PATH="$WAS\_LIBPATH":$LD\_LIBRARY\_PATH  export LD\_LIBRARY\_PATH ;;  SunOS)  LD\_LIBRARY\_PATH="$WAS\_LIBPATH":$LD\_LIBRARY\_PATH  export LD\_LIBRARY\_PATH ;;  HP-UX)  SHLIB\_PATH="$WAS\_LIBPATH":$SHLIB\_PATH  export SHLIB\_PATH ;;  OS/390)  PATH="$PATH":$binDir  export PATH  D\_ARGS=""$D\_ARGS" $DELIM -Dfile.encoding=ISO8859-1 $DELIM -Djava.ext.dirs="$JAVA\_EXT\_DIRS""  D\_ARGS=""$D\_ARGS" $DELIM -Dwas.serverstart.cell="$WAS\_CELL""  D\_ARGS=""$D\_ARGS" $DELIM -Dwas.serverstart.node="$WAS\_NODE""  D\_ARGS=""$D\_ARGS" $DELIM -Dwas.serverstart.server="$1""  X\_ARGS="-Xnoargsconversion" ;;  esac  PATH=${WAS\_DB2\_PATH\_VAR:+"$WAS\_DB2\_PATH\_VAR":}"$PATH"  export PATH  "$JAVA\_HOME"/bin/java \  "$OSGI\_INSTALL" "$OSGI\_CFG" \  $X\_ARGS \  $WAS\_DEBUG \  $CONSOLE\_ENCODING \  $D\_ARGS \  -classpath "$WAS\_CLASSPATH" \  $USER\_INSTALL\_PROP \  $JVM\_EXTRA\_CMD\_ARGS \  com.ibm.ws.bootstrap.WSLauncher \  $SHELL "$CONFIG\_ROOT" "$WAS\_CELL" "$WAS\_NODE" "$@" $WORKSPACE\_ROOT\_PROP  launchExit=$?  if [ "$launchExit" = "0" ]  then  set\_script\_executable $@  fi  exit `expr $launchExit + $?`  ~ |

