**Build CBA using Jenkins**

Jenkins job can be used to pull up code from TFS repository into build machine. Build machine will require following tools and setup.

* GIT client to pull up code from TFS.
* Apache ANT tool to build CBA in build machine.
* Jenkins job can be refer for specific master branch or specific release branch code.

**Deployment of CBA into MDM Operational Server as an extension**

Once CBA is built in build machine with latest MDM 11.6 JARs using JENKINS+ GIT+ ANT tool, custom script can be used to deploy same CBA into MDM WAS server.

Following are script components used for custom script process.

1. installCba.properties

Property file will have following four parameters.

was.profile.dir=/opt/IBM/WAS/WebSphere9/AppServer/profiles/Dmgr01

mdm.install.dir=/opt/IBM/110610

cba.assets.dir=/home/wasadmin/deploy/cba

mdm.admin.password=<password>

remove.cba.name=testCBA.cba

remove.cba.bundleVersion=1.0.0.202106221600

1. addCba.sh
2. addCba.py
3. installCba.sh

installCba.sh is master script which will read installCba.property file and will copy addCba.sh and addCba.py into $WAS\_PROFILE\_DIR/bin.

Then script will invoke addCba.sh for each cba present in cba.assets.dir. Script will in turn invokes the addCba.py using the wsadmin.sh tool from WAS and passes in parameters CBA location, Instance Id, Bundle Name and Bundle Version. The addCba.py file adds the local repository bundle, adds it as an extension to the Enterprise Business Archive in MDM operational server Business Level Application, edits the composition unit and saves it. if there are multiple “.cba” files available inside “cba.assets.dir” then script will assume that more than one cba need to be deployed into WAS directory.

Similarly, removeCba.sh and removeCba.py can be used to remove any existing CBA from MDM server. Last two properties used in property file being used for CBA removal.

########################## installCba.sh Starts Here ###################################

#!/bin/bash

echo "read variables from property file"

MDM\_INSTALL\_DIR=$(cat installCba.properties | grep mdm.install.dir | awk -F "mdm.install.dir=" '{print $2}' | sed 's|\\||g')

WAS\_PROFILE\_DIR=$(cat installCba.properties | grep was.profile.dir | awk -F "was.profile.dir=" '{print $2}' | sed 's|\\||g')

CBA\_ASSETS\_DIR=$(cat installCba.properties | grep cba.assets.dir | awk -F "cba.assets.dir=" '{print $2}' | sed 's|\\||g')

MDM\_ADMIN\_PASSWORD=$(cat installCba.properties | grep mdm.admin.password | awk -F "mdm.admin.password=" '{print $2}' | sed 's|\\||g')

REMOVE\_CBA\_NAME=$(cat installCba.properties | grep remove.cba.name | awk -F "remove.cba.name=" '{print $2}' | sed 's|\\||g')

REMOVE\_CBA\_BUNDLEVERSION=$(cat installCba.properties | grep remove.cba.bundleVersion | awk -F "remove.cba.bundleVersion=" '{print $2}' | sed 's|\\||g')

#MDM\_DB\_TYPE=$(cat installCba.properties | grep mdm.db.type | awk -F "mdm.db.type=" '{print $2}' | sed 's|\\||g')

export PATH=$PATH:$WAS\_PROFILE\_DIR/bin

CURRENT\_DIR=$(pwd)

echo "Moving scripts to bin folder in WAS Profile Directory"

cp addCba.sh $WAS\_PROFILE\_DIR/bin

cp addCba.py $WAS\_PROFILE\_DIR/bin

#Copy script to remove existing CBA

#cp removeCba.sh $WAS\_PROFILE\_DIR/bin

#cp removeCba.py $WAS\_PROFILE\_DIR/bin

CURRENT\_DIR=$(pwd)

cd $WAS\_PROFILE\_DIR/bin

#echo "Remove exsiting CBA, uncomment below line to remove cba configured in property file"