### stopServer.sh

|  |
| --- |
| #!/bin/sh  # All Rights Reserved \* Licensed Materials - Property of IBM  # 5724-I63, 5724-H88, 5655-N01, 5733-W60 (C) COPYRIGHT International Business Machines Corp., 1997,2005  # US Government Users Restricted Rights - Use, duplication or disclosure  # restricted by GSA ADP Schedule Contract with IBM Corp.  # Server Stop Tool  # Launch Arguments:  #  # serverName - the name of the server process to be stopped.  # -nowait  # -quiet  # -trace  # -timeout <seconds>  # -statusport <portnumber>  # -conntype <connector type>  # -port <portnumber>  # -username <name>  # -password <password>  # -help  # -?  #  # Bootstrap values ...  SHELL=com.ibm.ws.runtime.WsServerStop  binDir=`dirname $0`  . $binDir/setupCmdLine.sh  #The following is added through defect 236497.1  CMD\_NAME=`basename $0`  if [ -f ${JAVA\_HOME}/bin/java ]; then  JAVA\_EXE="${JAVA\_HOME}/bin/java"  else  JAVA\_EXE="${JAVA\_HOME}/jre/bin/java"  fi  if [ "${INV\_PRF\_SPECIFIED:=}" != "" ] && [ "$INV\_PRF\_SPECIFIED" = "true" ]  then  ${JAVA\_EXE} -cp "${WAS\_HOME}/lib/commandlineutils.jar" com.ibm.ws.install.commandline.utils.CommandLineUtils -specifiedProfileNotExists -profileName $WAS\_PROFILE\_NAME  exit 1  elif [ "${WAS\_USER\_SCRIPT\_FILE\_NOT\_EXISTS:=}" != "" ] && [ "$WAS\_USER\_SCRIPT\_FILE\_NOT\_EXISTS" = "true" ]  then  ${JAVA\_EXE} -cp "${WAS\_HOME}/lib/commandlineutils.jar" com.ibm.ws.install.commandline.utils.CommandLineUtils -wasUserScriptFileNotExists -wasUserScript $WAS\_USER\_SCRIPT  exit 1  elif [ "${NO\_DFT\_PRF\_EXISTS:=}" != "" ] && [ "$NO\_DFT\_PRF\_EXISTS" = "true" ]  then  ${JAVA\_EXE} -cp "${WAS\_HOME}/lib/commandlineutils.jar" com.ibm.ws.install.commandline.utils.CommandLineUtils -noDefaultProfile -commandName $CMD\_NAME  exit 1  fi  #zOS  # "$JAVA\_HOME"/bin/java -Dfile.encoding=ISO8859-1 -Xnoargsconversion \  # -Xbootclasspath/p:"$WAS\_BOOTCLASSPATH" \  # $DEBUG \  # "$CLIENTSAS" \  # "$CLIENTSSL" \  # "$CLIENTSOAP" \  # -Dws.ext.dirs="$WAS\_EXT\_DIRS" \  # -Djava.ext.dirs="$JAVA\_EXT\_DIRS" \  # -classpath "$WAS\_CLASSPATH" \  # -Dwas.install.root="$WAS\_HOME" \  # $CONSOLE\_ENCODING \  # $USER\_INSTALL\_PROP \  # $JVM\_EXTRA\_CMD\_ARGS \  # com.ibm.ws.bootstrap.WSLauncher \  # $SHELL "$CONFIG\_ROOT" "$WAS\_CELL" "$WAS\_NODE" "$@"  #ASV  # "$JAVA\_HOME"/bin/java \  # -Xbootclasspath/p:"$WAS\_BOOTCLASSPATH" \  # $DEBUG \  # "$CLIENTSAS" \  # "$CLIENTSSL" \  # "$CLIENTSOAP" \  # -Dws.ext.dirs="$WAS\_EXT\_DIRS" \  # -classpath "$WAS\_CLASSPATH" \  # -Dwas.install.root="$WAS\_HOME" \  # $USER\_INSTALL\_PROP \  # com.ibm.ws.bootstrap.WSLauncher \  # $SHELL "$CONFIG\_ROOT" "$WAS\_CELL" "$WAS\_NODE" "$@"  # For debugging the launcher itself  # WAS\_DEBUG="-Djava.compiler=NONE -Xdebug -Xnoagent -Xrunjdwp:transport=dt\_socket,server=y,suspend=y,address=7777"  # Setup the initial java invocation;  DELIM=" "  #Common args...  D\_ARGS="-Dws.ext.dirs="$WAS\_EXT\_DIRS" $DELIM -Dwas.install.root="$WAS\_HOME" $DELIM -Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager $DELIM -Djava.util.logging.configureByServer=true"  X\_ARGS=-Xbootclasspath/p:"$WAS\_BOOTCLASSPATH"  #Platform specific args...  PLATFORM=`/bin/uname`  case $PLATFORM in  AIX)  LIBPATH="$WAS\_LIBPATH":$LIBPATH  export LIBPATH ;;  Linux)  LD\_LIBRARY\_PATH="$WAS\_LIBPATH":$LD\_LIBRARY\_PATH  export LD\_LIBRARY\_PATH ;;  SunOS)  LD\_LIBRARY\_PATH="$WAS\_LIBPATH":$LD\_LIBRARY\_PATH  export LD\_LIBRARY\_PATH ;;  HP-UX)  SHLIB\_PATH="$WAS\_LIBPATH":$SHLIB\_PATH  export SHLIB\_PATH ;;  OS/390)  D\_ARGS=""$D\_ARGS" $DELIM -Dfile.encoding=ISO8859-1 $DELIM -Djava.ext.dirs="$JAVA\_EXT\_DIRS""  X\_ARGS=""$X\_ARGS" $DELIM -Xnoargsconversion" ;;  esac  "$JAVA\_HOME"/bin/java \  "$OSGI\_INSTALL" "$OSGI\_CFG" \  $X\_ARGS \  $WAS\_DEBUG \  "$CLIENTSAS" \  "$CLIENTSSL" \  "$CLIENTSOAP" \  ${JAASSOAP:+"$JAASSOAP"} \  $D\_ARGS \  -classpath "$WAS\_CLASSPATH" \  $CONSOLE\_ENCODING \  $USER\_INSTALL\_PROP \  $JVM\_EXTRA\_CMD\_ARGS \  com.ibm.wsspi.bootstrap.WSPreLauncher -nosplash -application com.ibm.ws.bootstrap.WSLauncher \  $SHELL "$CONFIG\_ROOT" "$WAS\_CELL" "$WAS\_NODE" "$@" |

## 部署失败解决方法

|  |
| --- |
| 可用到缓存目录(jsp编译过来的文件)  cd /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/temp/webNode01/server1  用 rm -rf 文件名 删除项目临时文件 |

## websphere中应用的web.xml所在目录

|  |
| --- |
| /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/webNode01Cell/applications/  ismp-webapp\_war.ear/deployments/ismp-webapp\_war/ismp-webapp.war/WEB-INF |

## websphere日志

|  |
| --- |
| /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1 |