#./removeCba.sh $MDM\_INSTALL\_DIR $REMOVE\_CBA\_NAME $REMOVE\_CBA\_BUNDLEVERSION $MDM\_ADMIN\_PASSWORD

echo "Adding CBAs"

for cba\_file in $CBA\_ASSETS\_DIR/\*.cba;

do

./addCba.sh $MDM\_INSTALL\_DIR $cba\_file $MDM\_ADMIN\_PASSWORD

done

echo "Complete"

########################## installCba.sh Ends Here ###################################

########################## addCba.sh Starts Here ####################################

#!/bin/sh

#Place the file at <WAS\_PROFILE\_HOME>/bin

#Usage

#./addCba.sh <MDM\_INSTALL\_DIR> <CBA\_NAME\_WITH\_PATH> <WAS\_ADMIN\_PASSWORD>

if [[ $# -lt 3 || $# -gt 3 ]]; then

echo "Provide three arguments <MDM\_INSTALL\_DIR> <CBA\_NAME\_WITH\_PATH> <WAS\_ADMIN\_PASSWORD>

exit

fi

MDM\_INSTALL\_DIR=$1

CBA\_PATH=$2

WAS\_ADMIN\_PASSWORD=$3

INSTANCE\_ID=$(cat ${MDM\_INSTALL\_DIR}/properties/mdm\_install.properties | grep user.instance.identifier | awk -F "user.instance.identifier=" '{print $2}' | sed 's|\\||g')

WAS\_ADMIN\_USER=$(cat ${MDM\_INSTALL\_DIR}/mds/scripts/soap.client.props | grep com.ibm.SOAP.loginUserid | awk -F "com.ibm.SOAP.loginUserid=" '{print $2}' | sed 's|\\||g')

WAS\_DEPLOY\_HOST=$(cat ${MDM\_INSTALL\_DIR}/properties/mdm\_install.properties | grep user.deploy.host | awk -F "user.deploy.host=" '{print $2}' | sed 's|\\||g')

WAS\_DEPLOY\_PORT=$(cat ${MDM\_INSTALL\_DIR}/properties/mdm\_install.properties | grep user.deploy.port | awk -F "user.deploy.port=" '{print $2}' | sed 's|\\||g')

mkdir -p tmp/cba

cp $CBA\_PATH tmp/cba

cd tmp/cba

unzip -o -q \*.cba

BUNDLE\_VERSION=`sed -n '/^Bundle-Version/s/^Bundle-Version: \*//p' META-INF/COMPOSITEBUNDLE.MF`

BUNDLE\_NAME=`sed -n '/^Bundle-Name/s/^Bundle-Name: \*//p' META-INF/COMPOSITEBUNDLE.MF`

cd ../..

rm -Rf tmp/cba

./wsadmin.sh -conntype SOAP -host $WAS\_DEPLOY\_HOST -port $WAS\_DEPLOY\_PORT -lang jython -username $WAS\_ADMIN\_USER -password $WAS\_ADMIN\_PASSWORD -f addCba.py $CBA\_PATH $MDM\_INSTANCE\_ID $BUNDLE\_NAME $BUNDLE\_VERSION

# Usage

#/wsadmin.sh -conntype SOAP -host <HOST> -port <SOAP\_PORT> -lang jython -username <WAS\_ADMIN\_USER> -password <WAS\_ADMIN\_PASSWORD> -f addCba.py <CBA\_PATH> <MDM\_INSTANCE\_ID> <BUNDLE\_NAME> <BUNDLE\_VERSION>

########################## addCba.sh Ends Here ###################

######################### addCba.py Starts Here ######################

def syncAllNodes():

for node in AdminConfig.list('Node').splitlines():

nodeName = AdminConfig.showAttribute(node, 'name')

nodeSync = AdminControl.completeObjectName('type=NodeSync,node=' + nodeName + ',\*')

try:

AdminControl.invoke(nodeSync, 'sync')

except:

pass

varCBAPath = sys.argv[0]varInstanceId = sys.argv[1]varBN = sys.argv[2]varBV = sys.argv[3]

AdminTask.addLocalRepositoryBundle('-file ' + varCBAPath)

syncAllNodes()

AdminConfig.save()

AdminTask.addOSGiExtension('-cuName com.ibm.mdm.hub.server.app-' + varInstanceId + '\_0001.eba -symbolicName ' + varBN + ' -version ' + varBV)

syncAllNodes()

AdminConfig.save()

AdminTask.editCompUnit('[-blaID WebSphere:blaname=MDM-operational-server-EBA-' + varInstanceId + ' -cuID WebSphere:cuname=com.ibm.mdm.hub.server.app-' + varInstanceId + '\_0001.eba -CompUnitStatusStep [[com.ibm.mdm.hub.server.app-' + varInstanceId + '.eba true]]]')

syncAllNodes()

AdminConfig.save()

############################# addCba.py End Here #######################

############################# removeCba.py Starts Here #######################

def syncAllNodes():

for node in AdminConfig.list('Node').splitlines():

nodeName = AdminConfig.showAttribute(node, 'name')

nodeSync = AdminControl.completeObjectName('type=NodeSync,node=' + nodeName + ',\*')

try:

AdminControl.invoke(nodeSync, 'sync')

except:

pass

varInstanceId = sys.argv[0]varBN = sys.argv[1]varBV = sys.argv[2]

AdminTask.removeOSGiExtension('-cuName com.ibm.mdm.hub.server.app-' + varInstanceId + '\_0001.eba -symbolicName ' + varBN + ' -version ' + varBV)

syncAllNodes()

AdminConfig.save()

AdminTask.editCompUnit('[-blaID WebSphere:blaname=MDM-operational-server-EBA-' + varInstanceId + ' -cuID WebSphere:cuname=com.ibm.mdm.hub.server.app-' + varInstanceId + '\_0001.eba -CompUnitStatusStep [[com.ibm.mdm.hub.server.app-' + varInstanceId + '.eba true]]]')

syncAllNodes()

AdminConfig.save()

AdminTask.removeLocalRepositoryBundle('-symbolicName ' + varBN + ' -version ' + varBV)

syncAllNodes()

AdminConfig.save()

############################# removeCba.py End Here #######################

############################# removeCba.sh Starts Here ######################

#!/bin/sh

#Place the file at <WAS\_PROFILE\_HOME>/bin

if [[ $# -lt 4 || $# -gt 4 ]]; then

echo "Provide four arguments <MDM\_INSTALL\_HOME> <BUNDLE\_NAME> <BUNDLE\_VERSION> <MDM\_ADMIN\_PASSWORD>"

echo "Example: ./removeCba.sh /opt/IBM/MDM Party.cba 12.0.0.201609080245 mdmadmin"

exit

fi

MDM\_INSTALL\_DIR=$1

BUNDLE\_NAME=$2

BUNDLE\_VERSION=$3

WAS\_ADMIN\_PASSWORD=$4

INSTANCE\_ID=$(cat ${MDM\_INSTALL\_DIR}/properties/mdm\_install.properties | grep user.instance.identifier | awk -F "user.instance.identifier=" '{print $2}' | sed 's|\\||g')

WAS\_ADMIN\_USER=$(cat ${MDM\_INSTALL\_DIR}/mds/scripts/soap.client.props | grep com.ibm.SOAP.loginUserid | awk -F "com.ibm.SOAP.loginUserid=" '{print $2}' | sed 's|\\||g')

./wsadmin.sh -lang jython -username $WAS\_ADMIN\_USER -password $WAS\_ADMIN\_PASSWORD -f removeCba.py $INSTANCE\_ID $BUNDLE\_NAME $BUNDLE\_VERSION

############################# removeCba.sh End Here ######################